Back End
Basic training

Low Code Studio

How to use this tutorial ?

Welcome to your first journey
with Convertigo Low Code Studio.
Let’s explore its many features.

Concepts & Definitions

Convertigo uses many concepts
you may not be familiar with.
Find answers with this icon.

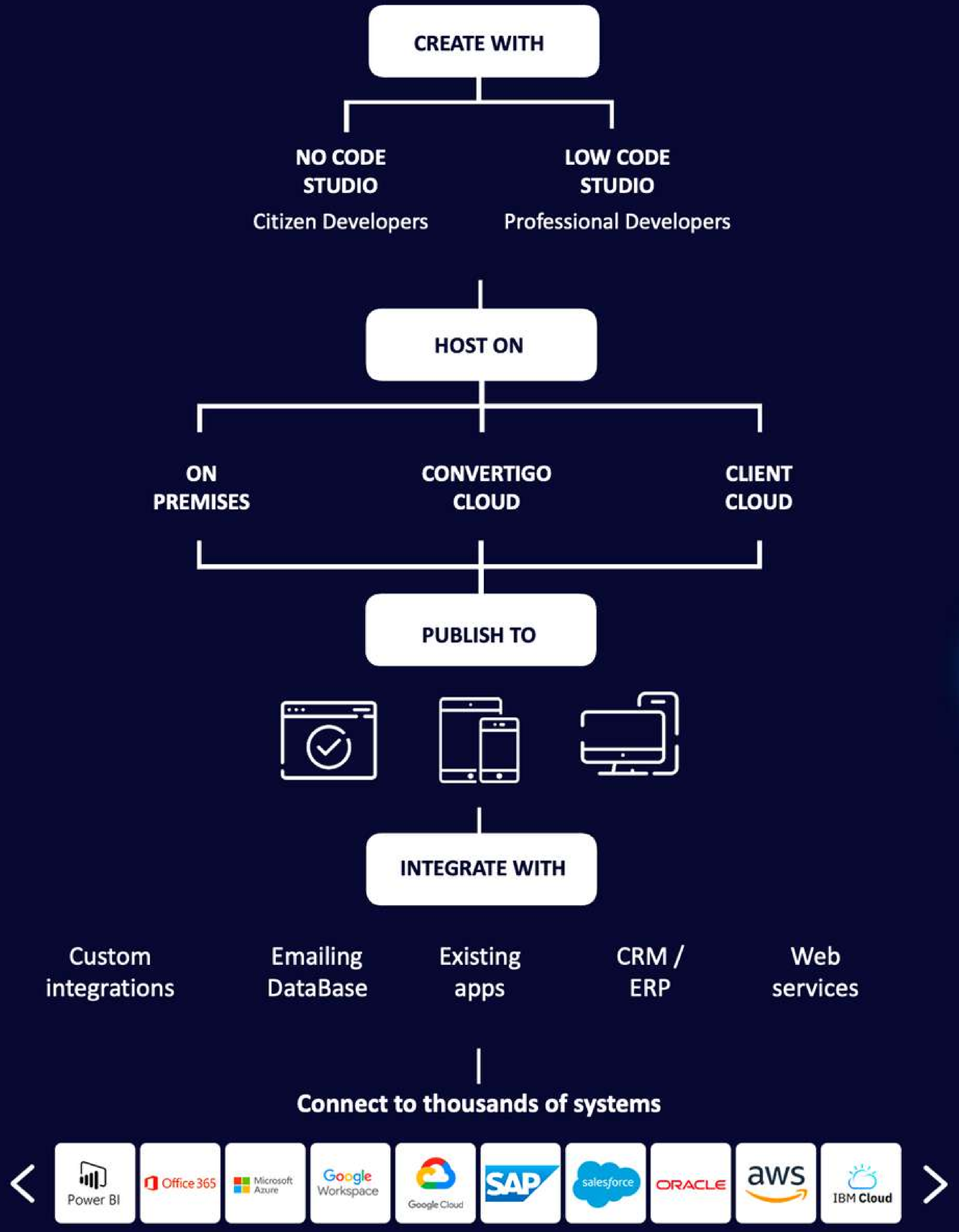
Practice time

You prefer to skip the concepts
and start by practice.
Go straight to this icon.

convertigo

What is Convertigo Low code Platform ?

- Full Stack
- Low Code
- Open Source
- Application Development Platform



CREATE WITH

NO CODE STUDIO
Citizen Developers

LOW CODE STUDIO
Professional Developers

HOST ON

ON PREMISES

CONVERTIGO CLOUD

CLIENT CLOUD

PUBLISH TO

INTEGRATE WITH

Custom integrations

Emailing DataBase

Existing apps

CRM / ERP

Web services

Connect to thousands of systems

What can you do with

Convertigo Low code Studio ?

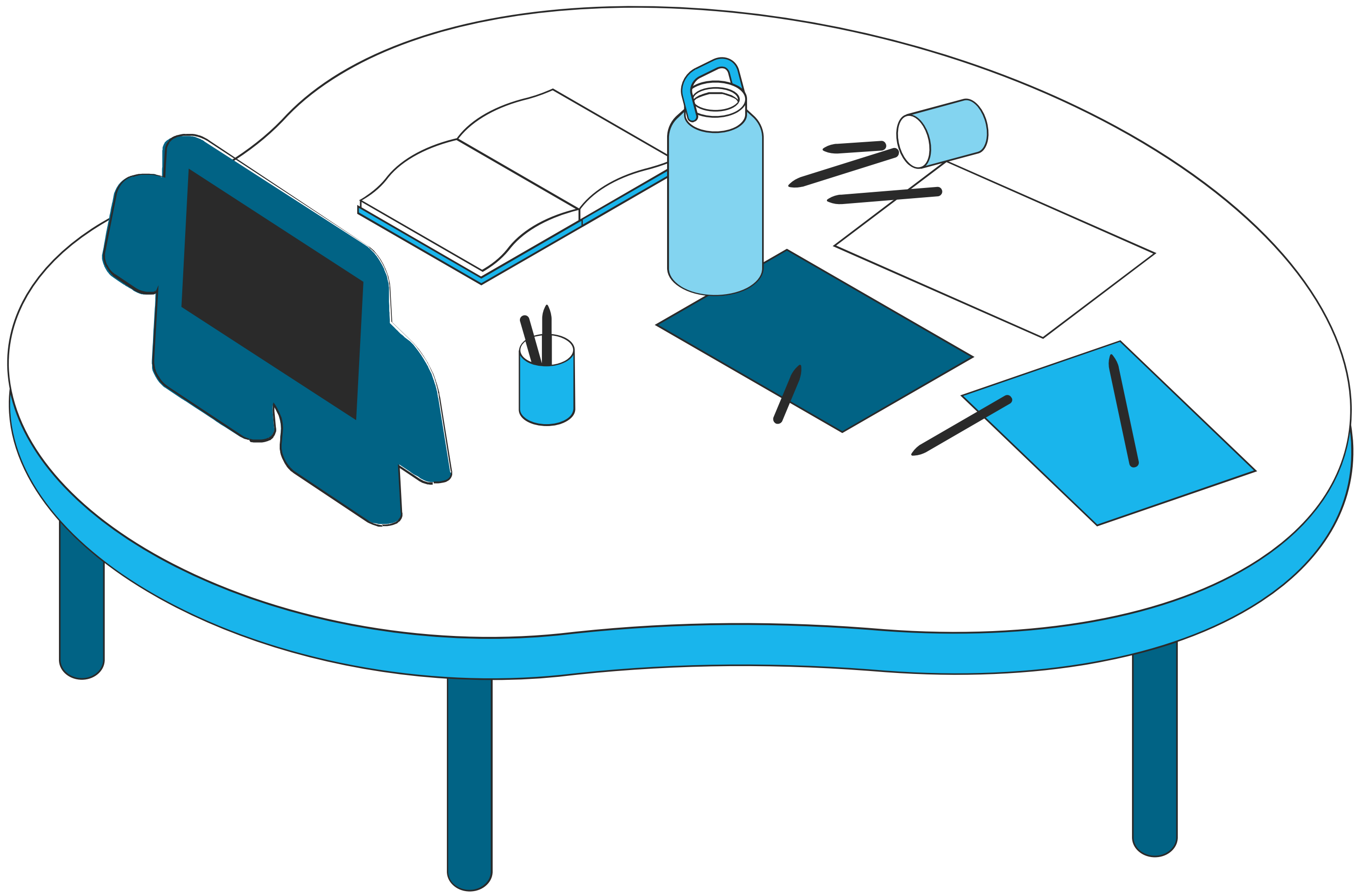
- ☑ Connect to back end systems with **Connectors**
- ☑ Exchange data with the backend using **Transactions**
- ☑ Define backend flows and business logic with **Sequences**
- ☑ Create web and mobile user interfaces with **Pages** and **UI Components**
- ☑ Create **iOS, Android, Progressive Web Apps** and **Web applications** from the same project
- ☑ Define and execute **Test cases**
- ☑ Share your projects with **Git versioning**

Table of Contents

convertigo

1 – Introduction

Overview of the studio.



1.1 Technical knowledge

1.2 Global architecture

1.3 Convertigo Server

1.4 Objects in Convertigo

1.5 Back-end Objects

1.6 Studio Interface

1.7 Panels & Views

1.1 Technical knowledge

The following concepts are necessary for mastering the studio.

WEB TECHNOLOGIES

- XML and XPath
- JavaScript & JSON
- HTTP requests
- REST API

ALGORITHMS

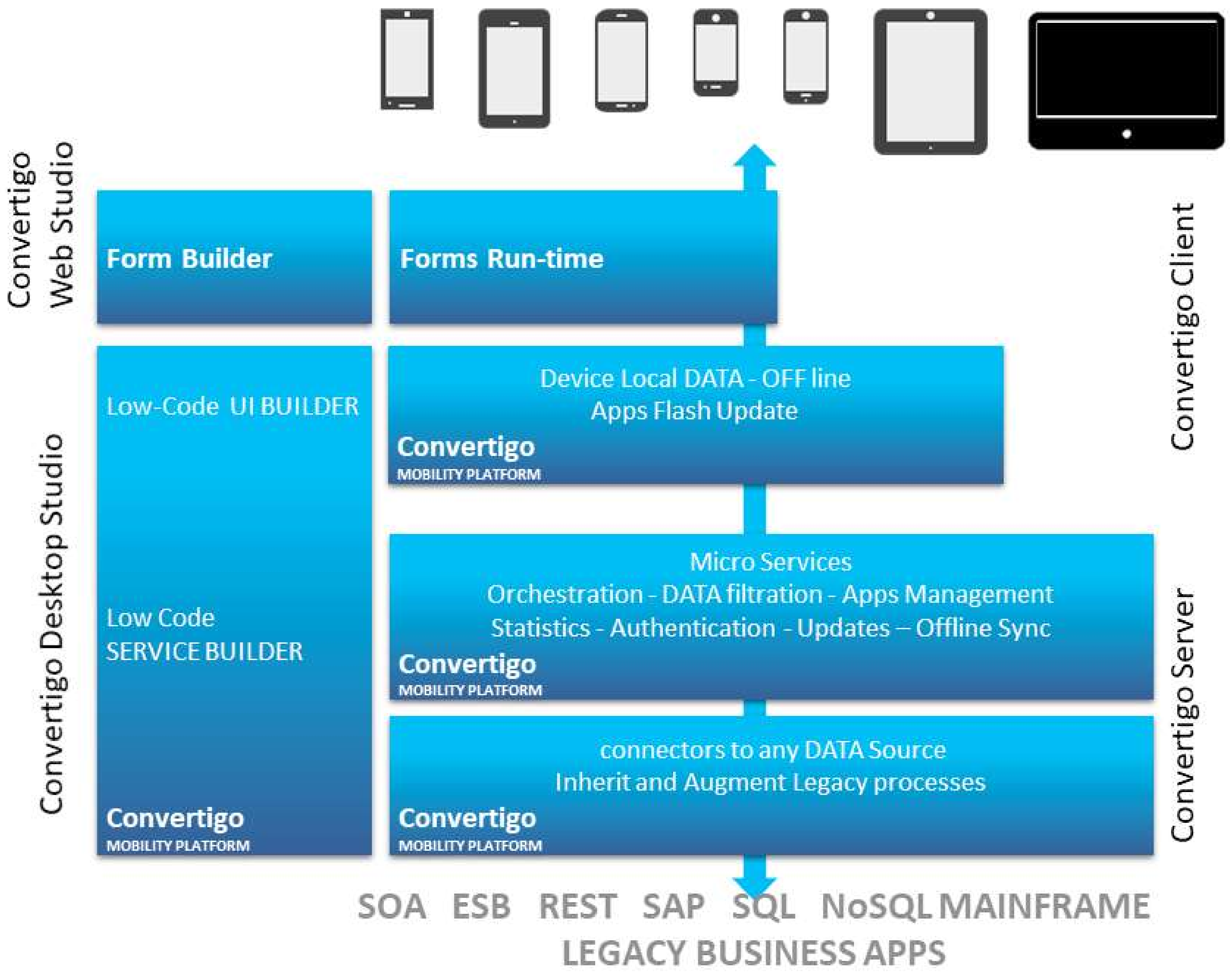
- Pseudocode basics
- Loops, Conditional statements...

DATABASES

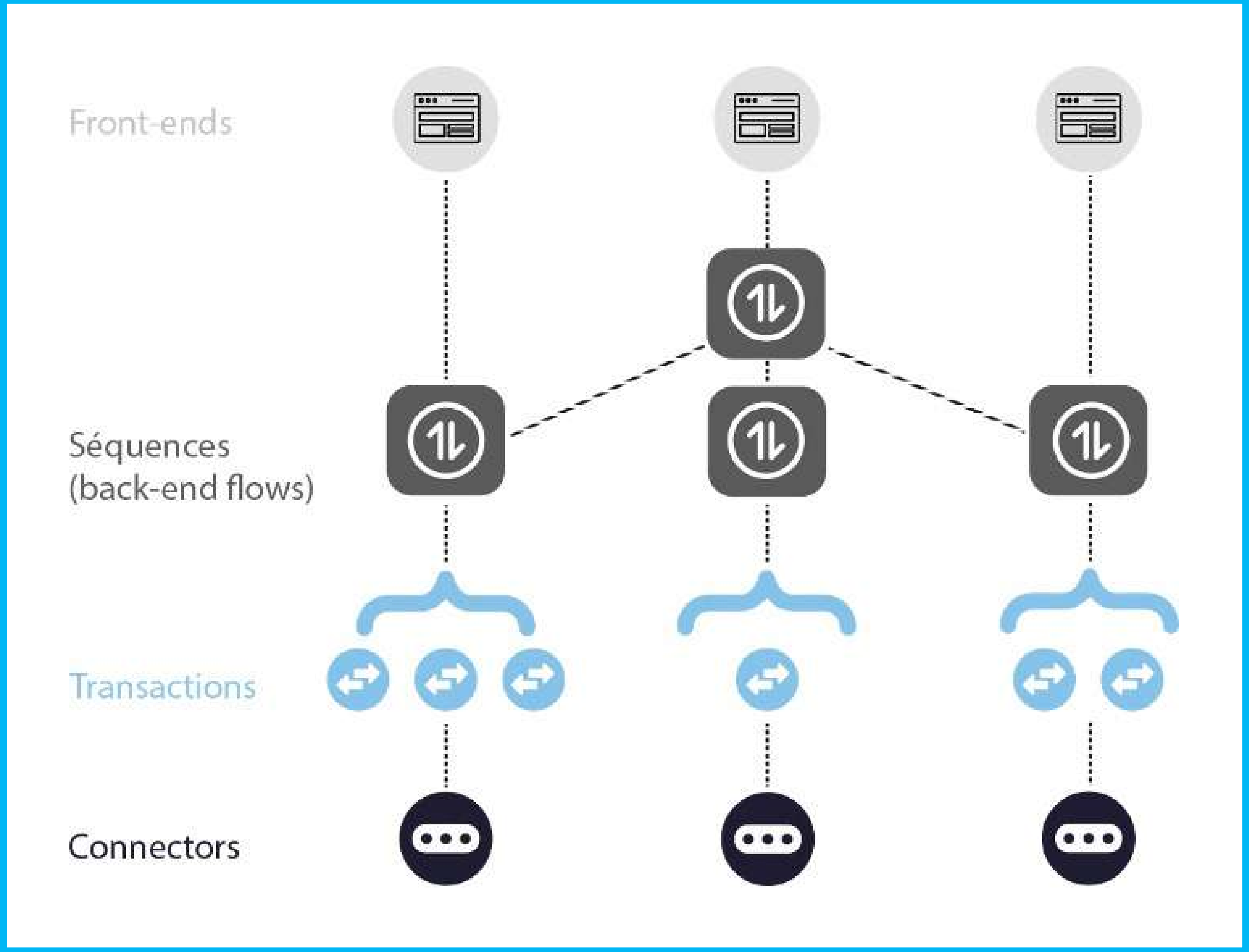
- SQL basics
- NoSQL basics

1.2 Global architecture

- Full Stack Low Code Platform
- Built on Docker Container Micro service Architecture
- Based on Open Standard technologies
  - (Eclipse, Node, Cordova, Angular, Java, Docker, Kubernetes, CouchDB, …)
- Out of the box Connectors to Enterprise Data
- 100% Offline Data sync technology
- Interface with AI & ChatBots
- Includes 100% Web Studio for Form Builder



1.3 Convertigo Server



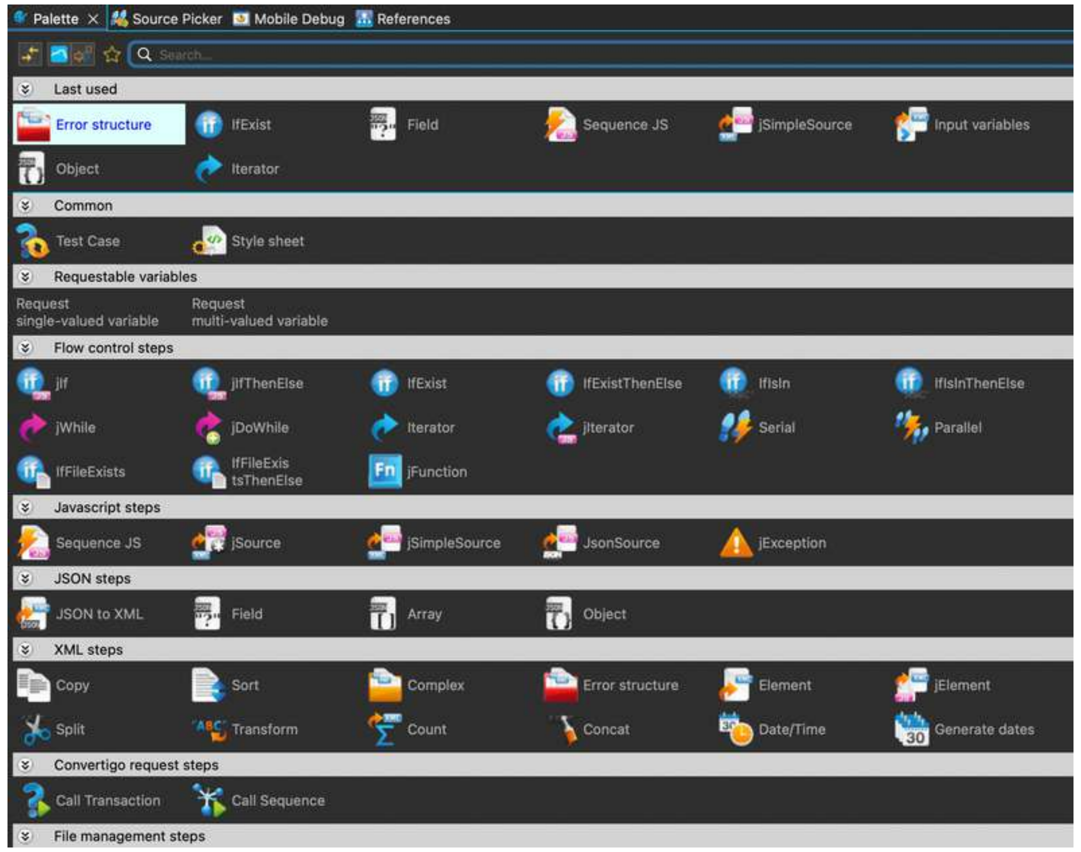
- ☑ Runs the **back-end** of the application
- ☑ Can be used to **deploy** as many apps as wanted
- ☑ Handles data in a **NoSQL database**
- ☑ Provides **connectors** to many **data providers** (SQL, Web services, Legacy apps running on mainframes...)
- ☑ Runs in **Docker container platforms** as a **Docker Image** (Cloud providers, Kubernetes on premises...)

1.4 Objects in Convertigo

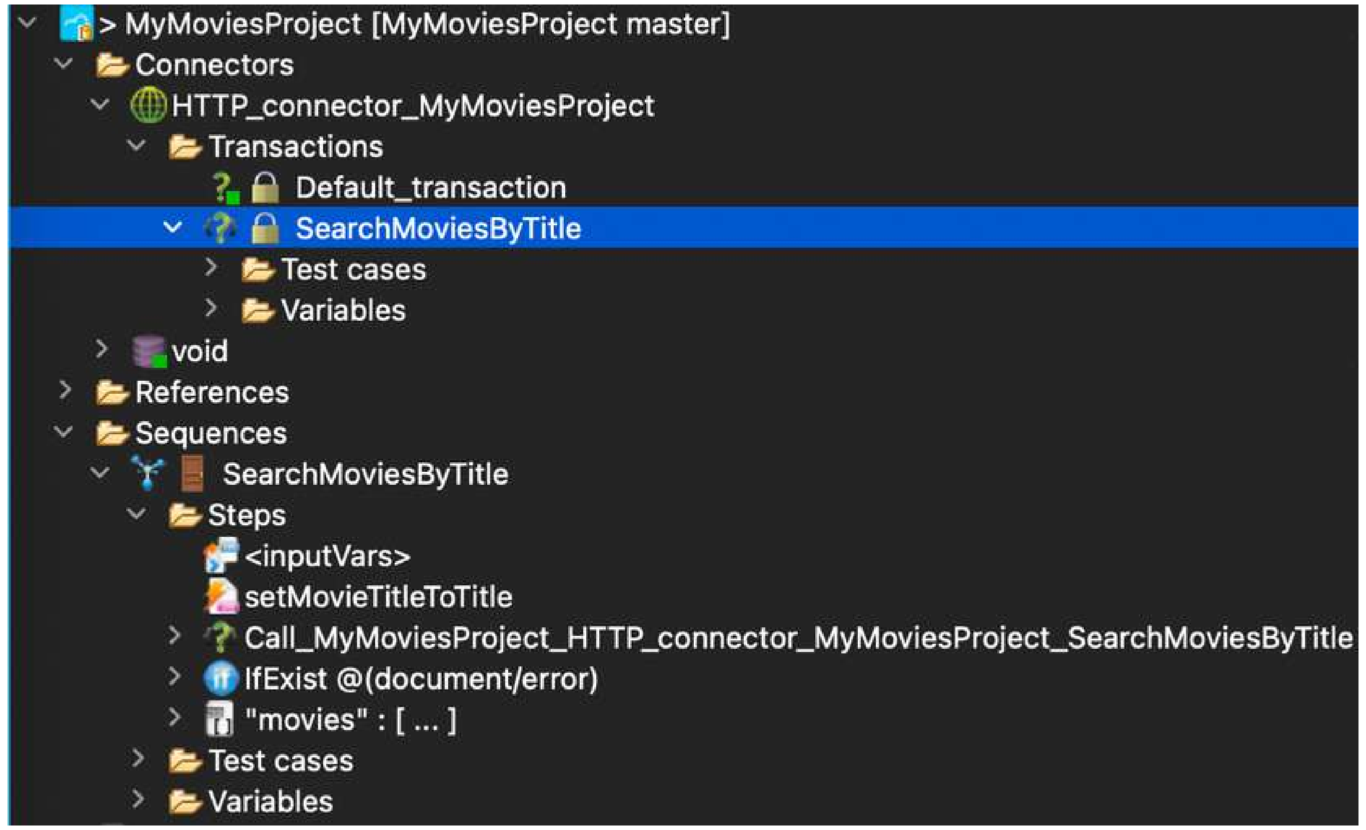
In Convertigo, **Objects** refer to **structured components** that **encapsulate data, functions, and properties**.

Objects are used to r**epresent and manipulate various elements** in Convertigo projects.

The objects are available in the **Palette view**.



A Convertigo project is **organized in a treeview**. where you **drop objects dragged from the palette**.



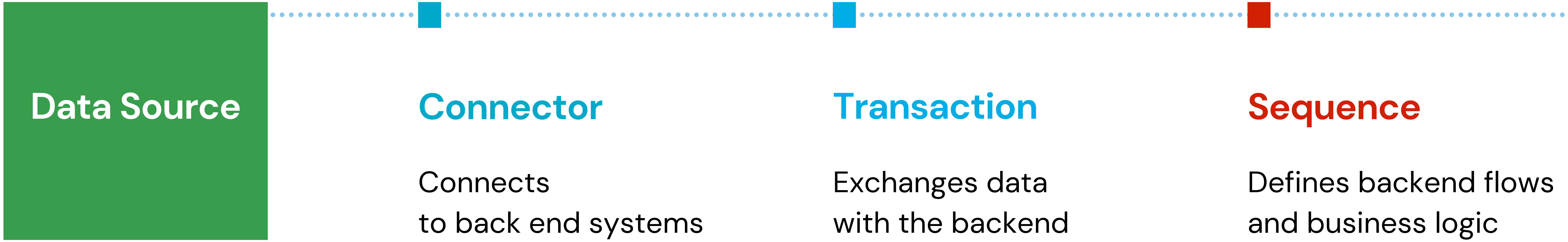
1.5 Back-end Objects

In a Convertigo project, **back-end objects** handle the **back end processing**.

There are **3 main back-end objects** : **Connector, Transaction, and Sequence**.

Sequences interact with Connectors and Transactions

to read and write data to Databases, WebServices or Third party applications.



Data Source

Connector

Connects
to back end systems

Transaction

Exchanges data
with the backend

Sequence

Defines backend flows
and business logic

1.6 Studio Interface

The studio interface is divided in **5 main panels**. Each one contains **several views**.

PROJECTS PANEL

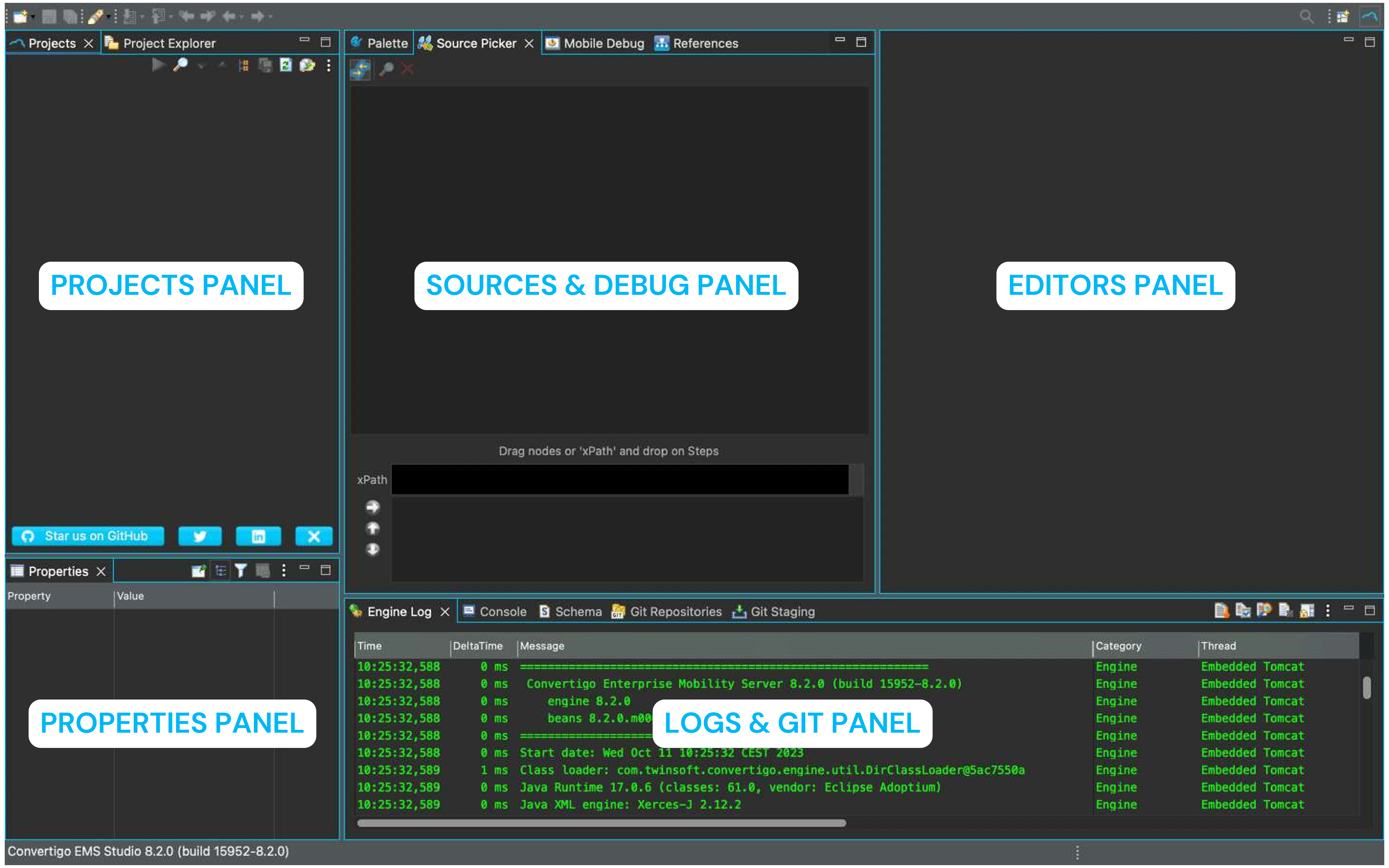
- PROJECTS
- PROJECT EXPLORER

SOURCES & DEBUG PANEL

- PALETTE
- SOURCE PICKER
- REFERENCES
- MOBILE DEBUG

PROPERTIES PANEL

- PROPERTIES



```
Time          DeltaTime  Message                                                                    Category  Thread
10:25:32,588       0 ms  ===========================================================                Engine    Embedded Tomcat
10:25:32,588       0 ms   Convertigo Enterprise Mobility Server 8.2.0 (build 15952-8.2.0)           Engine    Embedded Tomcat
10:25:32,588       0 ms      engine 8.2.0                                                           Engine    Embedded Tomcat
10:25:32,588       0 ms      beans 8.2.0.m00                                                        Engine    Embedded Tomcat
10:25:32,588       0 ms  ===================                                                        Engine    Embedded Tomcat
10:25:32,588       0 ms  Start date: Wed Oct 11 10:25:32 CEST 2023                                  Engine    Embedded Tomcat
10:25:32,589       1 ms  Class loader: com.twinsoft.convertigo.engine.util.DirClassLoader@5ac7550a  Engine    Embedded Tomcat
10:25:32,589       0 ms  Java Runtime 17.0.6 (classes: 61.0, vendor: Eclipse Adoptium)              Engine    Embedded Tomcat
10:25:32,589       0 ms  Java XML engine: Xerces-J 2.12.2                                           Engine    Embedded Tomcat
```

EDITORS PANEL

- VISUAL APP VIEWER
- CODE EDITORS
- CONNECTORS & SEQUENCES RESPONSES

LOGS & GIT PANEL

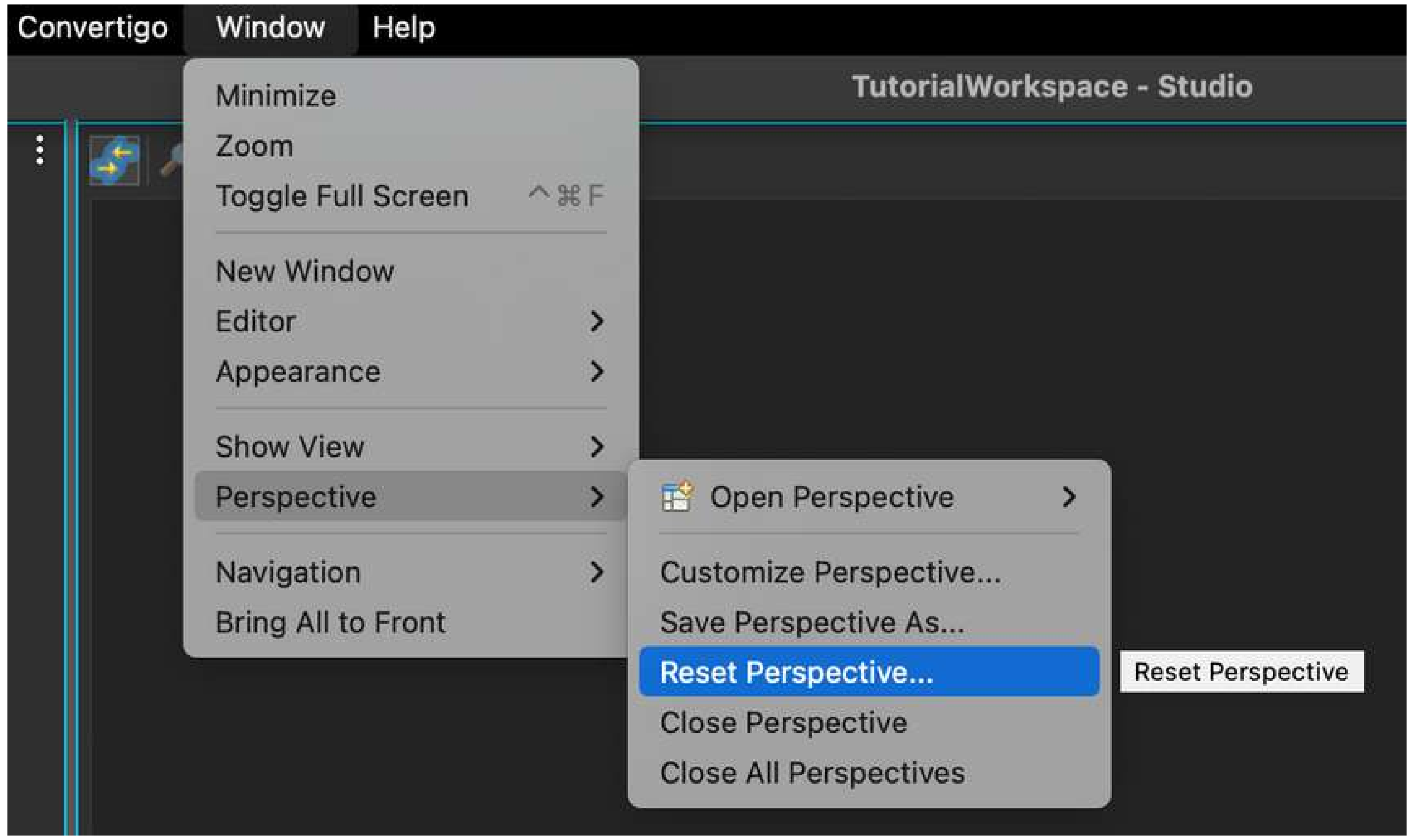
- ENGINE LOG
- CONSOLE
- SCHEMA
- GIT STAGING
- GIT REPOSITORIES

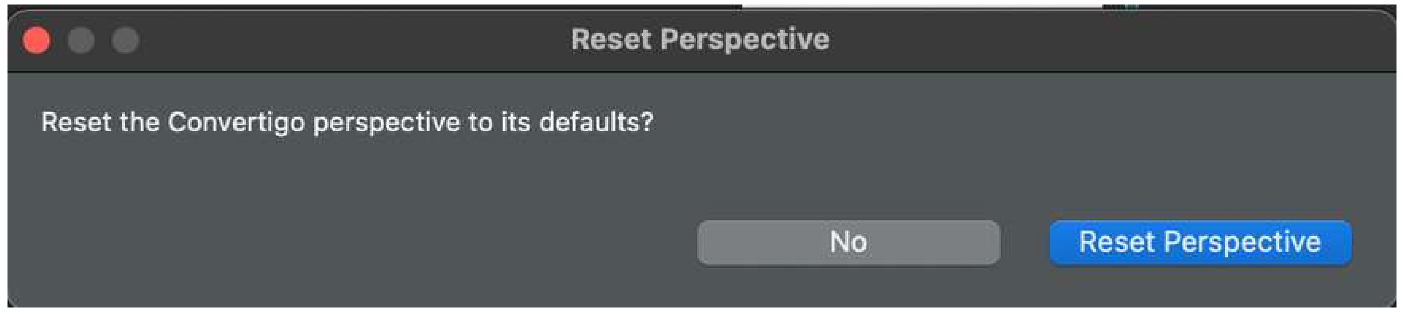
1.6 Studio Interface

The way views are organized is called a **perspective**.

Each view can be moved in other panels.

You can return to the original presentation or perspective by clicking on **Window**, then selecting **Perspective>**, then selecting **Reset Perspective**.





Reset the Convertigo perspective to its defaults?

1.7 Panels & Views

Projects Panel

PROJECTS

Displays the **projects in current workspace** and the **objects that compose them.**

PROJECT EXPLORER

Displays the projects **as files representing project assets, projects definitions as yaml** (for advanced users only)

Properties Panel

PROPERTIES

Displays the **properties of the object** selected in Projects view.

1.7 Panels & Views

Sources & Debug Panel

PALETTE

Displays all Convertigo **backend and frontend objects**.

SOURCE PICKER

Displays the **data sources for data binding** of the selected sequence step.

MOBILE DEBUG

Displays the **debugger for the front end** part.

REFERENCES

Displays **inside and outside project references** of the object selected in the Projects view.

1.7 Panels & Views

Logs & Git Panel

ENGINE LOG

Displays **Convertigo engine execution traces**.

SCHEMA

Displays the **XSD schema** used and/or generated by the project (input and output).

CONSOLE

Displays the **engine execution traces as text**.

1.7 Panels & Views

Logs & Git Panel

GIT REPOSITORIES

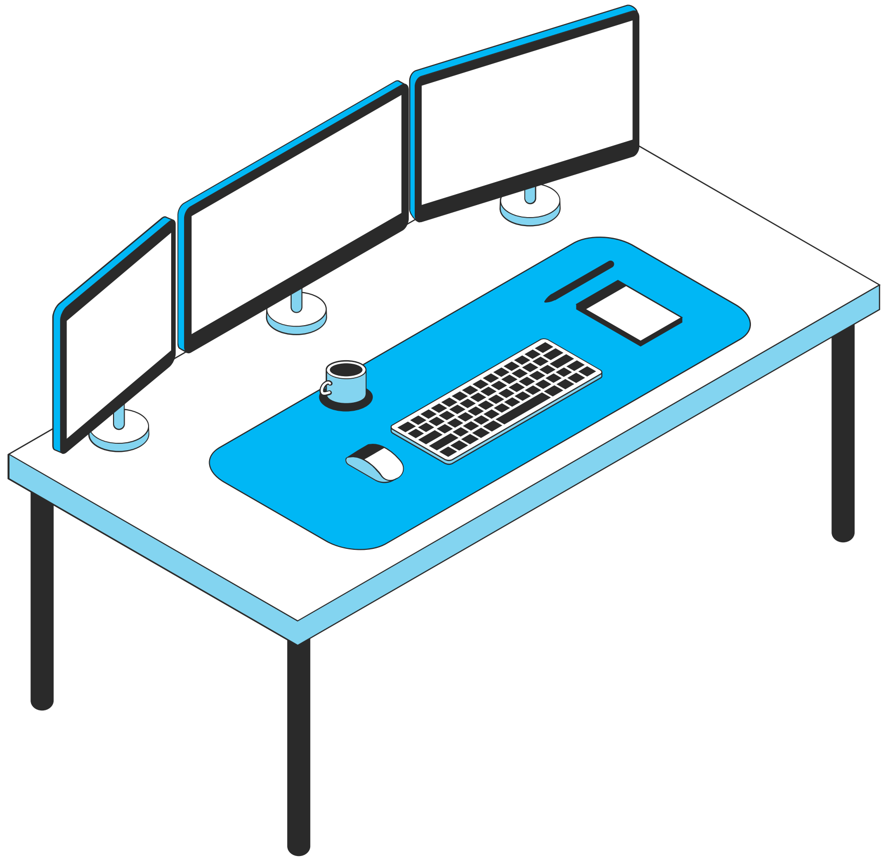
Displays the Git Repositories of the projects in your workspace.

GIT STAGING

Displays the files modified since the last commit and Git management features.

2 – Getting started

How to install and configure the studio.



2.1 Minimum System Requirements

2.2 Installation

2.3 Workspace & Convertigo archives file

2.4 Configuration

2.5 Create a project

2.6 Export a project

2.7 Import a project

2.1 Minimum system requirements

The following minimum system requirements are necessary for installing the studio.

WINDOWS

- Windows 10
- Windows 11

LINUX

- Ubuntu
  - version 20.04 (LTS)
  - version 22.04 (LTS)
- Debian version 11.0

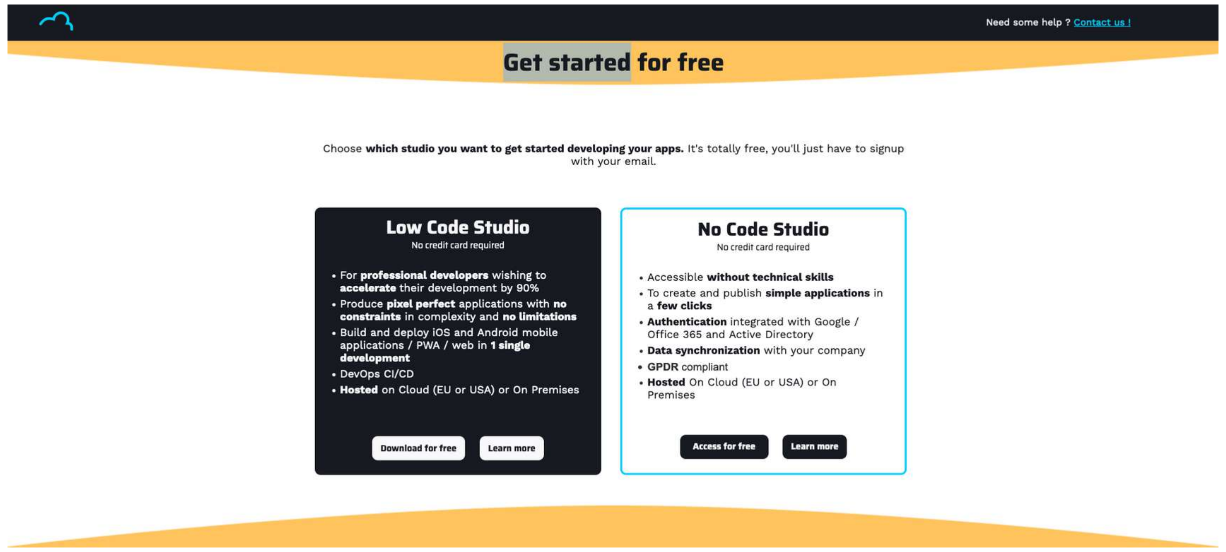
MAC OS

- Mac OS X
  - 10.5 (Leopard)
  - or greater
- Mac ARM

2.2 Installation

Go to the **Get started page** on **https://www.convertigo.com/get-started-page**.

Download the Low Code Studio package file for your operating system (Windows, Linux or Mac OS).



Get started for free

Choose **which studio you want to get started developing your apps.** It's totally free, you'll just have to signup with your email.

Low Code Studio

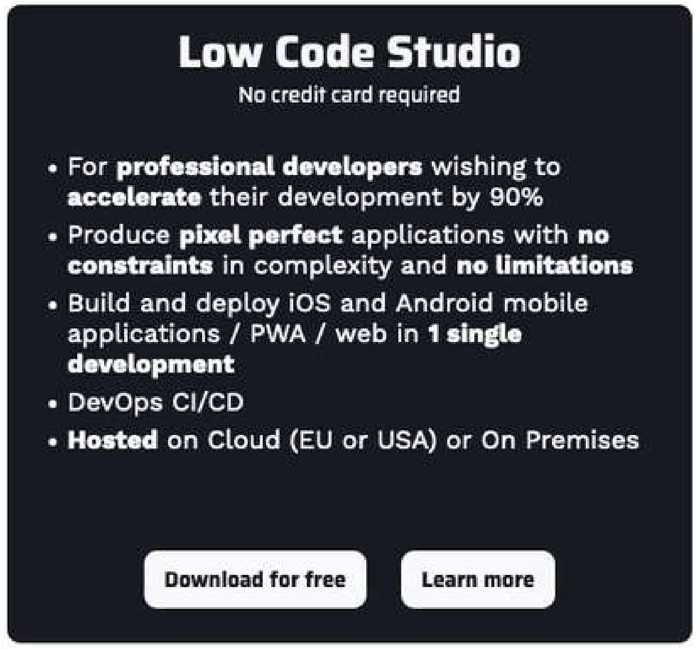
No credit card required

- For **professional developers** wishing to **accelerate** their development by 90%
- Produce **pixel perfect** applications with **no constraints** in complexity and **no limitations**
- Build and deploy iOS and Android mobile applications / PWA / web in **1 single development**
- DevOps CI/CD
- **Hosted** on Cloud (EU or USA) or On Premises

No Code Studio

No credit card required

- Accessible **without technical skills**
- To create and publish **simple applications** in a **few clicks**
- **Authentication** integrated with Google / Office 365 and Active Directory
- **Data synchronization** with your company
- **GPDR** compliant
- **Hosted** On Cloud (EU or USA) or On Premises



Low Code Studio

No credit card required

- For **professional developers** wishing to **accelerate** their development by 90%
- Produce **pixel perfect** applications with **no constraints** in complexity and **no limitations**
- Build and deploy iOS and Android mobile applications / PWA / web in **1 single development**
- DevOps CI/CD
- **Hosted** on Cloud (EU or USA) or On Premises

Open the package file and install the studio in a destination directory where you have the rights to.

The installation package contains

- the Eclipse-based Convertigo Studio
- the embedded Convertigo Server with an Apache Tomcat application server

2.3 Workspace & Convertigo archives file

On first launch, you need to create a workspace for your projects.

A workspace is a directory where are saved

- Studio **configurations**
- Convertigo **projects**
- **Execution logs**

The workspace is **located outside** of the installation directory

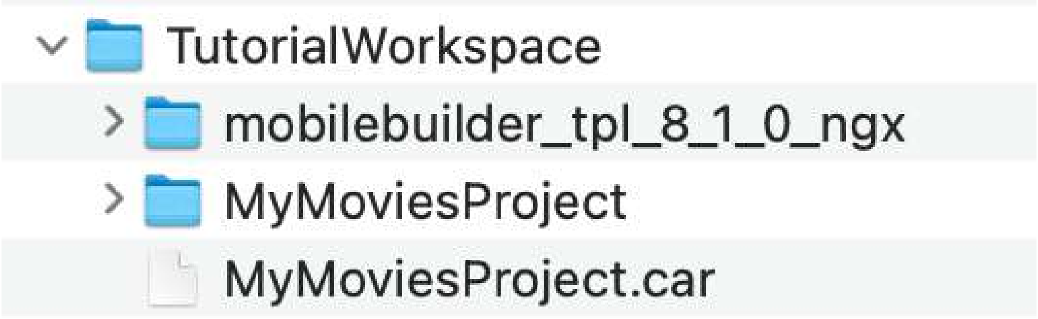
to **save your data**

if you need to **uninstall** or **re-install** the studio.

In Convertigo, the **import/export format** is **.car (Convertigo archives)** or **.zip**.

The .car is a zip file that contains all your project.

Example : A workspace with a Convertigo project and a .car file



```
TutorialWorkspace
  mobilebuilder_tpl_8_1_0_ngx
  MyMoviesProject
  MyMoviesProject.car
```

2.4 Configuration

After installation, the Studio needs to be configured on first launch.

CONFIGURATION PROCESS

Step 1	Select a directory as workspace
Step 2	Accept License
Step 3	Complete the workspace creation
Step 4	Configure proxy settings (Optional)
Step 5	Register with Convertigo Cloud Trial
Step 6	Welcome to Convertigo Low Code Studio

2.4 Configuration

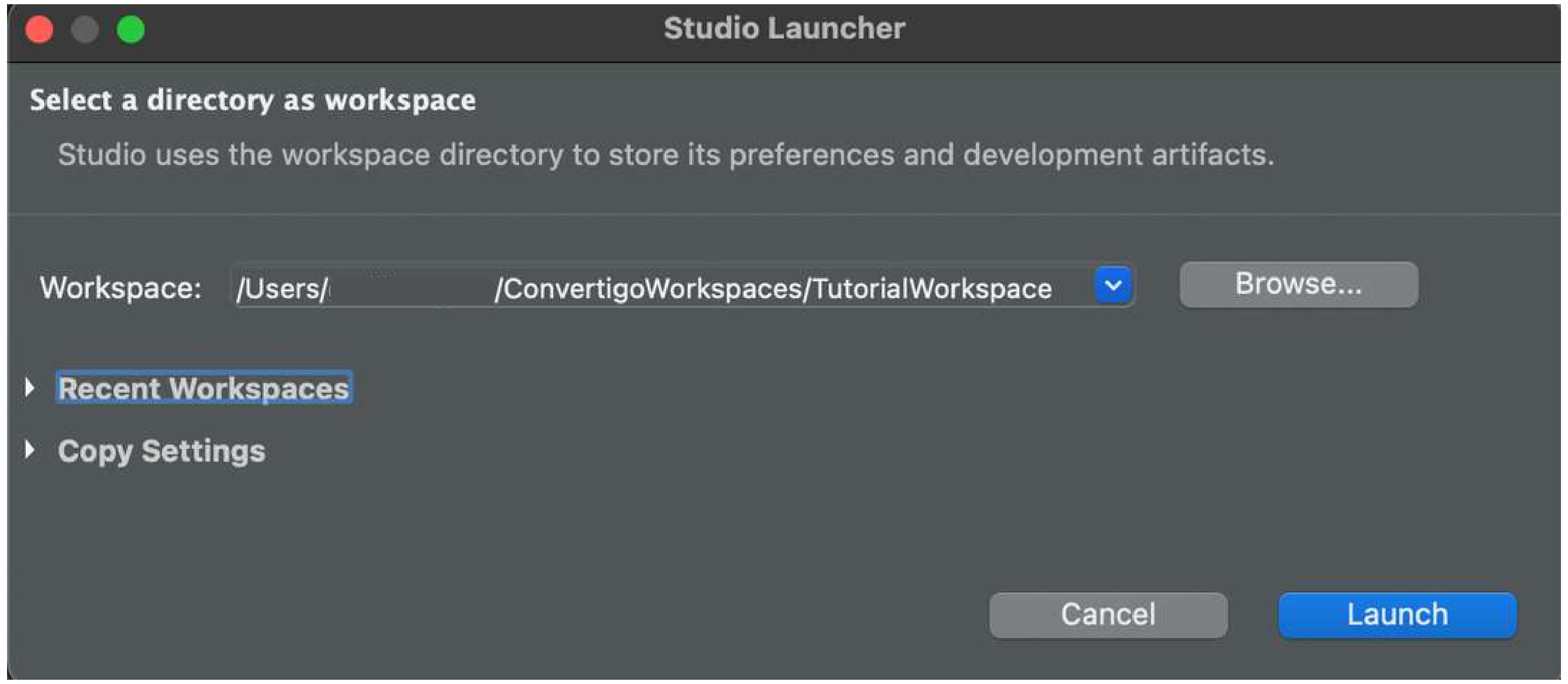
Step 1 – Select a directory as workspace

This is the first time we are going to launch the studio.
Let’s start by creating a **workspace** for your projects.

Launch the studio
and **select a folder**
where your **workspace will be created**.

You can

- select an **existing** folder or **create** one.
- create **as many workspaces** as you want
- **wherever you want** on your computer.



Select a directory as workspace

Studio uses the workspace directory to store its preferences and development artifacts.

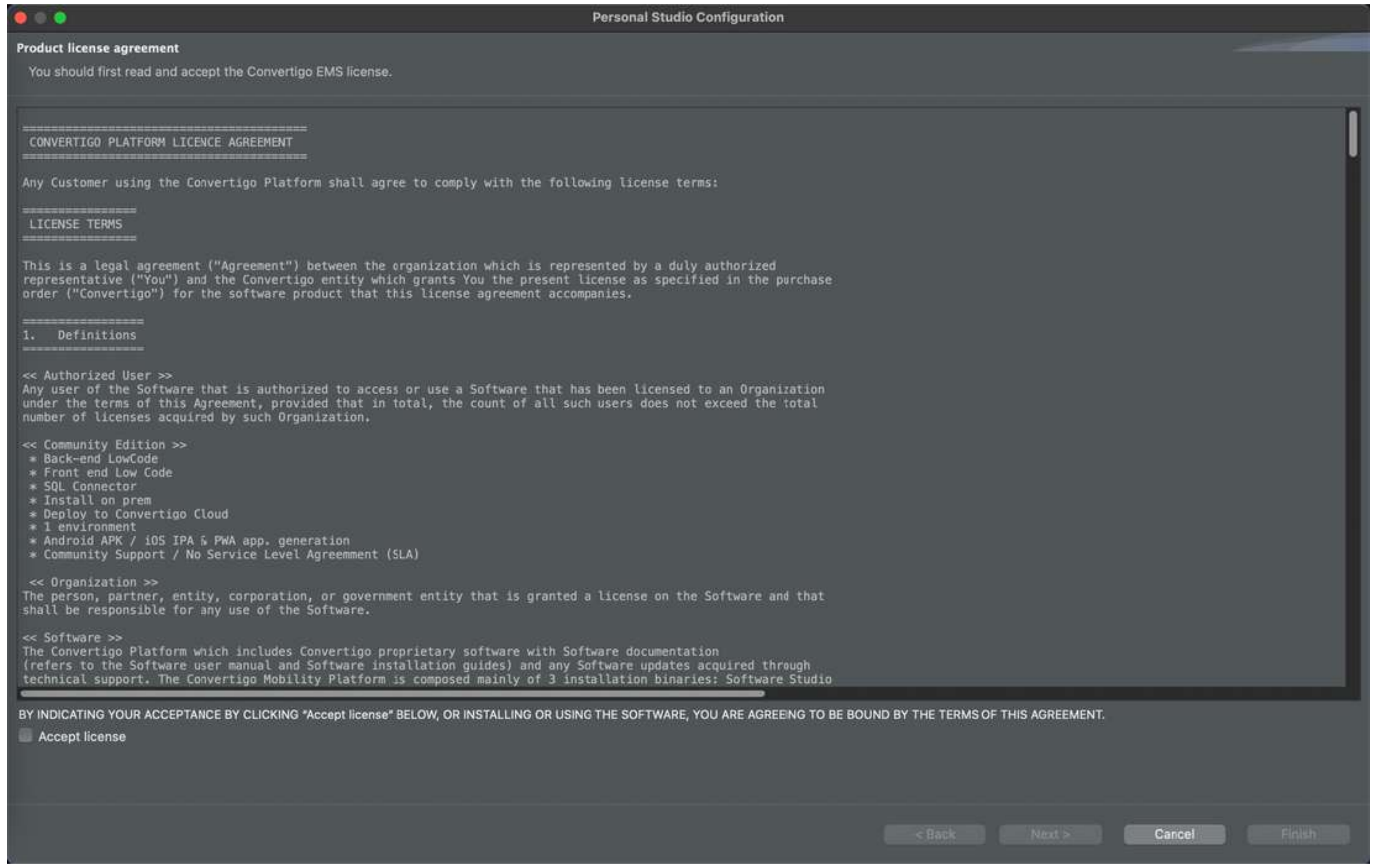
Workspace: /Users/[redacted]/ConvertigoWorkspaces/TutorialWorkspace

Good practice: create your workspace in your user folder on the same level as the Desktop and the Download folder – BUT NOT INSIDE THEM.

2.4 Configuration

Step 2 – Accept License

In the window **Personal studio Configuration**, **Accept License** and click on **Next >**.



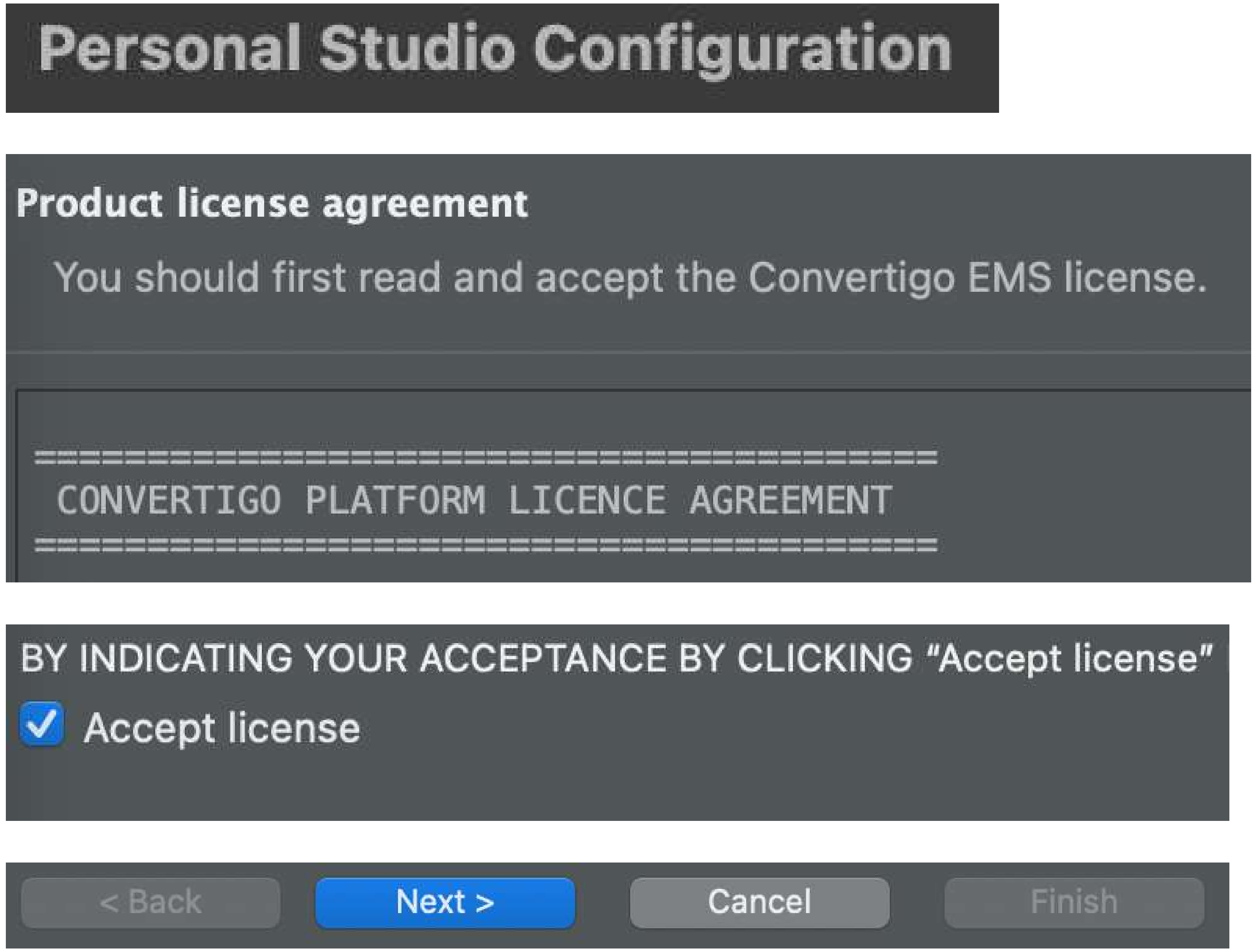
Product license agreement

You should first read and accept the Convertigo EMS license.

```
=========================================
 CONVERTIGO PLATFORM LICENCE AGREEMENT
=========================================

Any Customer using the Convertigo Platform shall agree to comply with the following license terms:

================
 LICENSE TERMS
================

This is a legal agreement ("Agreement") between the organization which is represented by a duly authorized
representative ("You") and the Convertigo entity which grants You the present license as specified in the purchase
order ("Convertigo") for the software product that this license agreement accompanies.

=================
1.   Definitions
=================

<< Authorized User >>
Any user of the Software that is authorized to access or use a Software that has been licensed to an Organization
under the terms of this Agreement, provided that in total, the count of all such users does not exceed the total
number of licenses acquired by such Organization.

<< Community Edition >>
 * Back-end LowCode
 * Front end Low Code
 * SQL Connector
 * Install on prem
 * Deploy to Convertigo Cloud
 * 1 environment
 * Android APK / iOS IPA & PWA app. generation
 * Community Support / No Service Level Agreement (SLA)

 << Organization >>
The person, partner, entity, corporation, or government entity that is granted a license on the Software and that
shall be responsible for any use of the Software.

<< Software >>
The Convertigo Platform which includes Convertigo proprietary software with Software documentation
(refers to the Software user manual and Software installation guides) and any Software updates acquired through
technical support. The Convertigo Mobility Platform is composed mainly of 3 installation binaries: Software Studio
```

BY INDICATING YOUR ACCEPTANCE BY CLICKING "Accept license" BELOW, OR INSTALLING OR USING THE SOFTWARE, YOU ARE AGREEING TO BE BOUND BY THE TERMS OF THIS AGREEMENT.

□ Accept license



Product license agreement

You should first read and accept the Convertigo EMS license.

```
=========================================
 CONVERTIGO PLATFORM LICENCE AGREEMENT
=========================================
```

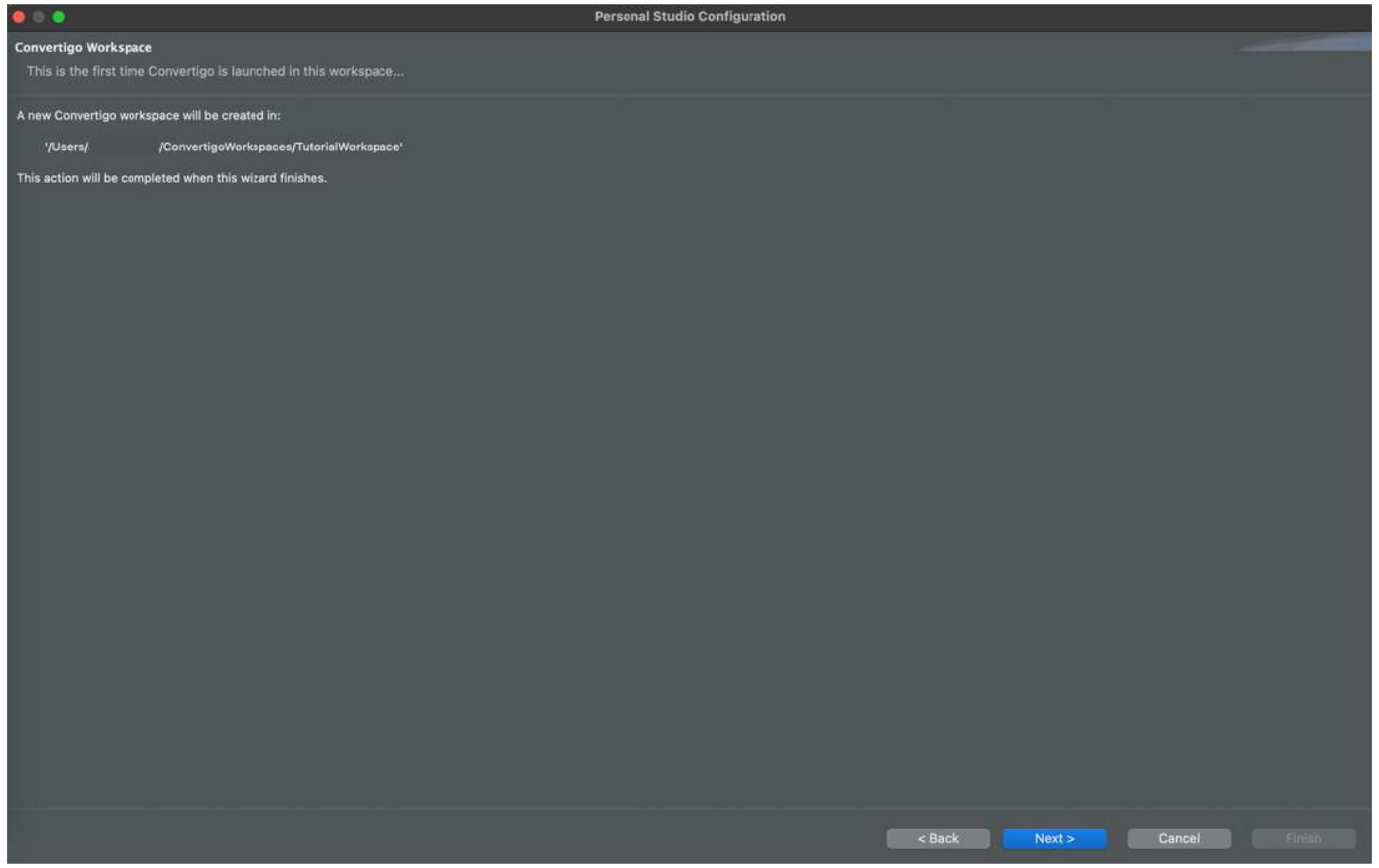
BY INDICATING YOUR ACCEPTANCE BY CLICKING "Accept license"

☑ Accept license

2.4 Configuration

Step 3 – Complete the workspace creation

To complete the creation of your workspace, click on **Next >**



Personal Studio Configuration

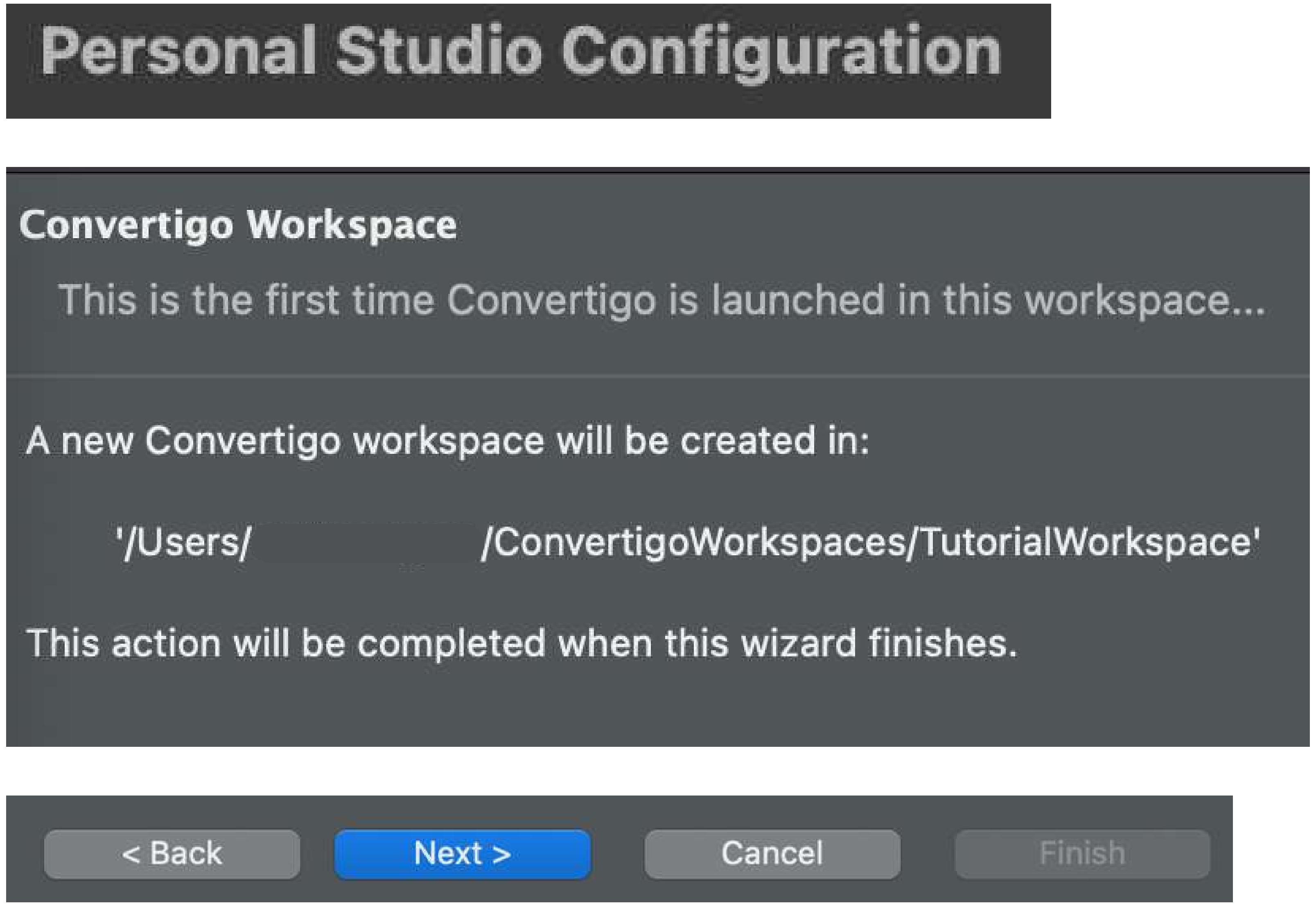
Convertigo Workspace

This is the first time Convertigo is launched in this workspace...

A new Convertigo workspace will be created in:

'/Users/ /ConvertigoWorkspaces/TutorialWorkspace'

This action will be completed when this wizard finishes.



Personal Studio Configuration

Convertigo Workspace

This is the first time Convertigo is launched in this workspace...

A new Convertigo workspace will be created in:

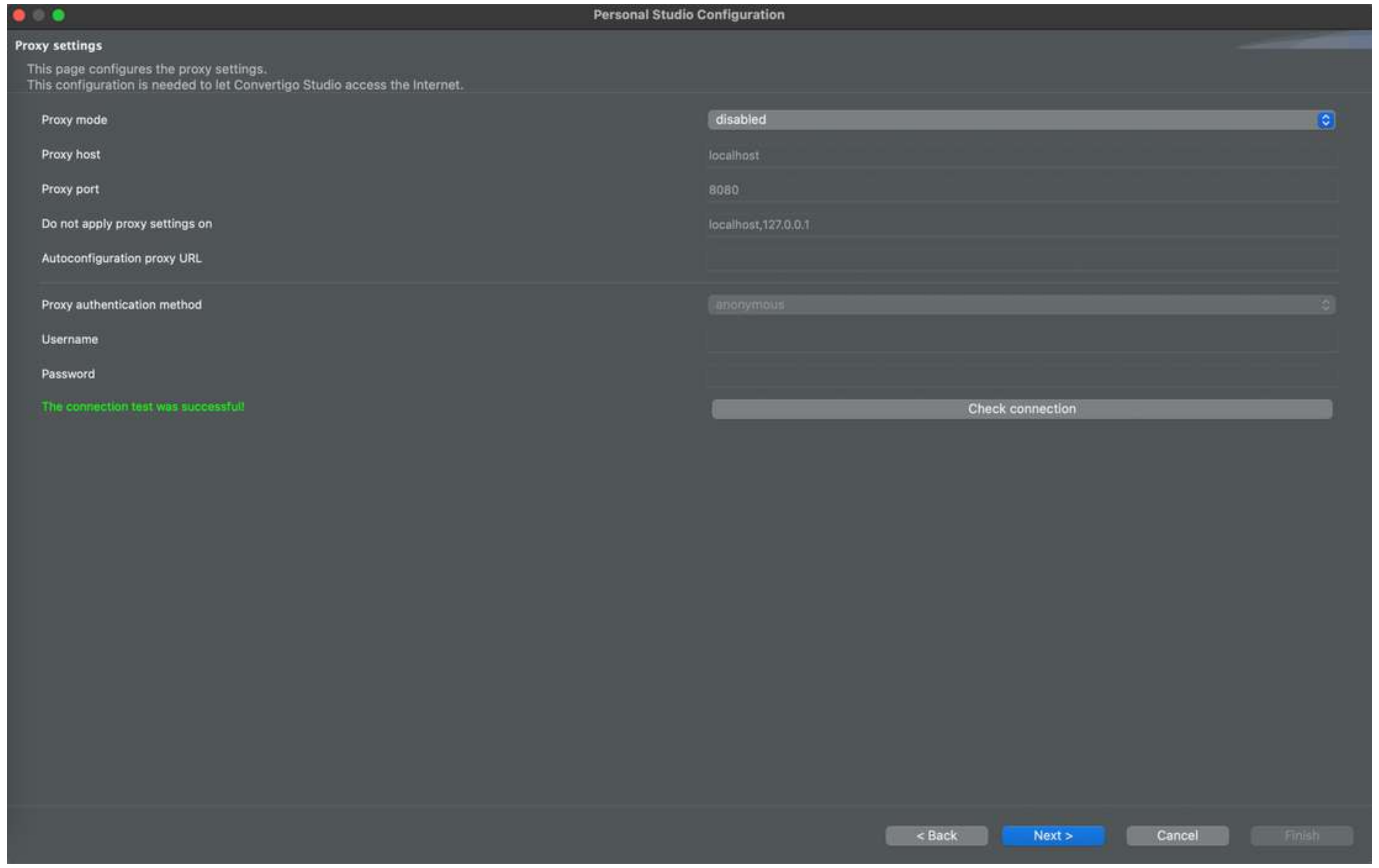
'/Users/ /ConvertigoWorkspaces/TutorialWorkspace'

This action will be completed when this wizard finishes.

2.4 Configuration

Step 4 – Configure proxy settings

Optional : You can **configure proxy settings** for Convertigo Studio to access the Internet



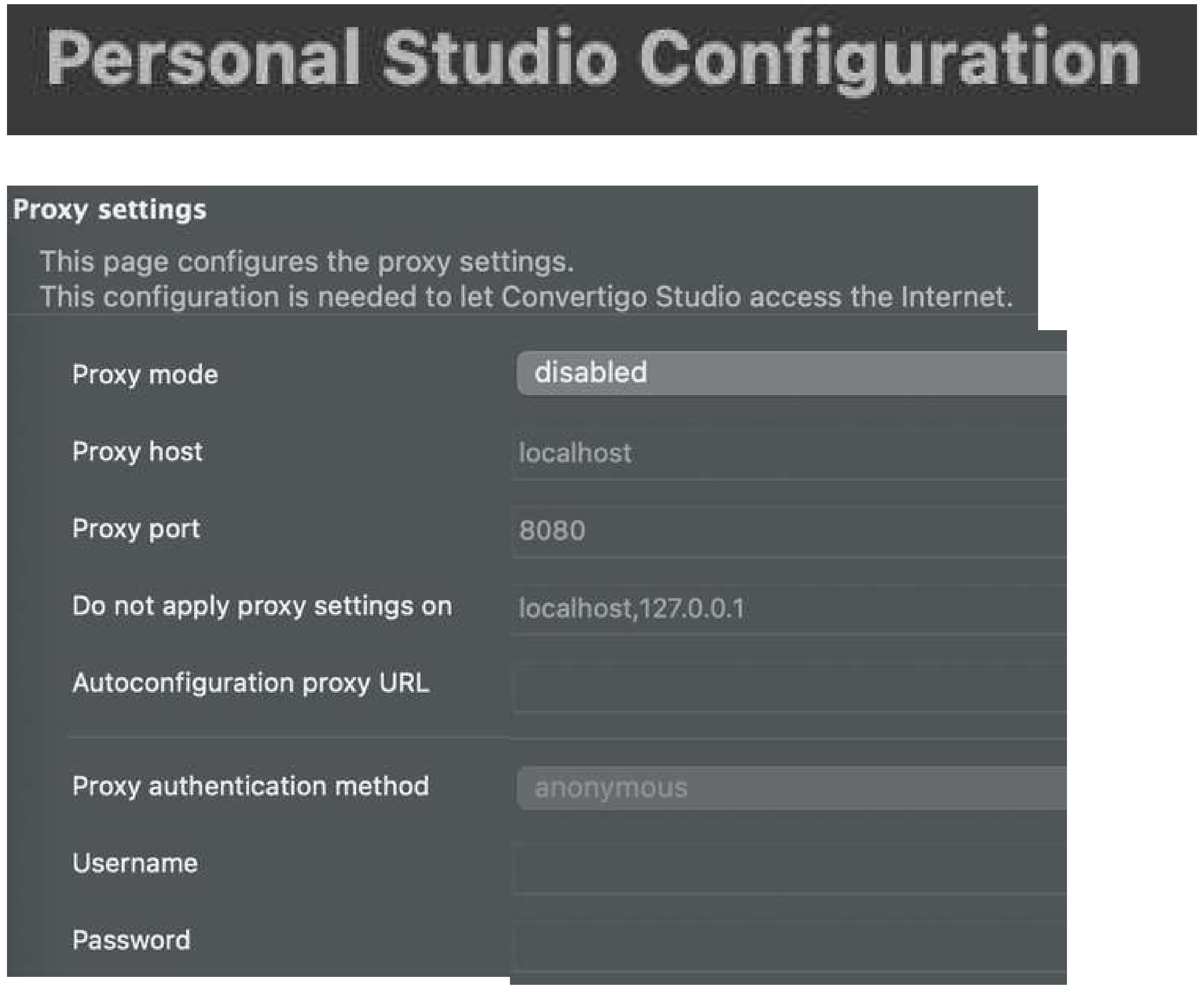
Personal Studio Configuration

Proxy settings

This page configures the proxy settings.
This configuration is needed to let Convertigo Studio access the Internet.

Proxy mode: disabled
Proxy host: localhost
Proxy port: 8080
Do not apply proxy settings on: localhost,127.0.0.1
Autoconfiguration proxy URL:
Proxy authentication method: anonymous
Username:
Password:

The connection test was successful!



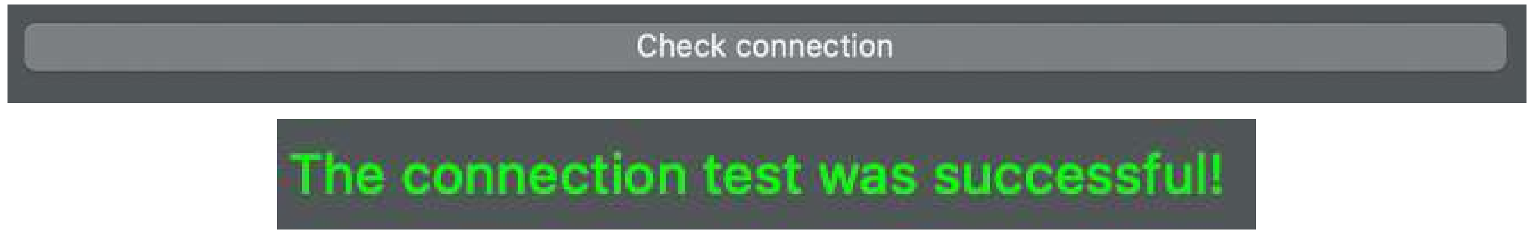
Personal Studio Configuration

Proxy settings

This page configures the proxy settings.
This configuration is needed to let Convertigo Studio access the Internet.

Proxy mode: disabled
Proxy host: localhost
Proxy port: 8080
Do not apply proxy settings on: localhost,127.0.0.1
Autoconfiguration proxy URL:
Proxy authentication method: anonymous
Username:
Password:

To check the connection, click on **Check connection**

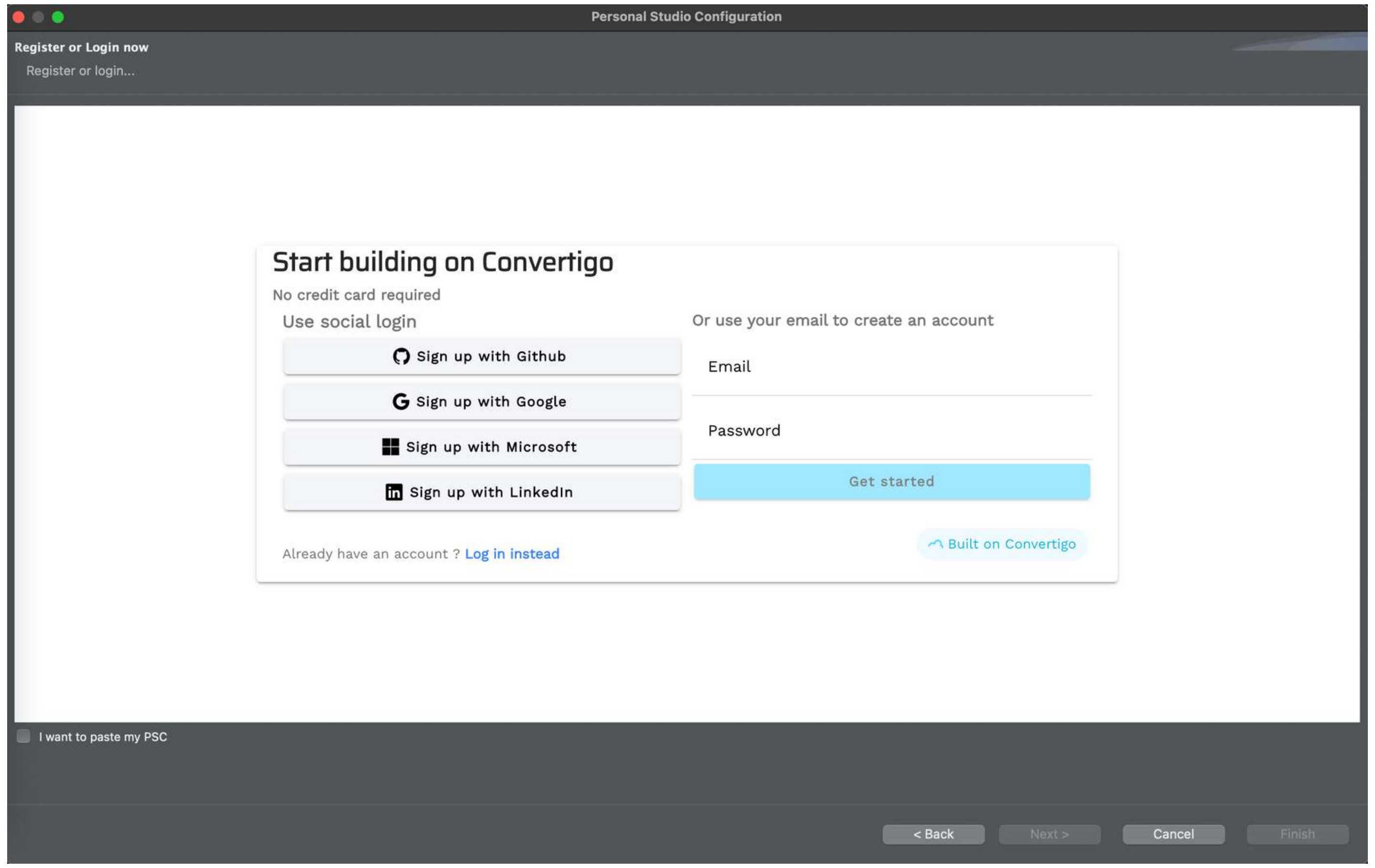


The connection test was successful!

2.4 Configuration

Step 5 – Register with Convertigo Cloud Trial

Complete your registration with **Convertigo Cloud Trial**
by entering your **email** and a **password**, or using a **Credential provider**.
It will create your **account** on **Convertigo Cloud Trial**.



Personal Studio Configuration

Register or Login now

Register or login...

Start building on Convertigo

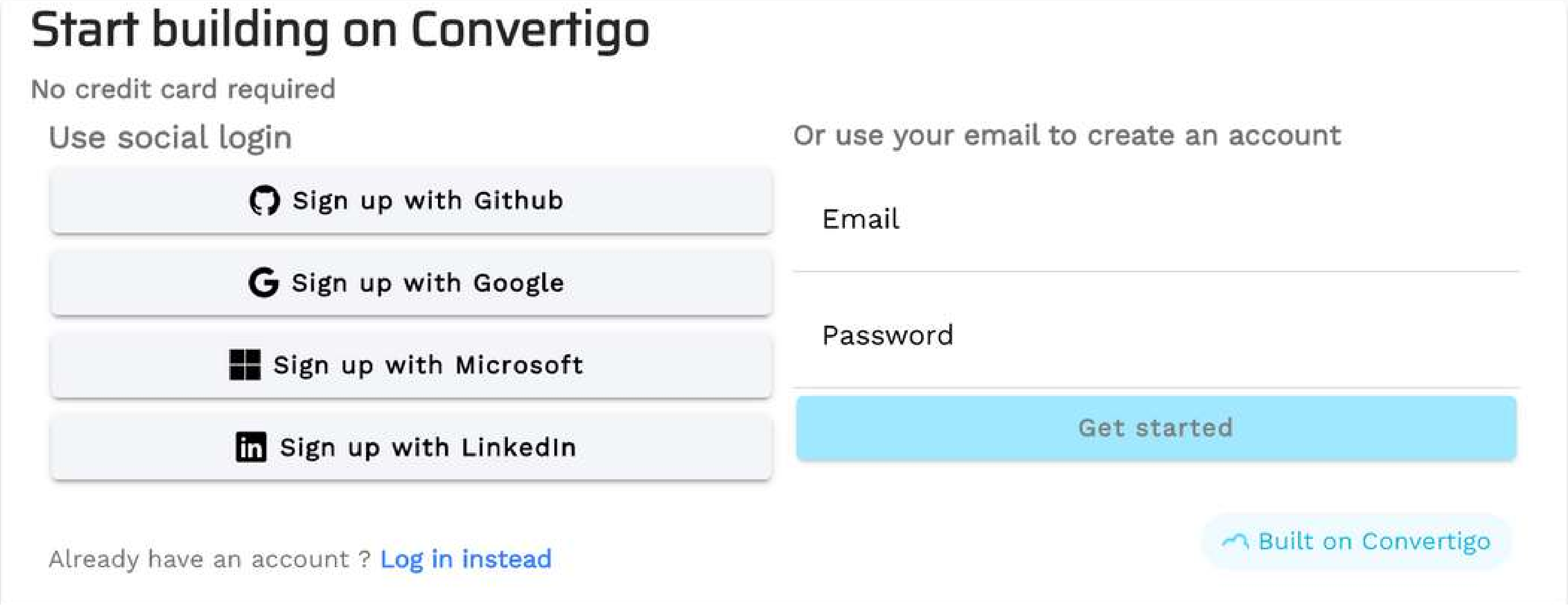
No credit card required

Use social login

Or use your email to create an account

Already have an account ? Log in instead

□ I want to paste my PSC



Start building on Convertigo

No credit card required

Use social login

Or use your email to create an account

Already have an account ? Log in instead

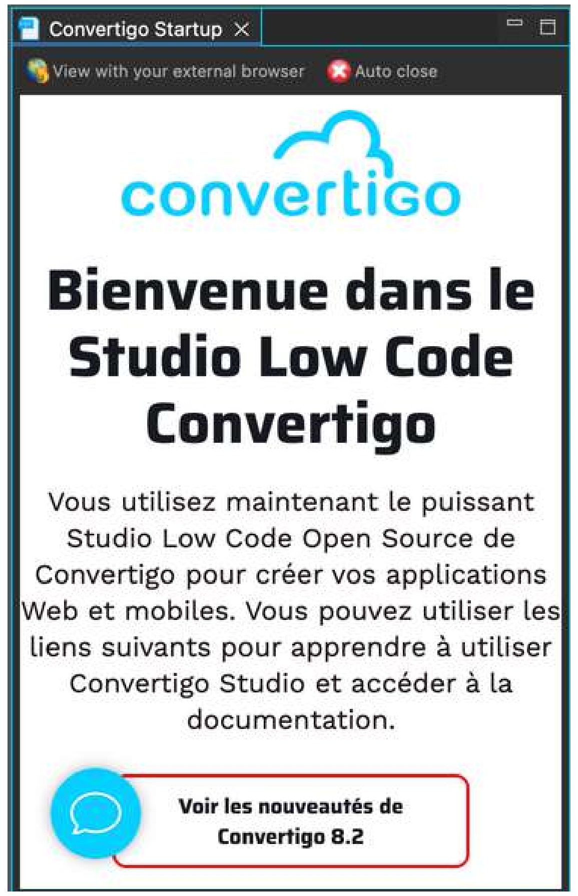
If you **create other workspaces**,
you will just need to **log in to this account**.

2.4 Configuration

Step 6 – Welcome to Convertigo Low Code Studio

On first lauch, you will find 2 additionnal views :

The **Convertigo Startup view** with a link to **Convertigo's website** and the studio's **documentation**.

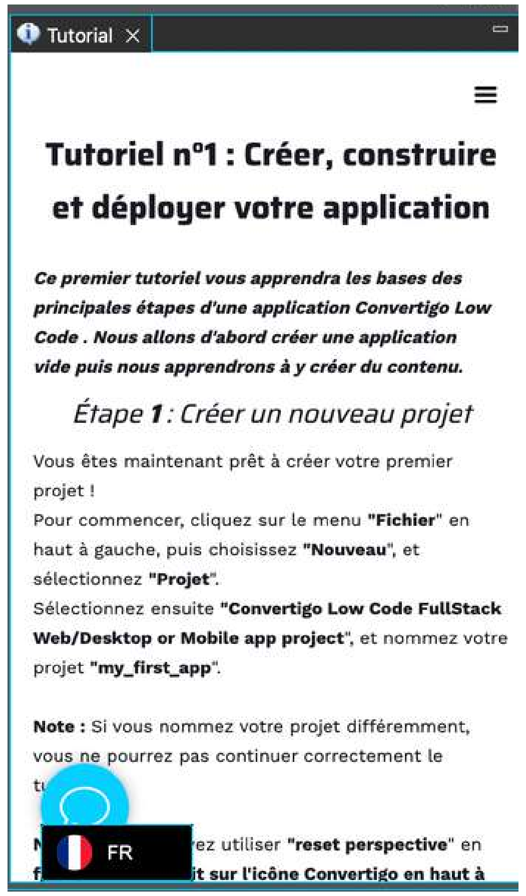


Bienvenue dans le Studio Low Code Convertigo

Vous utilisez maintenant le puissant Studio Low Code Open Source de Convertigo pour créer vos applications Web et mobiles. Vous pouvez utiliser les liens suivants pour apprendre à utiliser Convertigo Studio et accéder à la documentation.

Voir les nouveautés de Convertigo 8.2

The **Tutorial view** featuring **exercises** to help you **getting started** with the studio.



Tutoriel n°1 : Créer, construire et déployer votre application

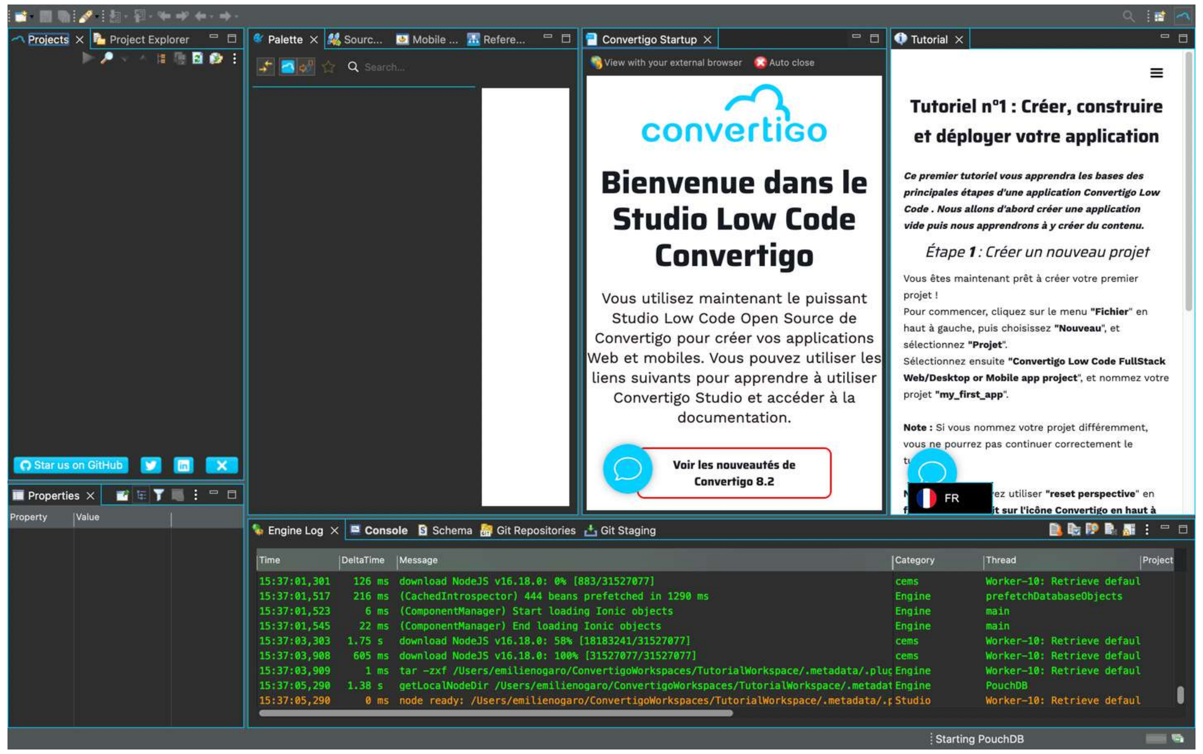
Ce premier tutoriel vous apprendra les bases des principales étapes d'une application Convertigo Low Code . Nous allons d'abord créer une application vide puis nous apprendrons à y créer du contenu.

Étape 1 : Créer un nouveau projet

Vous êtes maintenant prêt à créer votre premier projet !
Pour commencer, cliquez sur le menu **"Fichier"** en haut à gauche, puis choisissez **"Nouveau"**, et sélectionnez **"Projet"**.
Sélectionnez ensuite **"Convertigo Low Code FullStack Web/Desktop or Mobile app project"**, et nommez votre projet **"my_first_app"**.

Note : Si vous nommez votre projet différemment, vous ne pourrez pas continuer correctement le tu

N ez utiliser **"reset perspective"** en f **it sur l'icône Convertigo en haut à**



```
Time          DeltaTime  Message                                                                        Category  Thread                       Project
15:37:01,301   126 ms   download NodeJS v16.18.0: 0% [883/31527077]                                     cems      Worker-10: Retrieve defaul
15:37:01,517   216 ms   (CachedIntrospector) 444 beans prefetched in 1290 ms                            Engine    prefetchDatabaseObjects
15:37:01,523     6 ms   (ComponentManager) Start loading Ionic objects                                  Engine    main
15:37:01,545    22 ms   (ComponentManager) End loading Ionic objects                                    Engine    main
15:37:03,303   1.75 s   download NodeJS v16.18.0: 58% [18183241/31527077]                               cems      Worker-10: Retrieve defaul
15:37:03,908   605 ms   download NodeJS v16.18.0: 100% [31527077/31527077]                              cems      Worker-10: Retrieve defaul
15:37:03,909     1 ms   tar -zxf /Users/emilienogaro/ConvertigoWorkspaces/TutorialWorkspace/.metadata/.plug Engine  Worker-10: Retrieve defaul
15:37:05,290   1.38 s   getLocalNodeDir /Users/emilienogaro/ConvertigoWorkspaces/TutorialWorkspace/.metadat Engine  PouchDB
15:37:05,290     0 ms   node ready: /Users/emilienogaro/ConvertigoWorkspaces/TutorialWorkspace/.metadata/.p Studio  Worker-10: Retrieve defaul
Starting PouchDB
```

2.5 Create a project

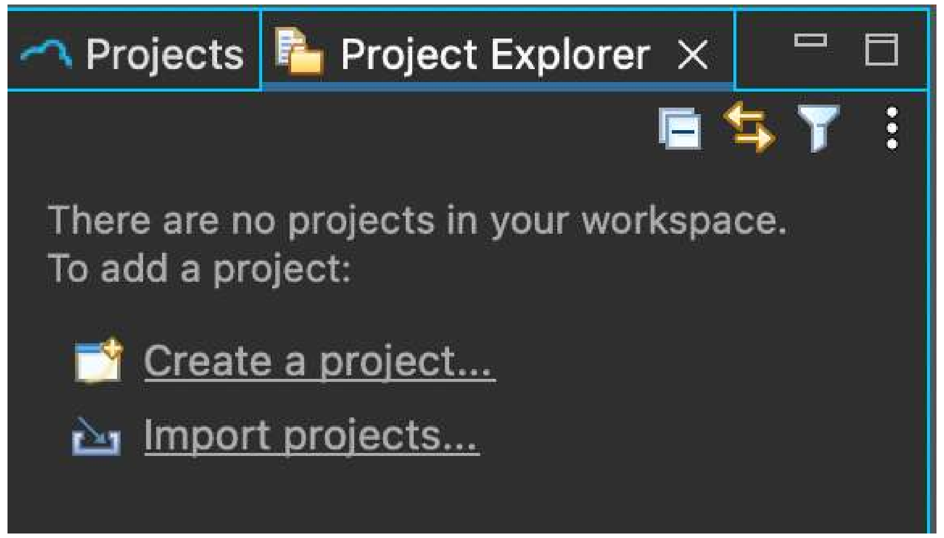
There are several ways to create a project in Convertigo.

When you create a project **for the first time** in a **new workspace**, you can :

First option :

click on **Create a project**

in the **Project Explorer view**



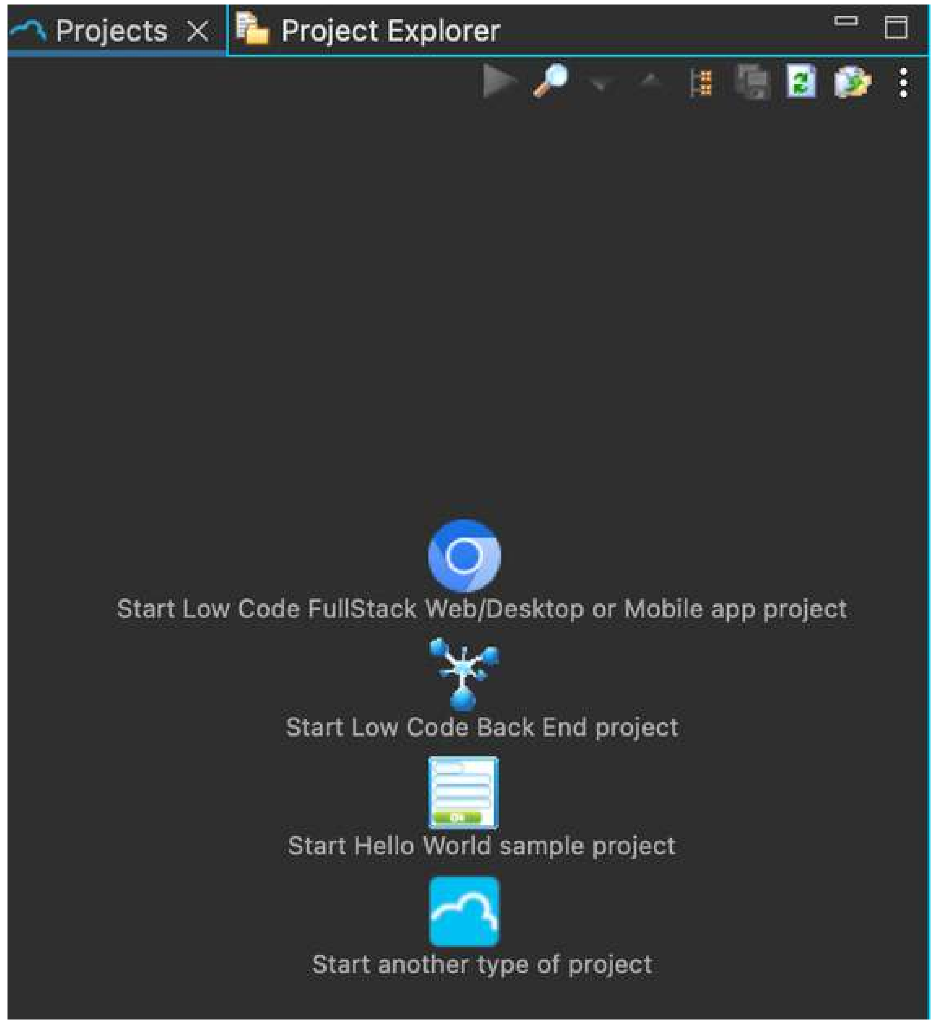
There are no projects in your workspace.
To add a project:

Second option :

click on

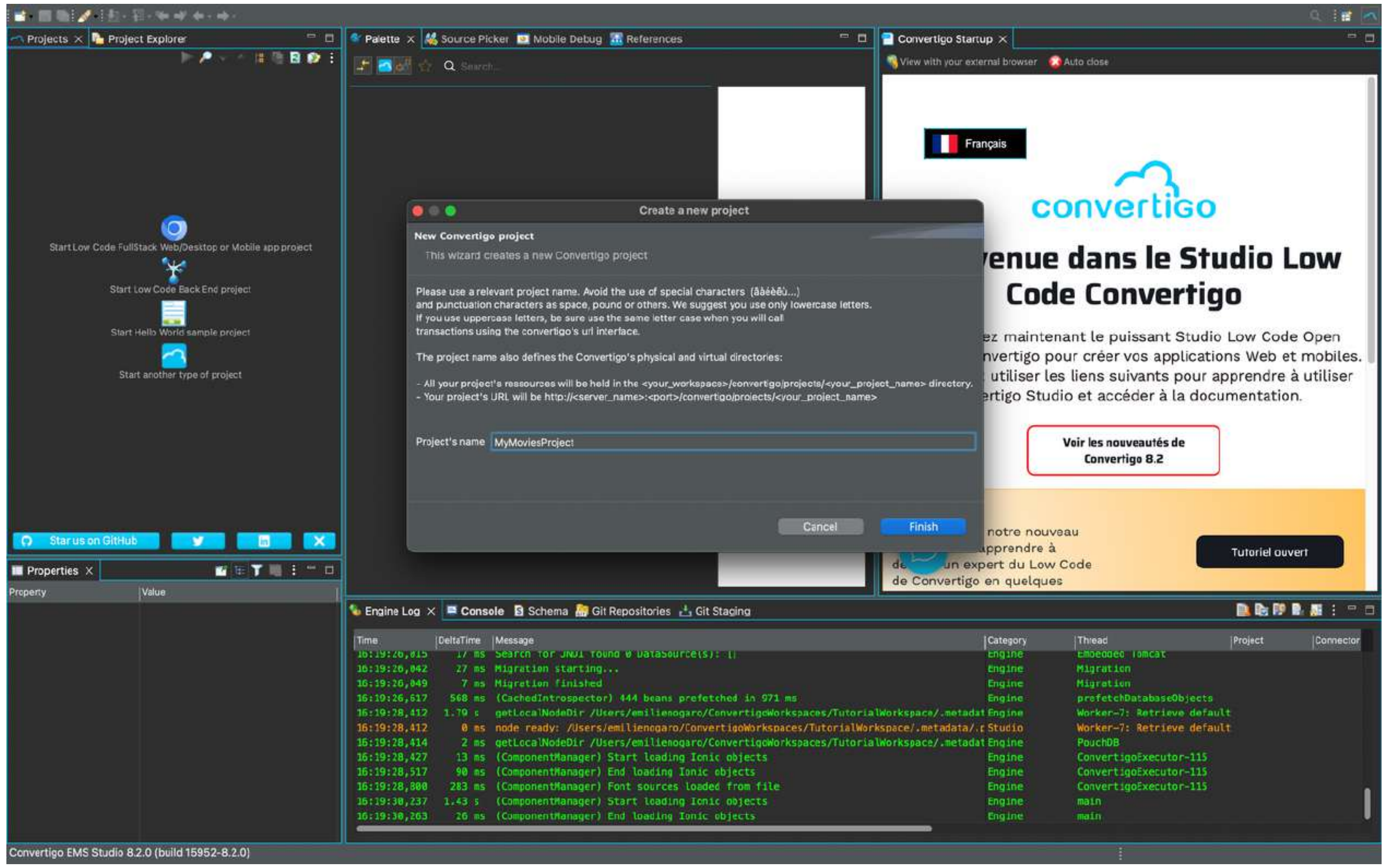
Start Low Code Fullstack Web/Desktop or Mobile app project

in the **Project view**


Projects
Project Explorer
Start Low Code FullStack Web/Desktop or Mobile app project
Start Low Code Back End project
Start Hello World sample project
Start another type of project

2.5 Create a project

The **Create a new project windows** appears.



Projects
Project Explorer
Palette
Source Picker
Mobile Debug
References
Convertigo Startup
View with your external browser
Auto close
Start Low Code FullStack Web/Desktop or Mobile app project
Start Low Code Back End project
Start Hello World sample project
Start another type of project
Star us on GitHub
Properties
Property
Value
Français

convertigo

…venue dans le Studio Low Code Convertigo

…ez maintenant le puissant Studio Low Code Open …nvertigo pour créer vos applications Web et mobiles. … utiliser les liens suivants pour apprendre à utiliser …ertigo Studio et accéder à la documentation.

Voir les nouveautés de Convertigo 8.2

… notre nouveau … apprendre à de… un expert du Low Code de Convertigo en quelques

Tutoriel ouvert

Create a new project

New Convertigo project

This wizard creates a new Convertigo project

Please use a relevant project name. Avoid the use of special characters (âàéèêù...)
and punctuation characters as space, pound or others. We suggest you use only lowercase letters.
If you use uppercase letters, be sure use the same letter case when you will call
transactions using the convertigo's url interface.

The project name also defines the Convertigo's physical and virtual directories:

- All your project's ressources will be held in the <your_workspace>/convertigo/projects/<your_project_name> directory.
- Your project's URL will be http://<server_name>:<port>/convertigo/projects/<your_project_name>

Project's name: MyMoviesProject

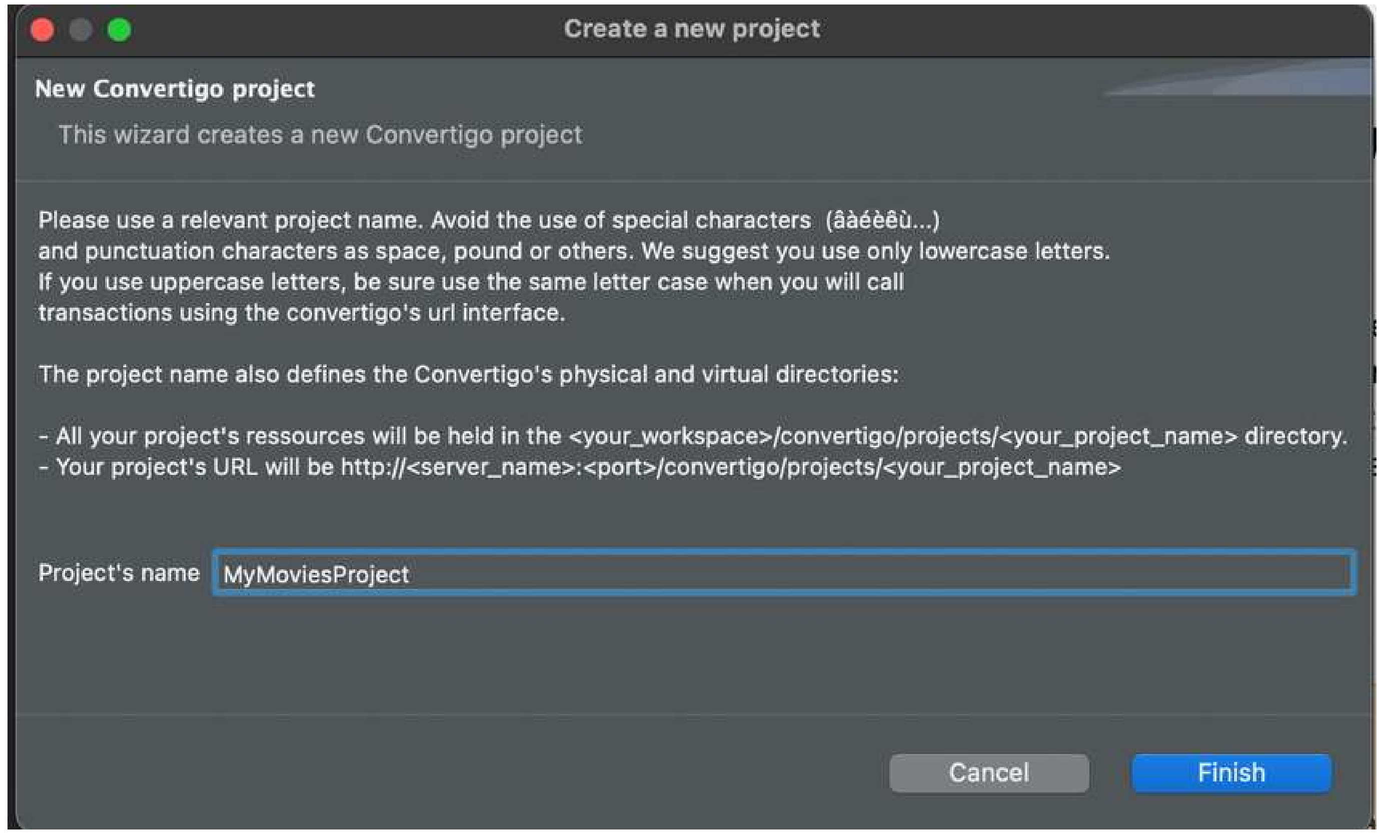
Cancel | Finish

Engine Log | Console | Schema | Git Repositories | Git Staging

```
Time          DeltaTime  Message                                                              Category  Thread                       Project  Connector
16:19:26,015     17 ms   Search for JNDI found 0 DataSource(s): []                            Engine    Embedded Tomcat
16:19:26,042     27 ms   Migration starting...                                                Engine    Migration
16:19:26,049      7 ms   Migration finished                                                   Engine    Migration
16:19:26,617    568 ms   (CachedIntrospector) 444 beans prefetched in 971 ms                  Engine    prefetchDatabaseObjects
16:19:28,412    1.79 s   getLocalNodeDir /Users/emilienogaro/ConvertigoWorkspaces/TutorialWorkspace/.metadat Engine  Worker-7: Retrieve default
16:19:28,412      0 ms   node ready: /Users/emilienogaro/ConvertigoWorkspaces/TutorialWorkspace/.metadata/.( Studio  Worker-7: Retrieve default
16:19:28,414      2 ms   getLocalNodeDir /Users/emilienogaro/ConvertigoWorkspaces/TutorialWorkspace/.metadat Engine  PouchDB
16:19:28,427     13 ms   (ComponentManager) Start loading Ionic objects                       Engine    ConvertigoExecutor-115
16:19:28,517     90 ms   (ComponentManager) End loading Ionic objects                         Engine    ConvertigoExecutor-115
16:19:28,800    283 ms   (ComponentManager) Font sources loaded from file                     Engine    ConvertigoExecutor-115
16:19:30,237    1.43 s   (ComponentManager) Start loading Ionic objects                       Engine    main
16:19:30,263     26 ms   (ComponentManager) End loading Ionic objects                         Engine    main
```

Convertigo EMS Studio 8.2.0 (build 15952-8.2.0)

Enter a project name, then click on **Finish**.



Create a new project

New Convertigo project

This wizard creates a new Convertigo project

Please use a relevant project name. Avoid the use of special characters (âàéèêù...)
and punctuation characters as space, pound or others. We suggest you use only lowercase letters.
If you use uppercase letters, be sure use the same letter case when you will call
transactions using the convertigo's url interface.

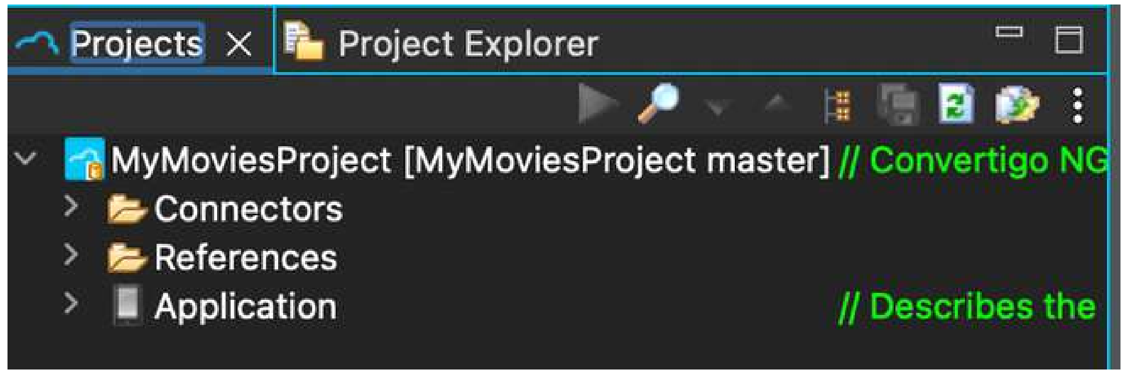
The project name also defines the Convertigo's physical and virtual directories:

- All your project's ressources will be held in the <your_workspace>/convertigo/projects/<your_project_name> directory.
- Your project's URL will be http://<server_name>:<port>/convertigo/projects/<your_project_name>

Project's name: MyMoviesProject

Cancel | Finish

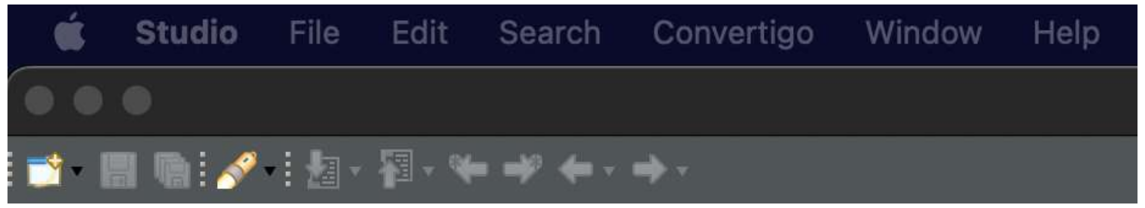
The project appears in the **Project view**.

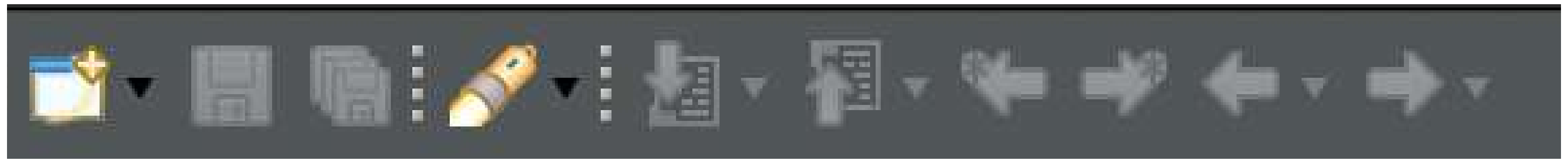


```
MyMoviesProject [MyMoviesProject master] // Convertigo NG
  > Connectors
  > References
  > Application                               // Describes the
```

2.5 Create a project

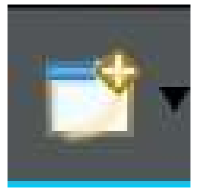
Another way to create a project is to use the toolbar in the **Project view**.





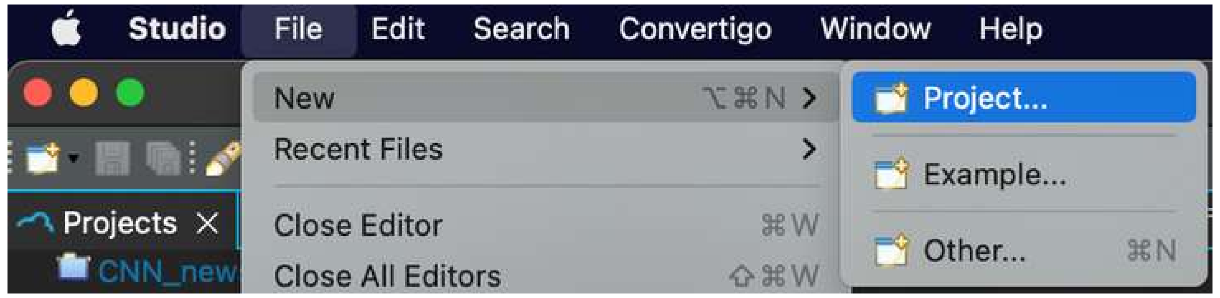
First option

Click directly on this icon.



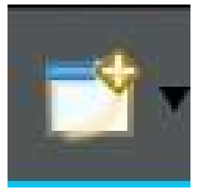
Click on **New›**

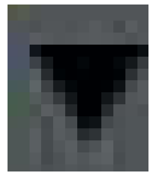
then select **Project**.



Second option

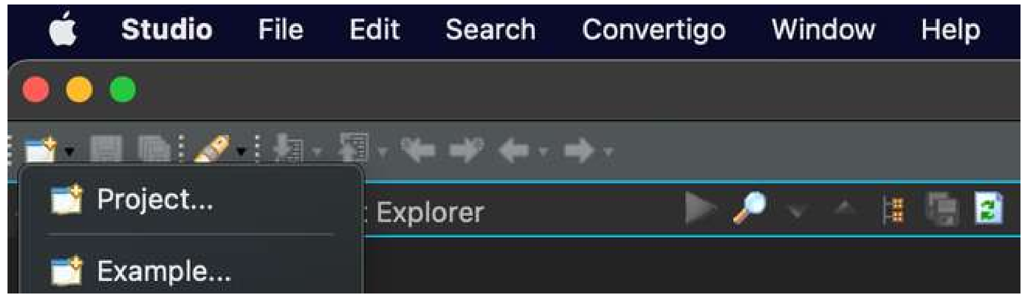
Click on the arrow on the right of this icon.





A sub menu appears.

Click on **Project** in the menu.

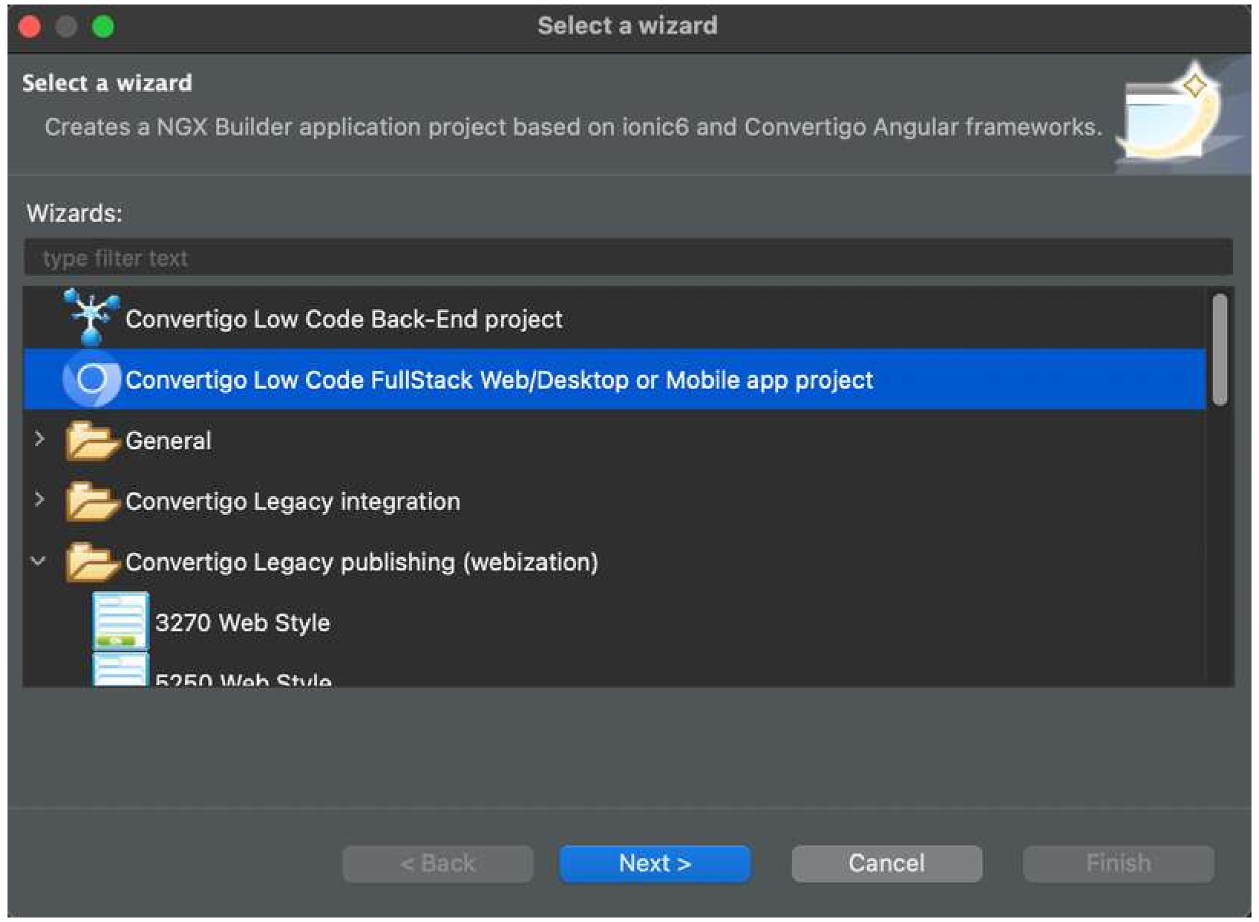


2.5 Create a project

Both options open the **Select a wizard window.**

Select **Convertigo Low Code FullStack Web/Desktop or Mobile app project**

and click on **Next>**.

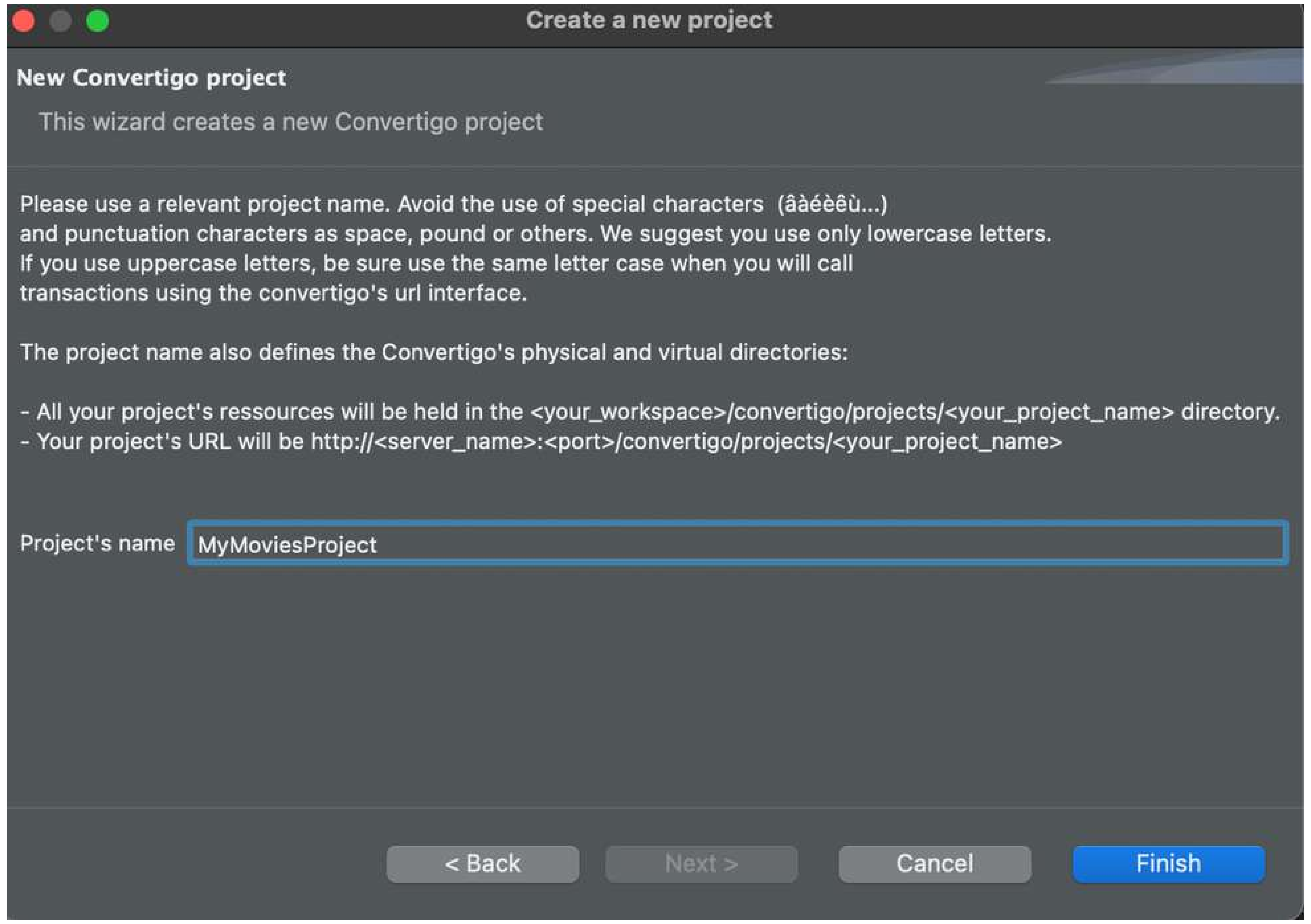

Select a wizard
Wizards:
type filter text
Convertigo Low Code Back-End project
Convertigo Low Code FullStack Web/Desktop or Mobile app project
General
Convertigo Legacy integration
Convertigo Legacy publishing (webization)
3270 Web Style
5250 Web Style
< Back
Next >
Cancel
Finish

Select a wizard

Creates a NGX Builder application project based on ionic6 and Convertigo Angular frameworks.

Convertigo Low Code FullStack Web/Desktop or Mobile app project

Enter a project name, then click on **Finish**.



New Convertigo project

This wizard creates a new Convertigo project

Please use a relevant project name. Avoid the use of special characters (âàéèêù...)
and punctuation characters as space, pound or others. We suggest you use only lowercase letters.
If you use uppercase letters, be sure use the same letter case when you will call
transactions using the convertigo's url interface.

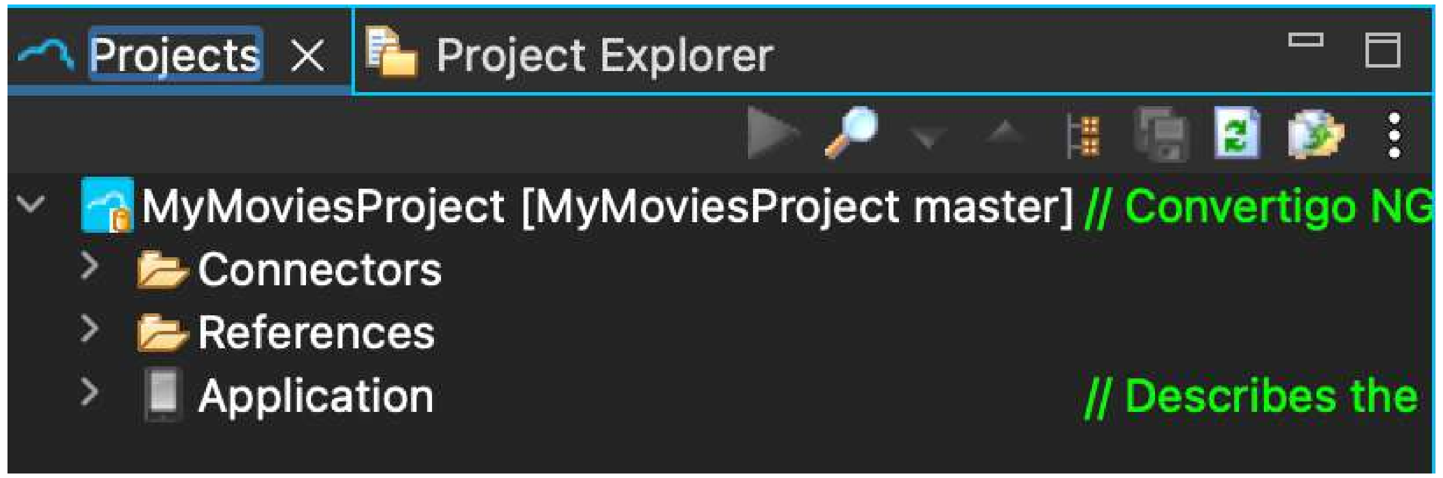
The project name also defines the Convertigo's physical and virtual directories:

- All your project's ressources will be held in the <your_workspace>/convertigo/projects/<your_project_name> directory.
- Your project's URL will be http://<server_name>:<port>/convertigo/projects/<your_project_name>

Project's name MyMoviesProject

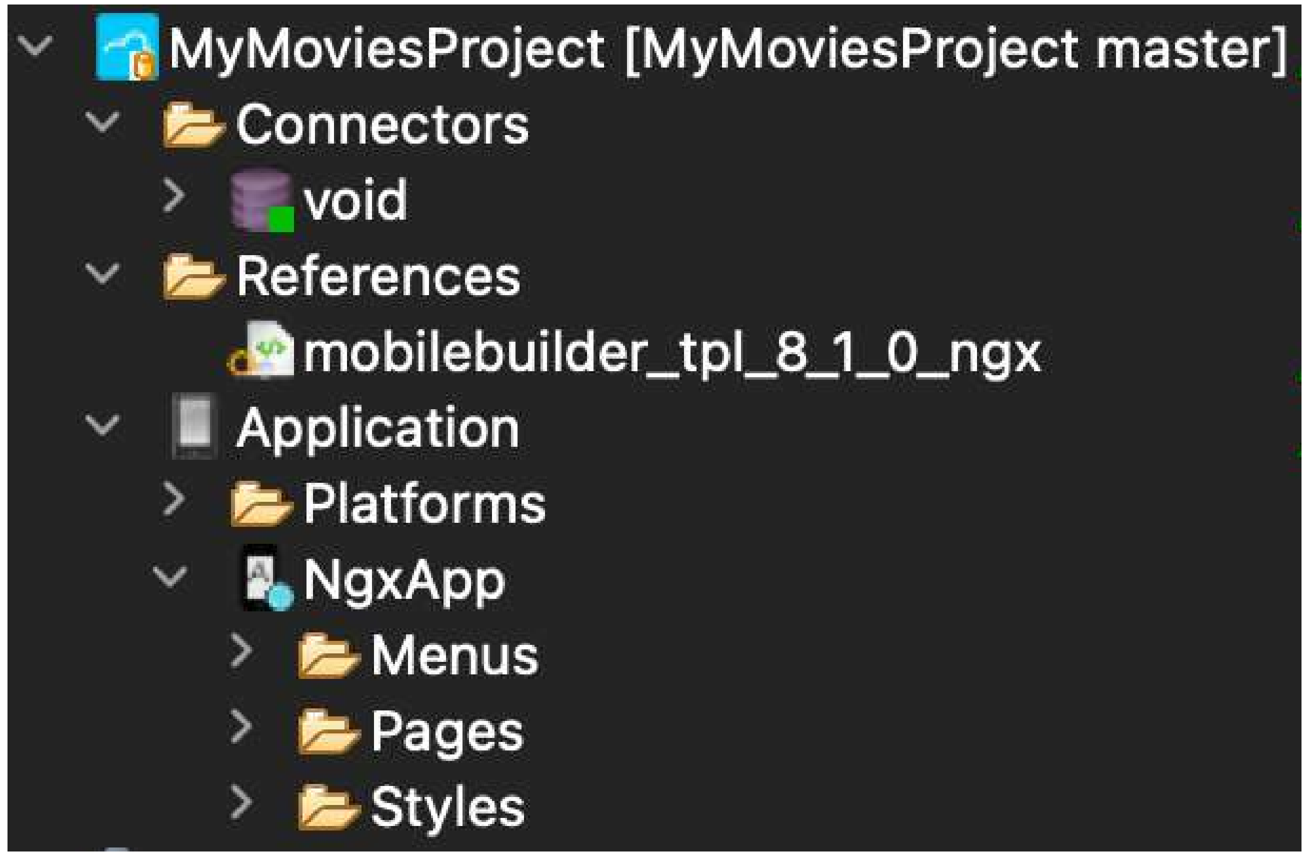
2.5 Create a project

As seen before, the project is created and appears in the **Project view**.



```
MyMoviesProject [MyMoviesProject master] // Convertigo NG
  Connectors
  References
  Application                             // Describes the
```

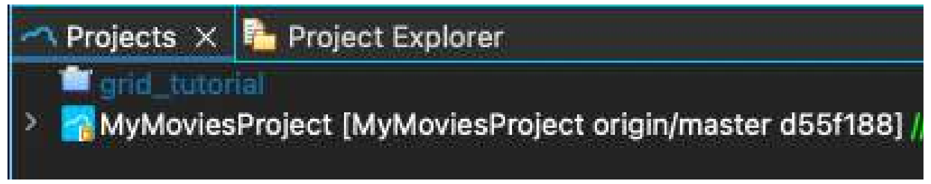
When created, a project has always the same structure.

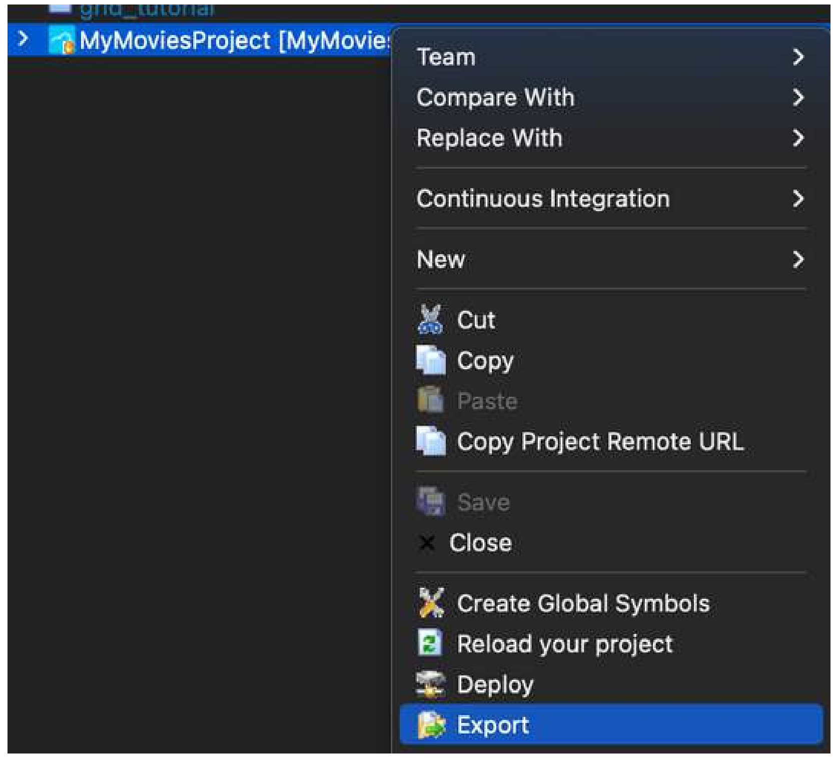


```
MyMoviesProject [MyMoviesProject master]
  Connectors
    void
  References
    mobilebuilder_tpl_8_1_0_ngx
  Application
    Platforms
    NgxApp
      Menus
      Pages
      Styles
```

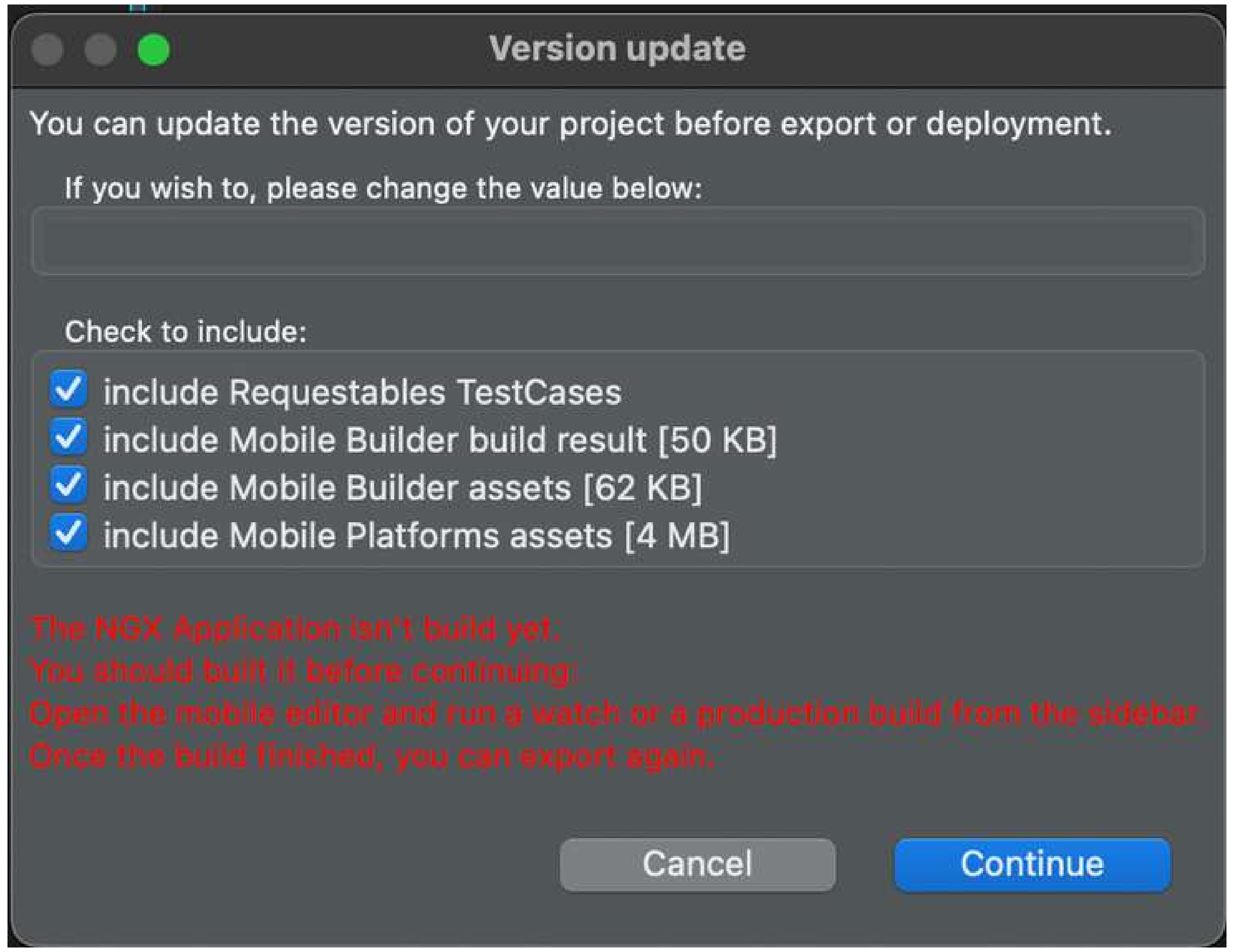
2.6 Export a project

Let's say you want to export a project:
right-click on the **name of your project** in the **Projects view**,
then click on **Export**.





The **Version update window** appears.



You can update the version of your project before export or deployment.

If you wish to, please change the value below:

Check to include:

- ☑ include Requestables TestCases
- ☑ include Mobile Builder build result [50 KB]
- ☑ include Mobile Builder assets [62 KB]
- ☑ include Mobile Platforms assets [4 MB]

The NGX Application isn't build yet.
You should built it before continuing.
Open the mobile editor and run a watch or a production build from the sidebar.
Once the build finished, you can export again.

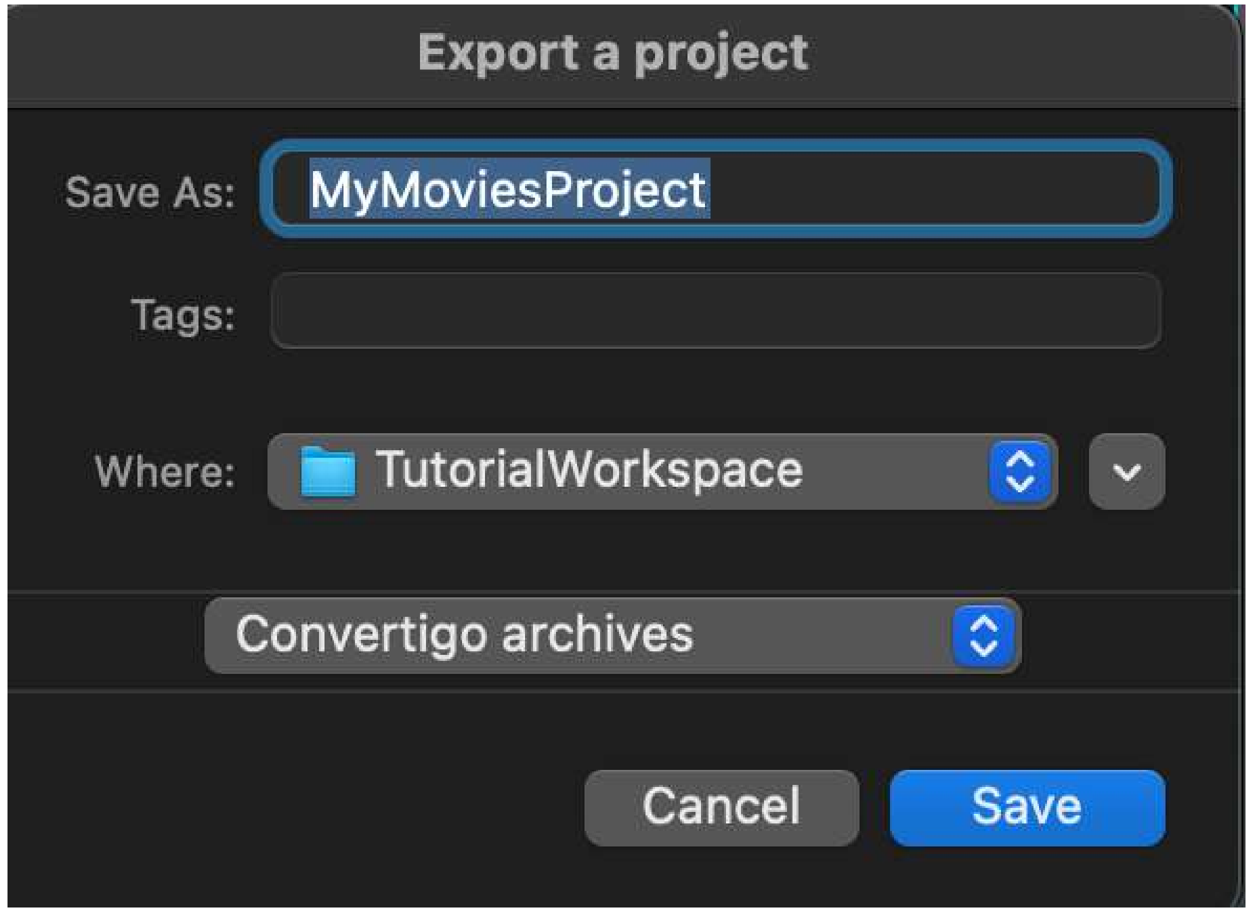
A message in red indicates that you need to run **a watch or a production build** from the **mobile editor.**

Building the project
is necessary **only for the frontend**.

For now, we are **working on the backend**,
so we can ignore this message,
and click on **Continue**.

2.6 Export a project

In the **Export a project window**,
you can change the name of the project,
and select where it is saved.



Export a project

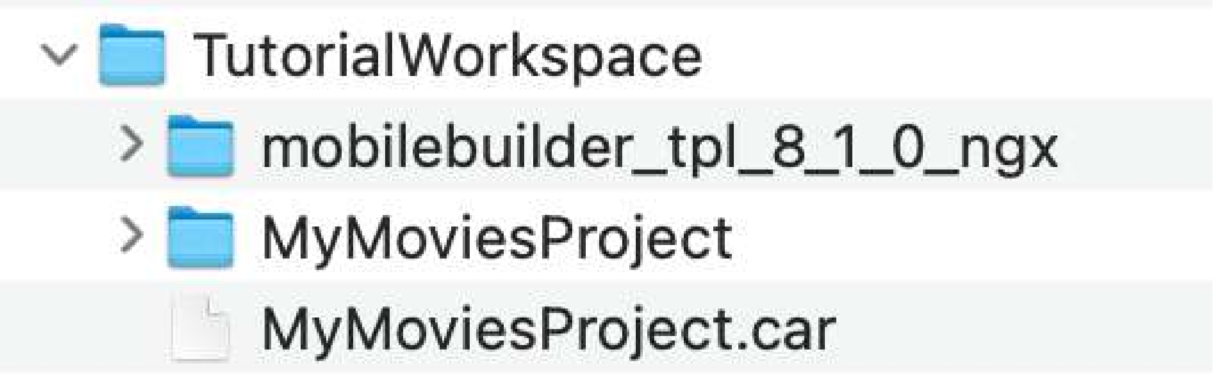
Save As: MyMoviesProject

Tags:

Where: TutorialWorkspace

Convertigo archives

In the folder where it was saved,
the project appears as a **.car file**.

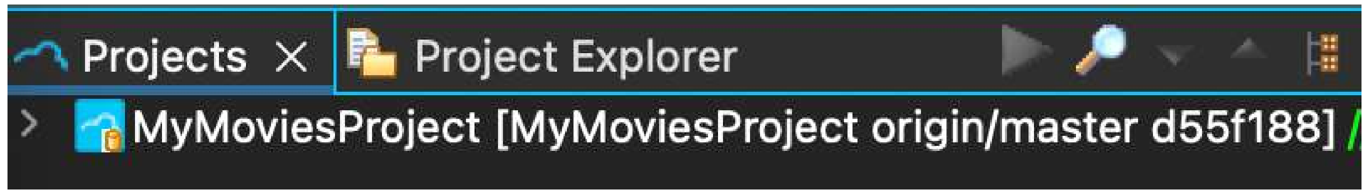


TutorialWorkspace
mobilebuilder_tpl_8_1_0_ngx
MyMoviesProject
MyMoviesProject.car

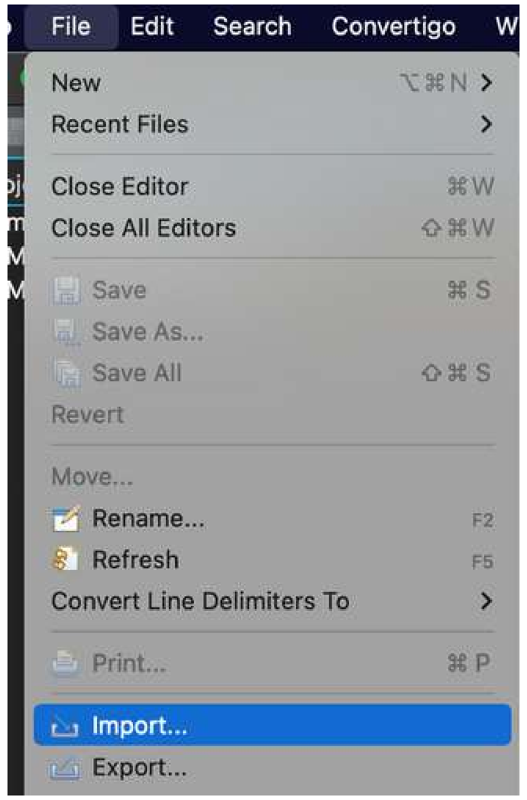
MyMoviesProject.car

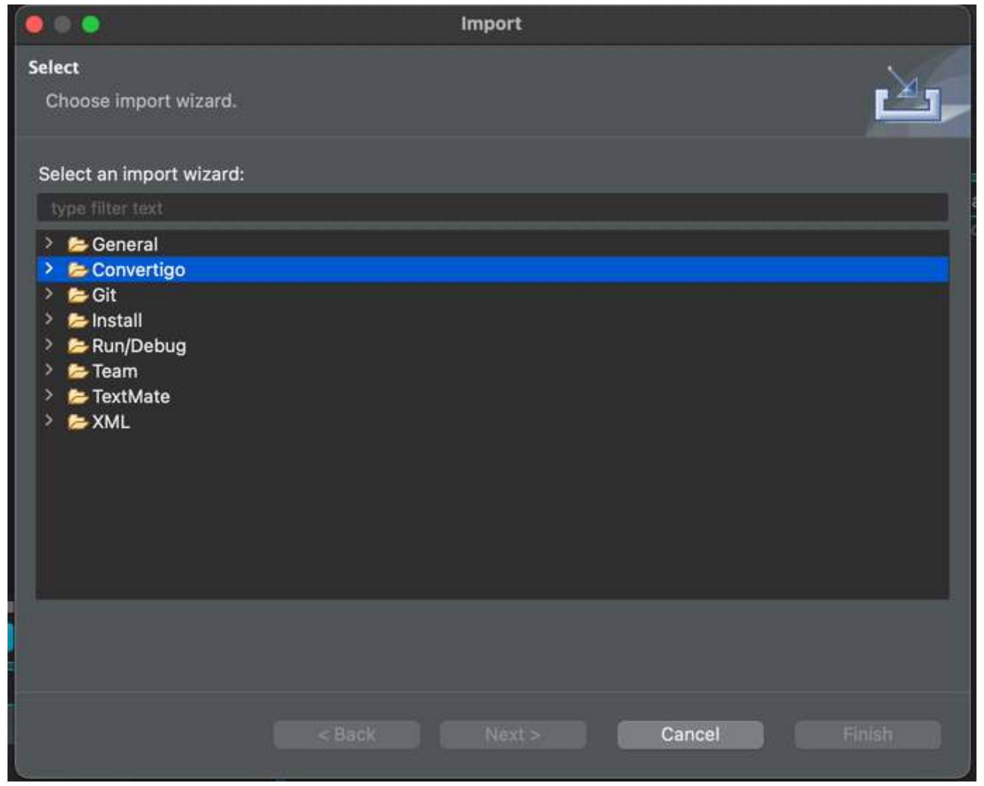
2.7 Import a project

Let’s say you want to import a project from a .car file in your workspace



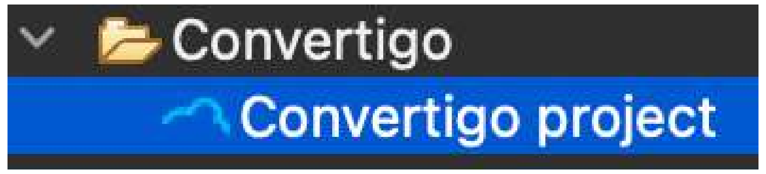
Click on **File**, then **Import** and the **Import windows appears**.

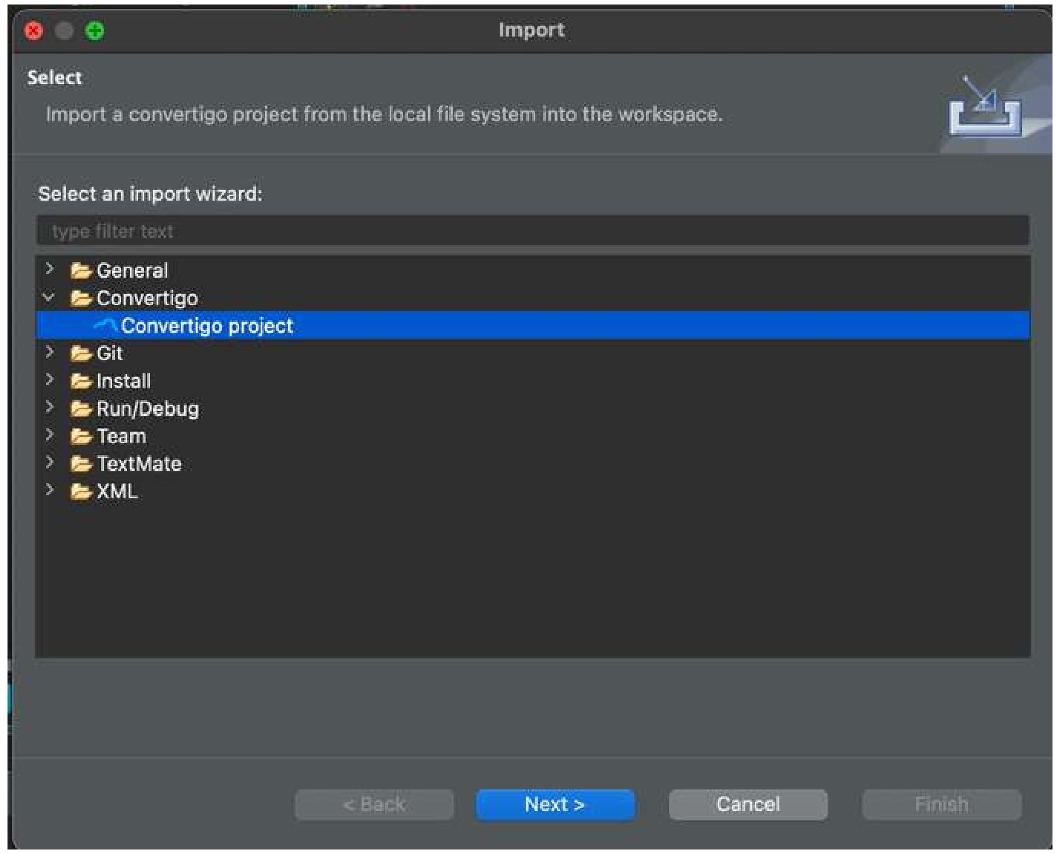




Choose import wizard.

In the **Import windows**, click on **Convertigo**, select **Convertigo project**, then click on **Next>**.

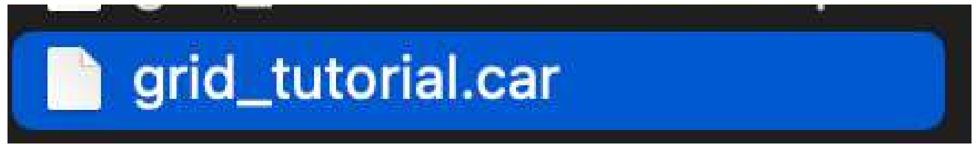




Import a convertigo project from the local file system into the workspace.

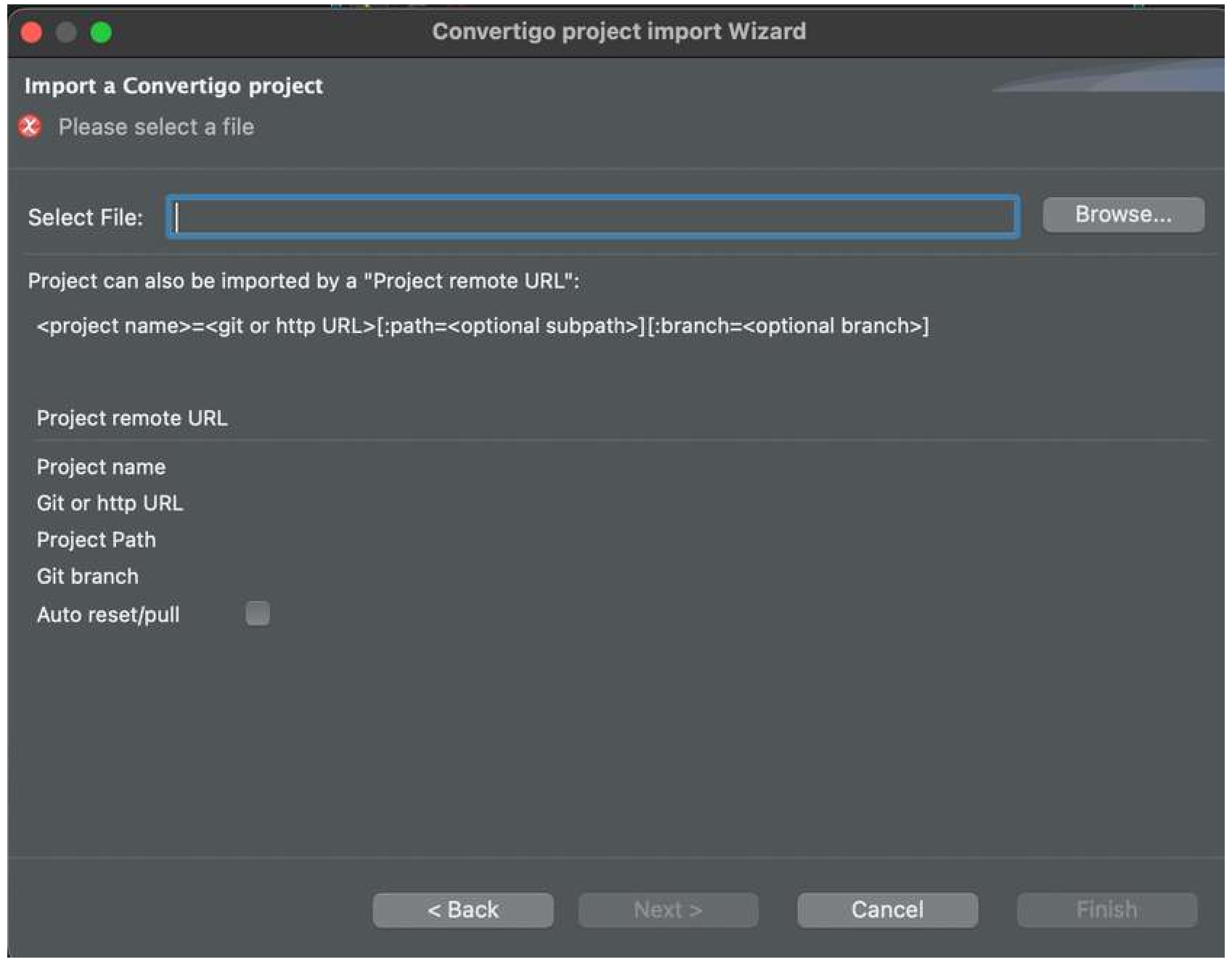
2.7 Import a project

In the **Convertigo Project Import window**, click on **Browse** to select a file (here grid_tutorial.car) anywhere in your computer.



grid_tutorial.car

Then click on **Next>**.



Convertigo project import Wizard

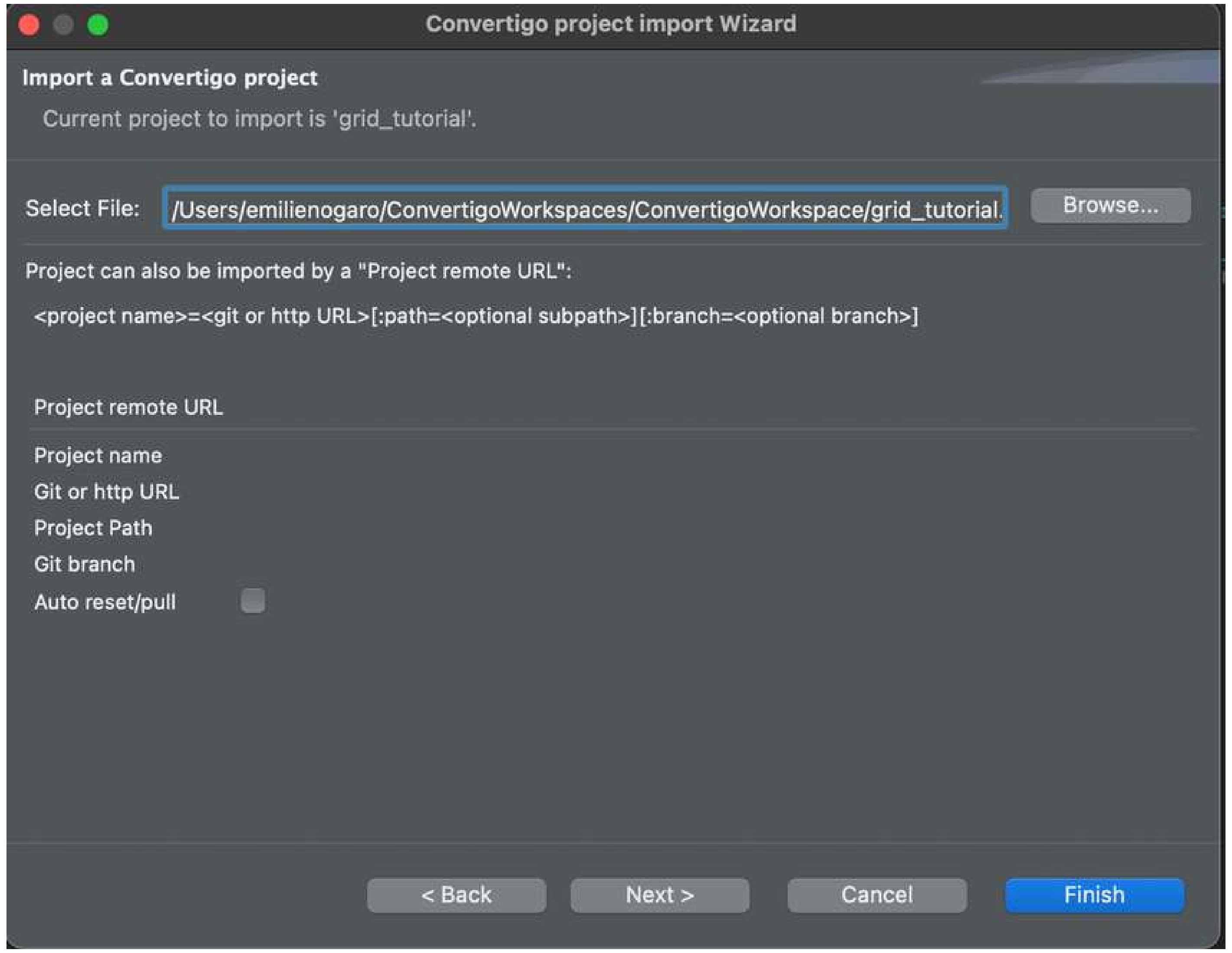
Import a Convertigo project

Please select a file

Select File:

Project can also be imported by a "Project remote URL":

<project name>=<git or http URL>[:path=<optional subpath>][:branch=<optional branch>]



Convertigo project import Wizard

Import a Convertigo project

Current project to import is 'grid_tutorial'.

Select File: /Users/emilienogaro/ConvertigoWorkspaces/ConvertigoWorkspace/grid_tutorial.

Project can also be imported by a "Project remote URL":

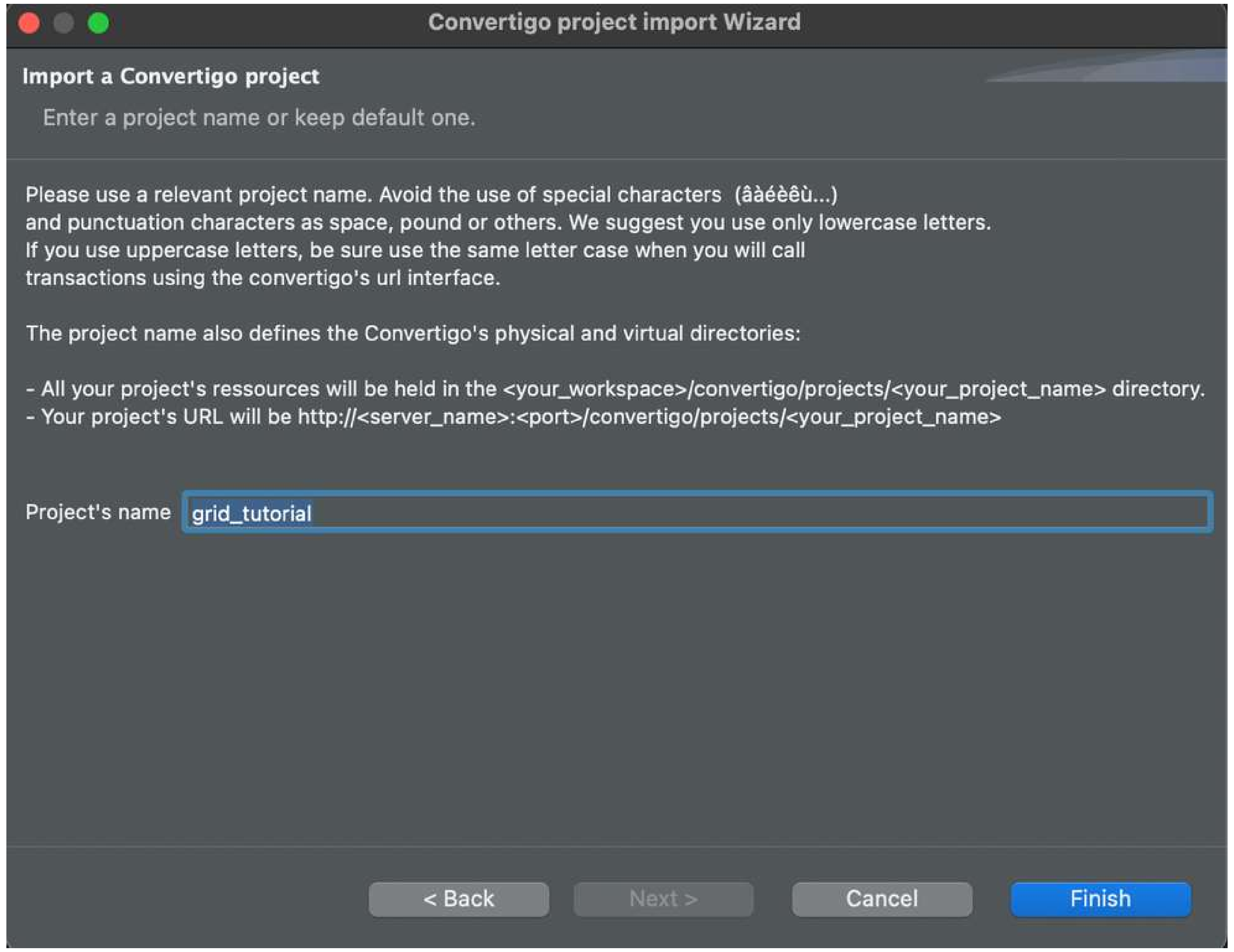
<project name>=<git or http URL>[:path=<optional subpath>][:branch=<optional branch>]

2.7 Import a project

You can rename the project

or keep the .default file name (.car file name).

Then click on **Finish**.



Convertigo project import Wizard

Import a Convertigo project

Enter a project name or keep default one.

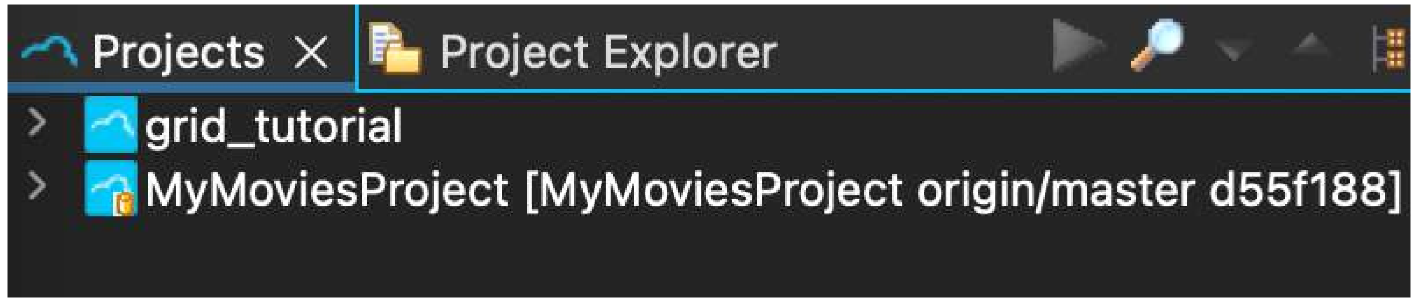
Please use a relevant project name. Avoid the use of special characters (âàéèêù...)
and punctuation characters as space, pound or others. We suggest you use only lowercase letters.
If you use uppercase letters, be sure use the same letter case when you will call
transactions using the convertigo's url interface.

The project name also defines the Convertigo's physical and virtual directories:

- All your project's ressources will be held in the <your_workspace>/convertigo/projects/<your_project_name> directory.
- Your project's URL will be http://<server_name>:<port>/convertigo/projects/<your_project_name>

Project's name: grid_tutorial

In the **Projects view**, the project appears.



- grid_tutorial
- MyMoviesProject [MyMoviesProject origin/master d55f188]

convertigo

3 – Web services Connectors & Transactions

How to consume a rest API.

3.1 Presentation of the API TMDB

3.2 HTTP connectors & JSON HTTP transactions

3.3 Create an HTTP connector

3.4 Configure the HTTP connector

3.5 Create a transaction

3.6 Add a token

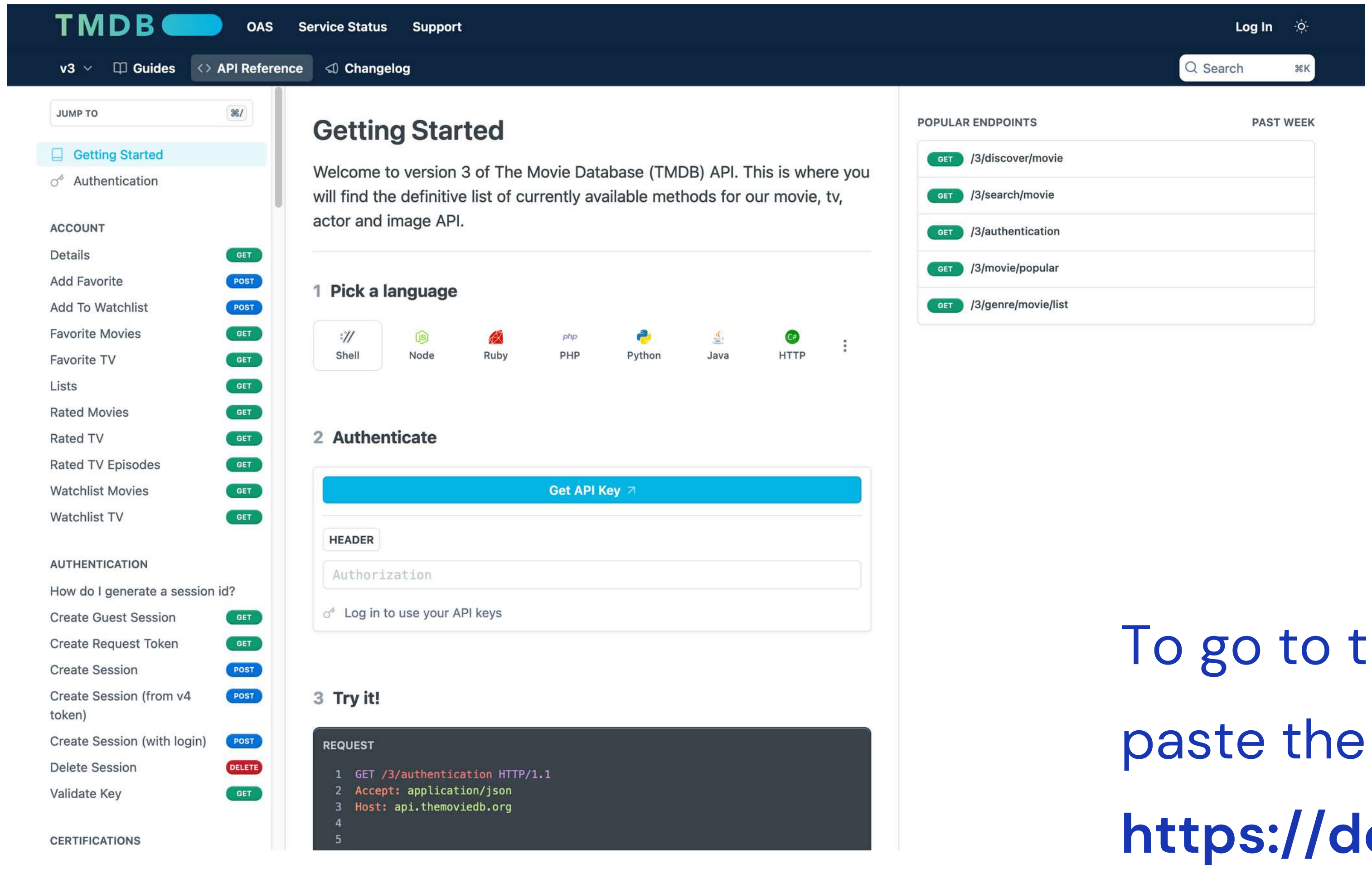
3.7 Edit the request path

3.8 Test the request

3.1 Presentation of the API TMDB

The Movie Database (TMDb) API provides access to a vast database of information related to movies and television shows.

It is commonly used by developers to integrate movie-related data into their applications, websites, and services



Getting Started

Welcome to version 3 of The Movie Database (TMDB) API. This is where you will find the definitive list of currently available methods for our movie, tv, actor and image API.

1 Pick a language

2 Authenticate

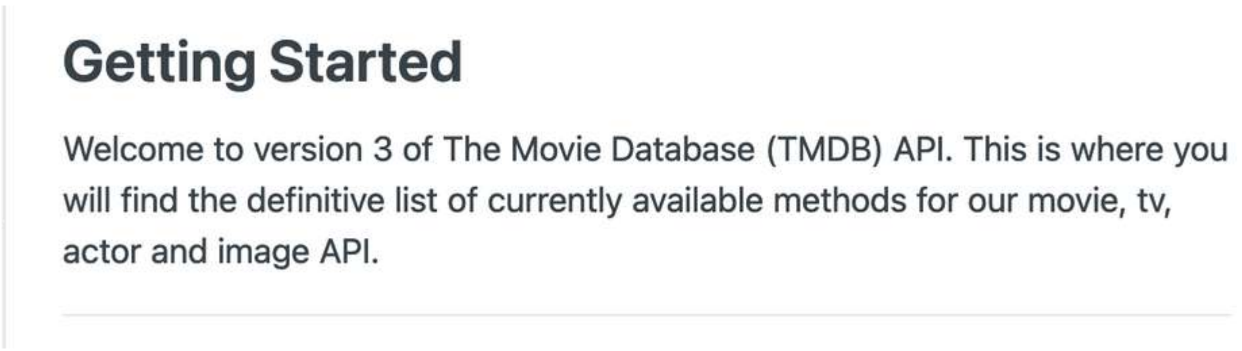
Log in to use your API keys

3 Try it!

REQUEST

```
GET /3/authentication HTTP/1.1
Accept: application/json
Host: api.themoviedb.org


```

POPULAR ENDPOINTS	PAST WEEK
GET	/3/discover/movie
GET	/3/search/movie
GET	/3/authentication
GET	/3/movie/popular
GET	/3/genre/movie/list



Getting Started

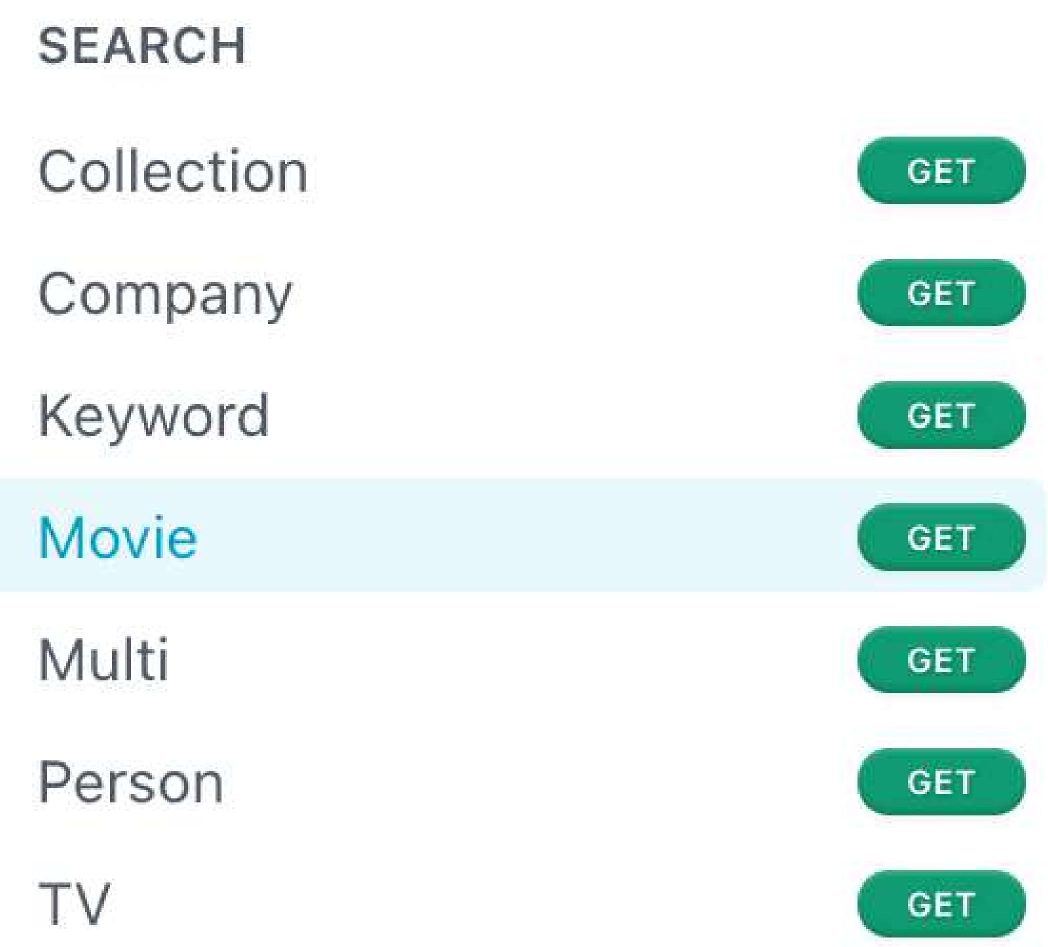
Welcome to version 3 of The Movie Database (TMDB) API. This is where you will find the definitive list of currently available methods for our movie, tv, actor and image API.

To go to the **Getting started page** of the API, paste the following link in your browser:
https://developer.themoviedb.org/reference/intro/getting-started

3.1 Presentation of the API TMDB

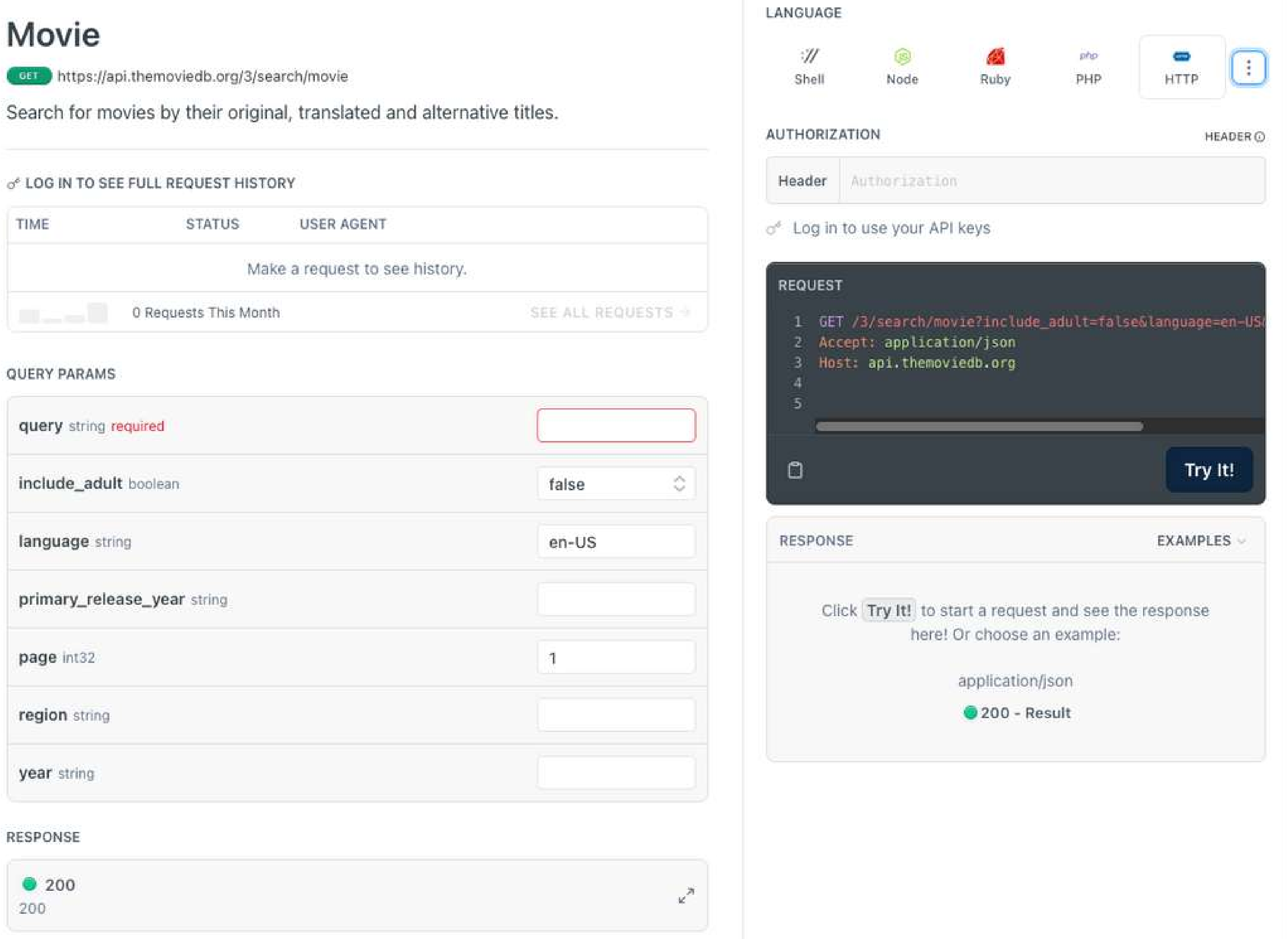
In the API TMDB documentation, a lot of different requests are available.

Let’s go to the **Search Movie page** (https://developer.themoviedb.org/reference/search-movie).



SEARCH

Collection	GET
Company	GET
Keyword	GET
Movie	GET
Multi	GET
Person	GET
TV	GET



Movie

GET https://api.themoviedb.org/3/search/movie

Search for movies by their original, translated and alternative titles.

LOG IN TO SEE FULL REQUEST HISTORY

TIME	STATUS	USER AGENT
Make a request to see history.		

0 Requests This Month SEE ALL REQUESTS →

QUERY PARAMS

query string required	
include_adult boolean	false
language string	en-US
primary_release_year string	
page int32	1
region string	
year string	

RESPONSE

● 200
200

AUTHORIZATION HEADER

Header Authorization

Log in to use your API keys

REQUEST

```
GET /3/search/movie?include_adult=false&language=en-US
Accept: application/json
Host: api.themoviedb.org


```

RESPONSE EXAMPLES

Click Try It! to start a request and see the response here! Or choose an example:

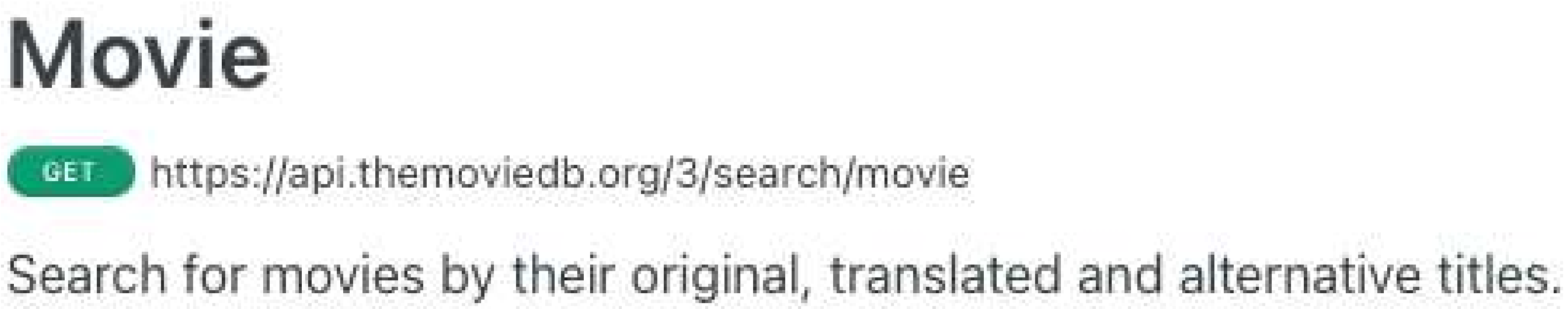
application/json

● 200 - Result

3.1 Presentation of the API TMDB

All the informations you need to write a **Search Movie HTTP REQUEST** are present on the **Search Movie page**.

GET HTTP Request url to search a movie

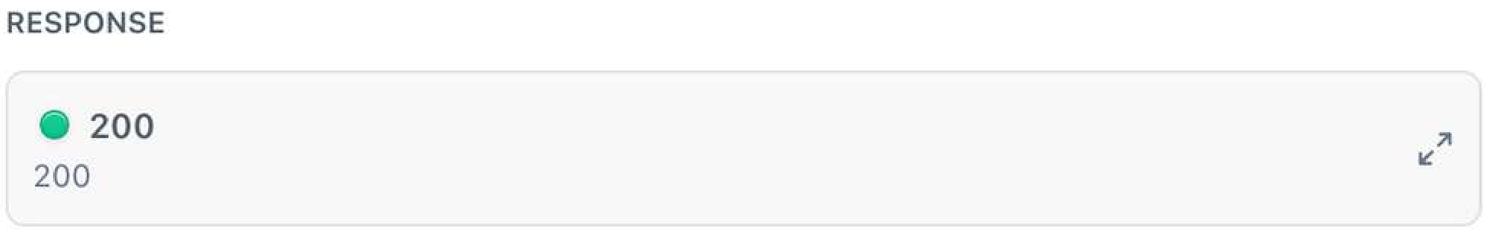


Movie

GET https://api.themoviedb.org/3/search/movie

Search for movies by their original, translated and alternative titles.

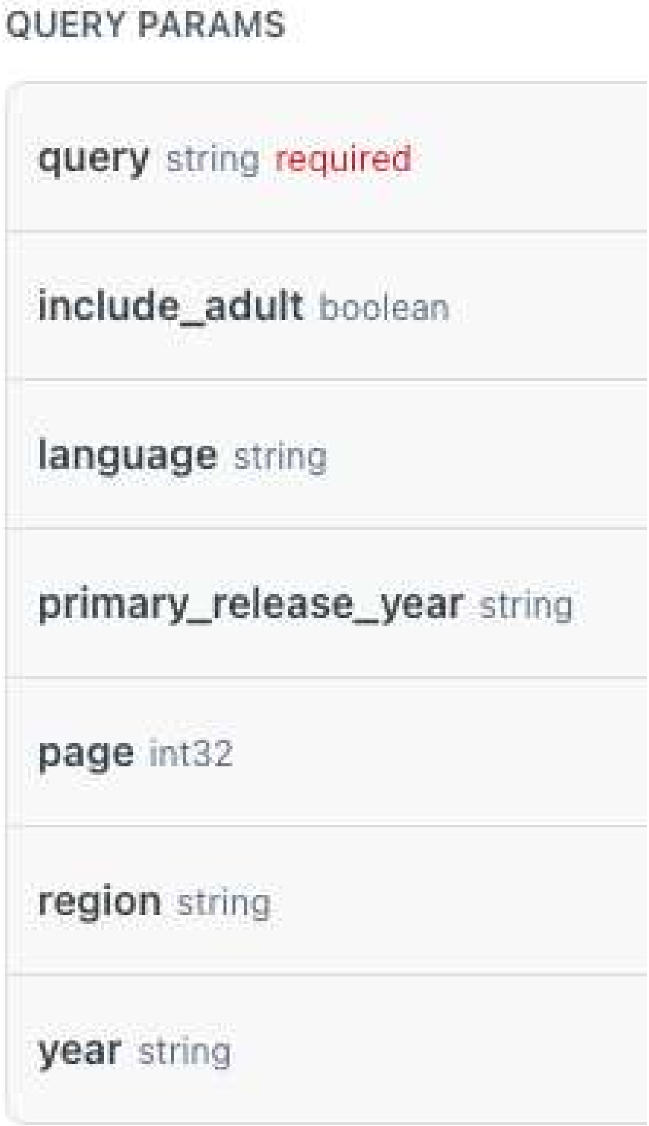
Expected Response code



RESPONSE

● 200
200

Required and Optional Query params.



QUERY PARAMS

query string required

include_adult boolean

language string

primary_release_year string

page int32

region string

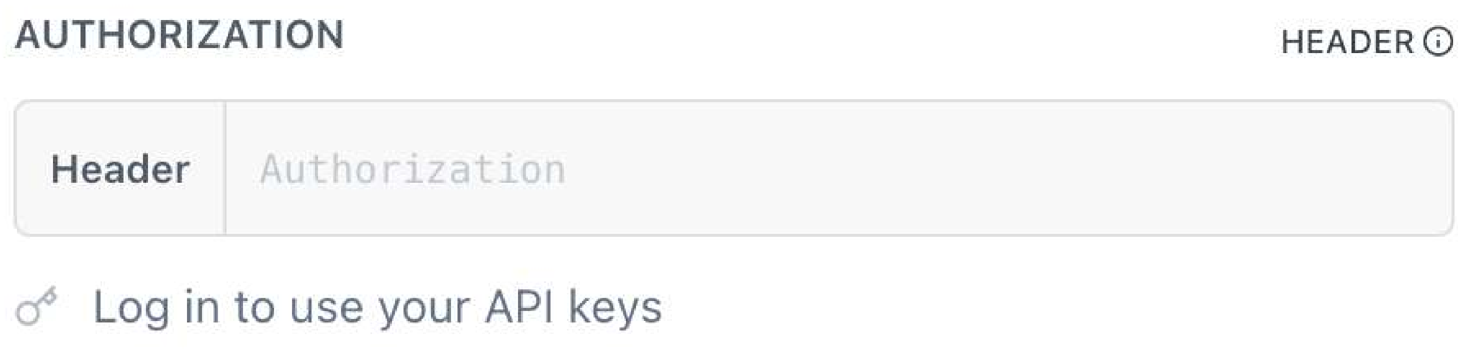
year string

3.1 Presentation of the API TMDB

To use the API TMDB, it is **necessary to create an account**.

Once registered, you will have an **API Key** or **personnal Access Token**.

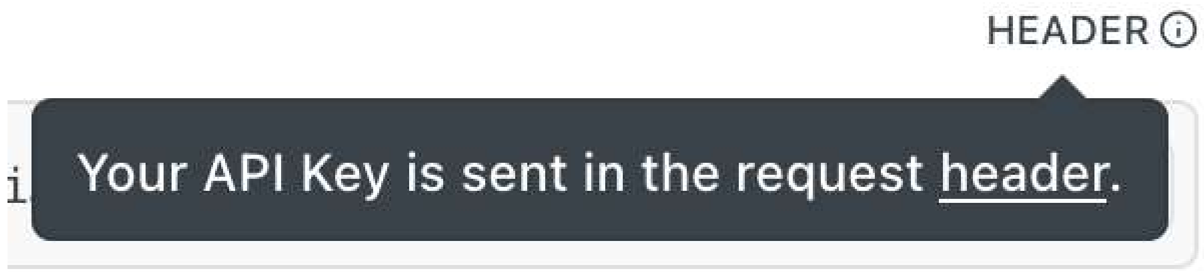
It will be used in the **request Header** as **Authorization param**.



AUTHORIZATION HEADER ⓘ

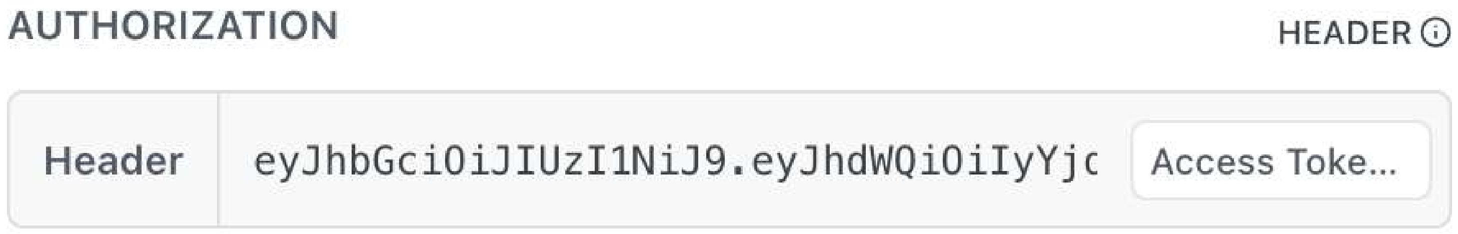
Header Authorization

Log in to use your API keys



HEADER ⓘ

Your API Key is sent in the request header.



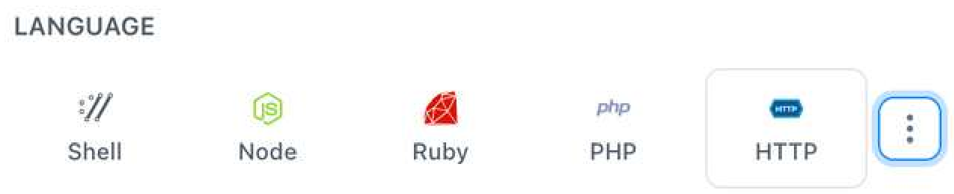
AUTHORIZATION HEADER ⓘ

Header eyJhbGciOiJIUzI1NiJ9.eyJhdWQiOiIyYjc

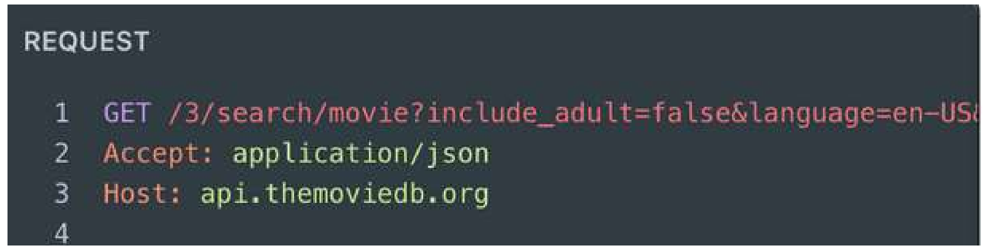
Access Toke...

The personnal access token appears automatically when you are logged in.

3.1 Presentation of the API TMDB



Change the Language to HTTP to see the request as HTTP

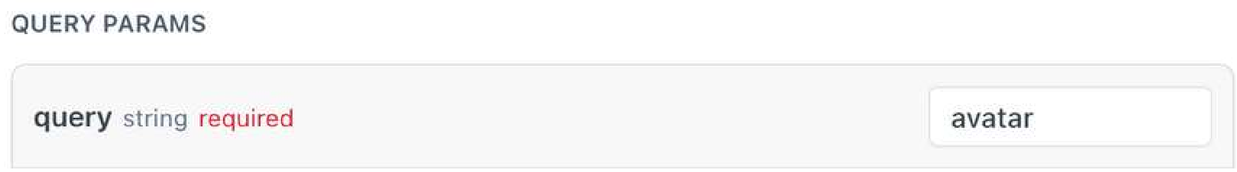


REQUEST

```
GET /3/search/movie?include_adult=false&language=en-US&
Accept: application/json
Host: api.themoviedb.org
```

Request url by default

GET /3/search/movie?include_adult=false&language=en-US&page=1

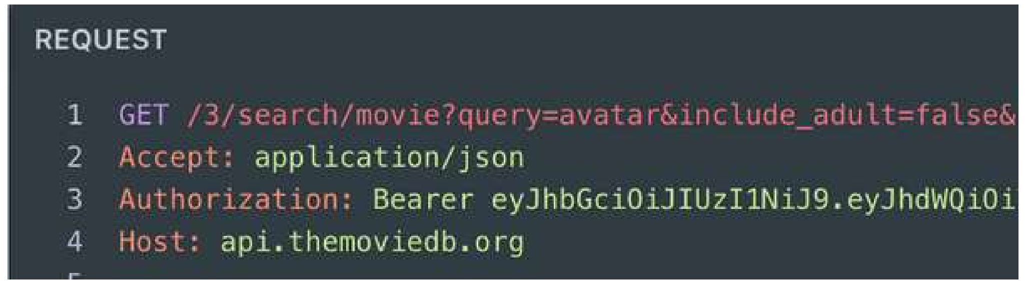


QUERY PARAMS

query string required: avatar

When you add a query param,

the request url changes to include it.



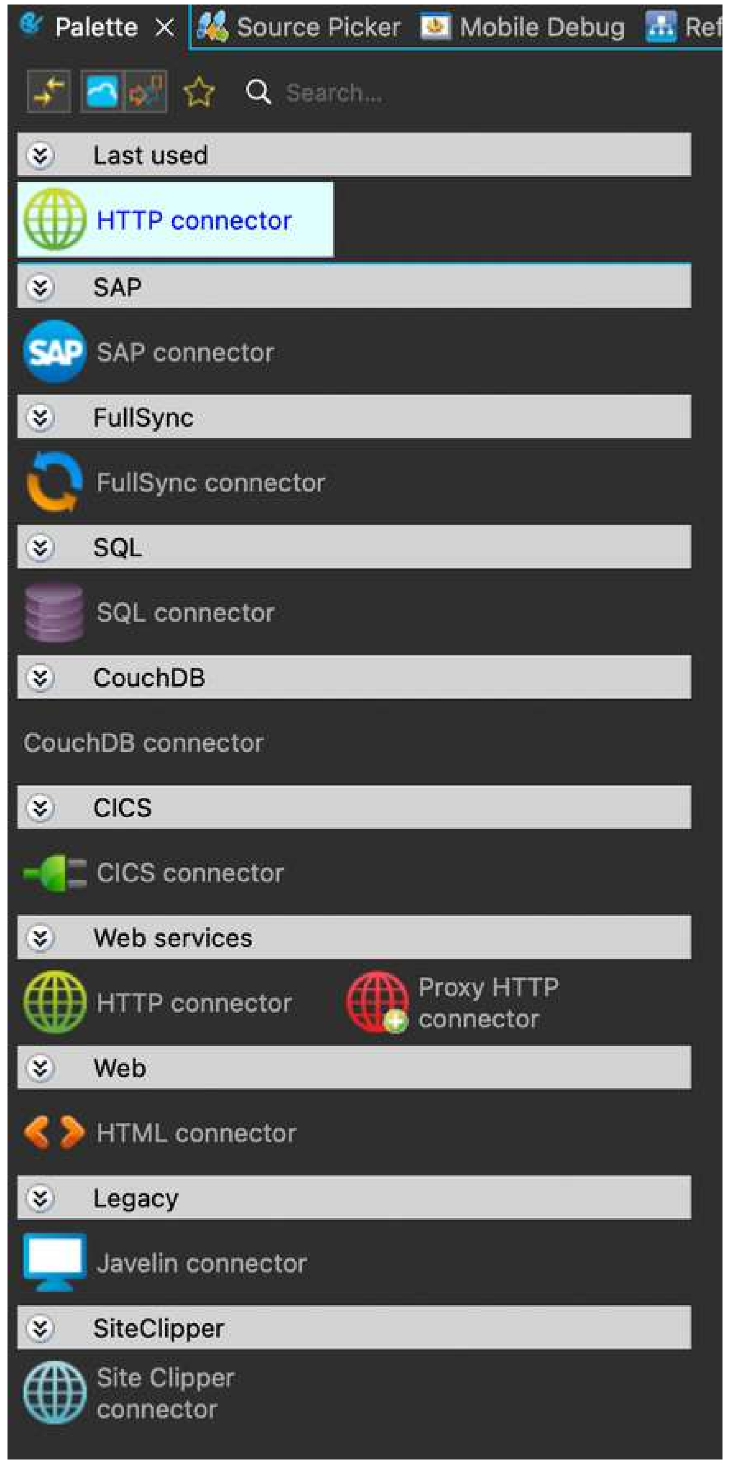
REQUEST

```
GET /3/search/movie?query=avatar&include_adult=false&
Accept: application/json
Authorization: Bearer eyJhbGciOiJIUzI1NiJ9.eyJhdWQiOi
Host: api.themoviedb.org
```

GET /3/search/movie?

query=avatar&include_adult=false&language=en-US&page=1

3.2 HTTP connectors & JSON HTTP transactions



Palette × Source Picker Mobile Debug Ref

Search...

Last used

HTTP connector

SAP

SAP connector

FullSync

FullSync connector

SQL

SQL connector

CouchDB

CouchDB connector

CICS

CICS connector

Web services

HTTP connector Proxy HTTP connector

Web

HTML connector

Legacy

Javelin connector

SiteClipper

Site Clipper connector

There are **different connectors and transactions** in Convertigo, used for **different data providers** (SQL, Web services, Legacy apps running on mainframes...).

HTTP connector

For a REST API, you use the **HTTP connector**. It allows Convertigo to connect and communicate with **HTTP servers**. It is used to **consume REST and SOAP web services**, and retrieve data using the **HTTP protocol**.

JSON HTTP transaction

To consume a **JSON web service**, you use a **JSON HTTP transaction**. It performs the conversion of JSON data from the web service into XML transaction output.

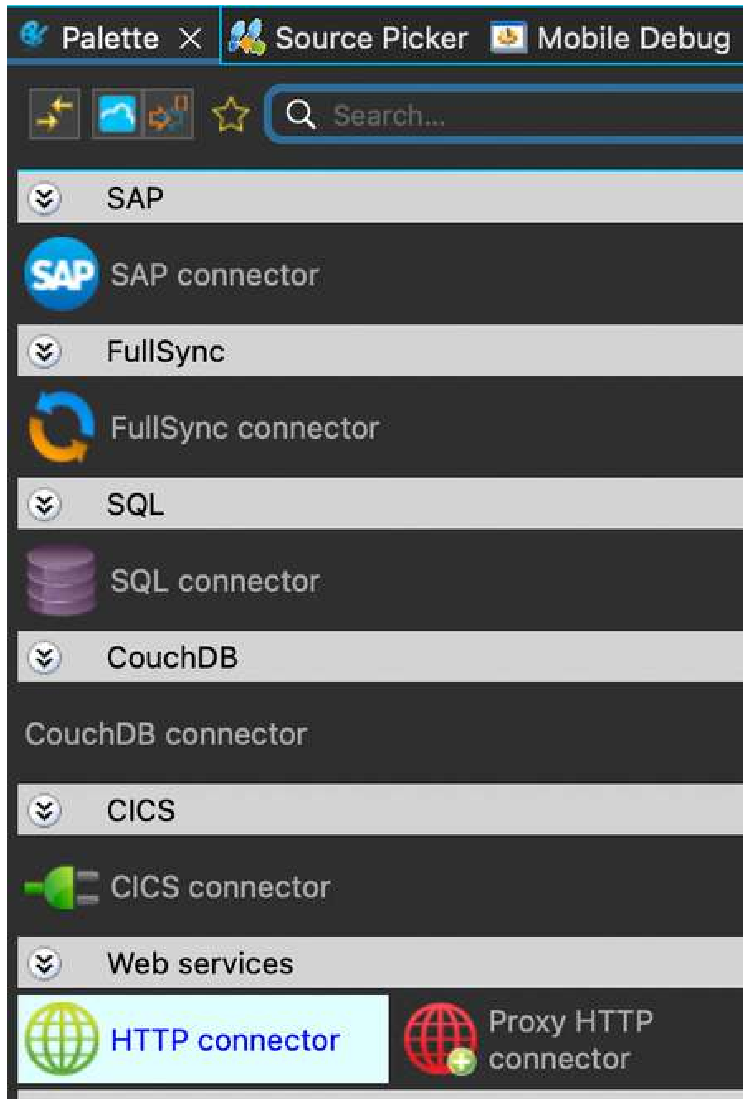
3.3 Create an HTTP connector

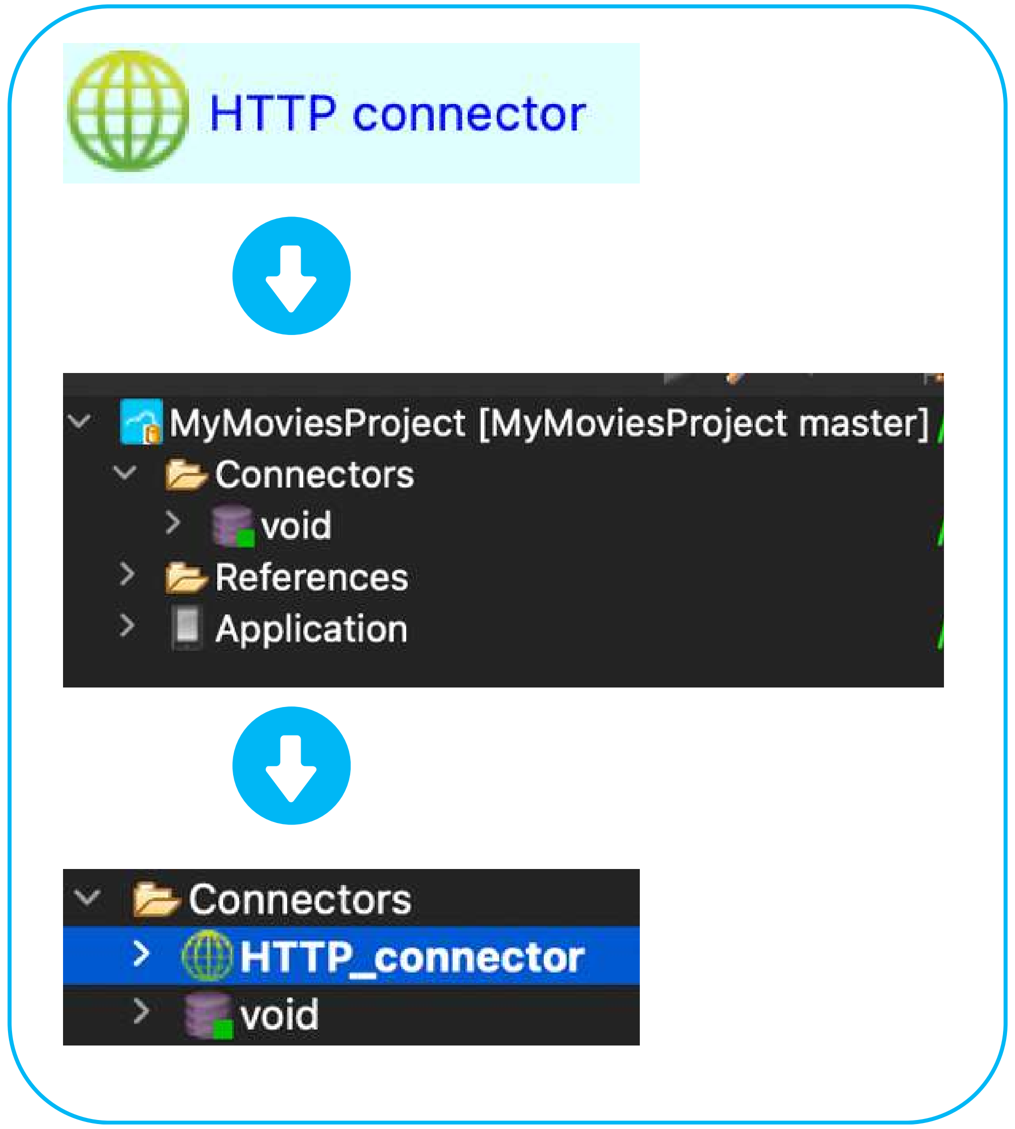
To connect to a **REST API**,

you need to **create an HTTP_connector** in the **Connectors folder**.

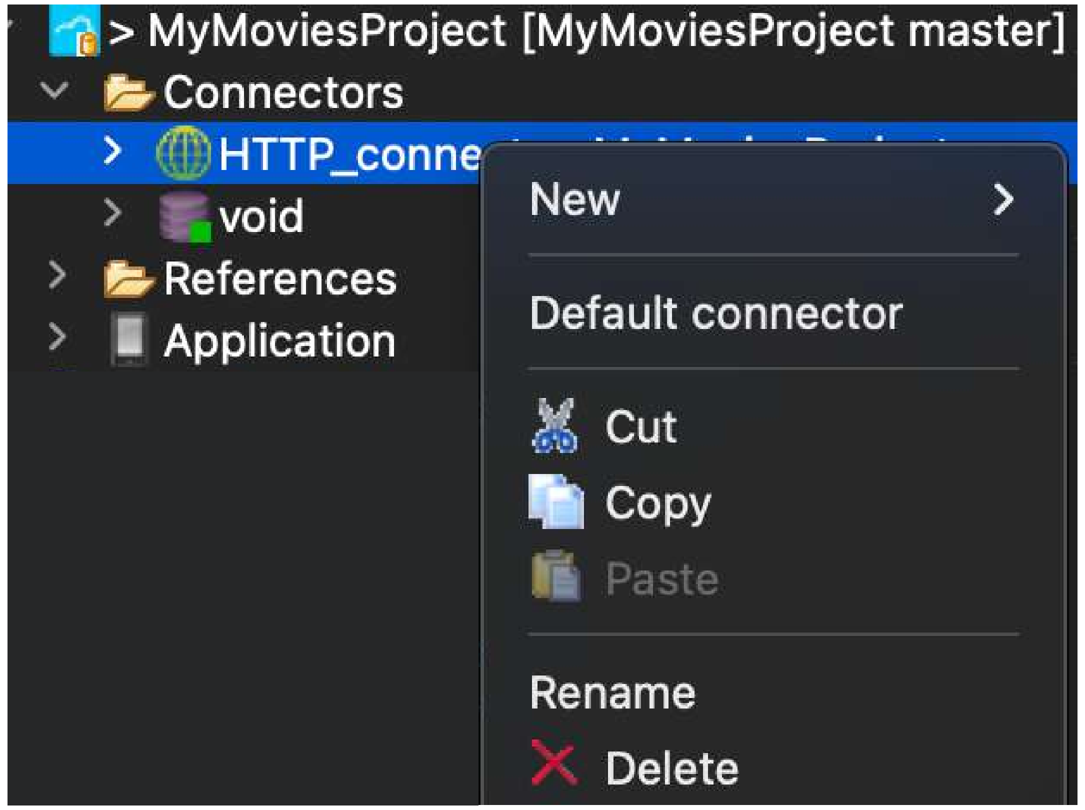
First option:

Drag and drop it from the palette into the folder.





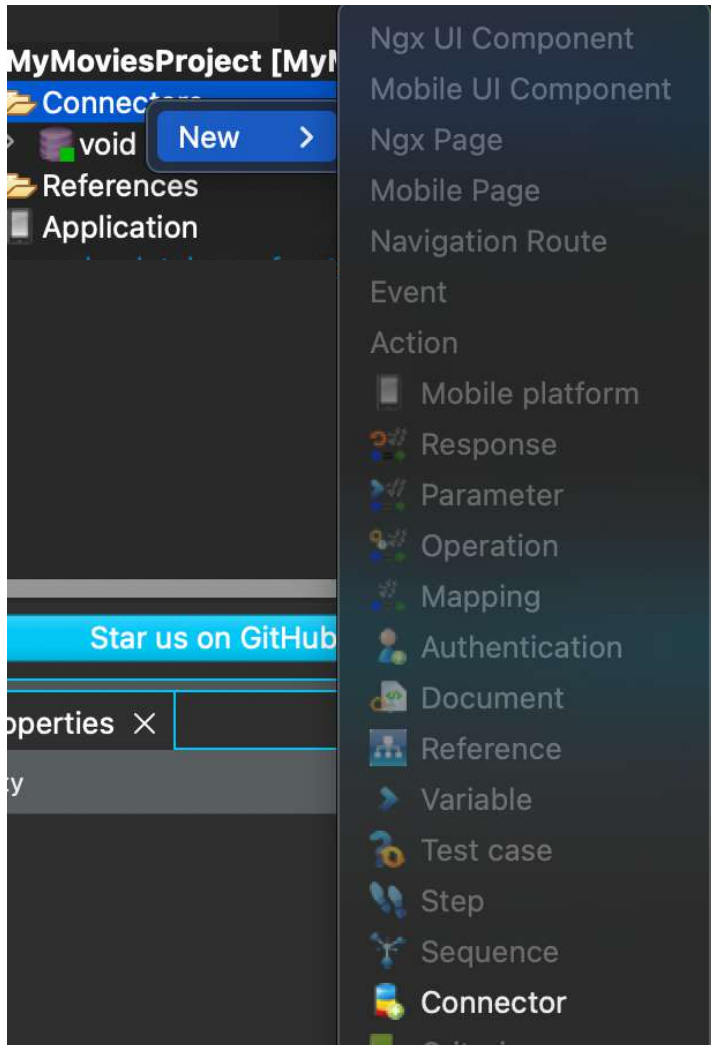
You can then **rename** the connector by right-clicking on it.

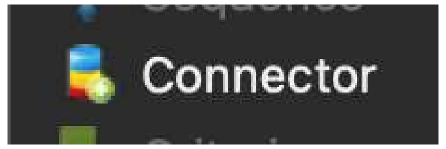


3.3 Create an HTTP connector

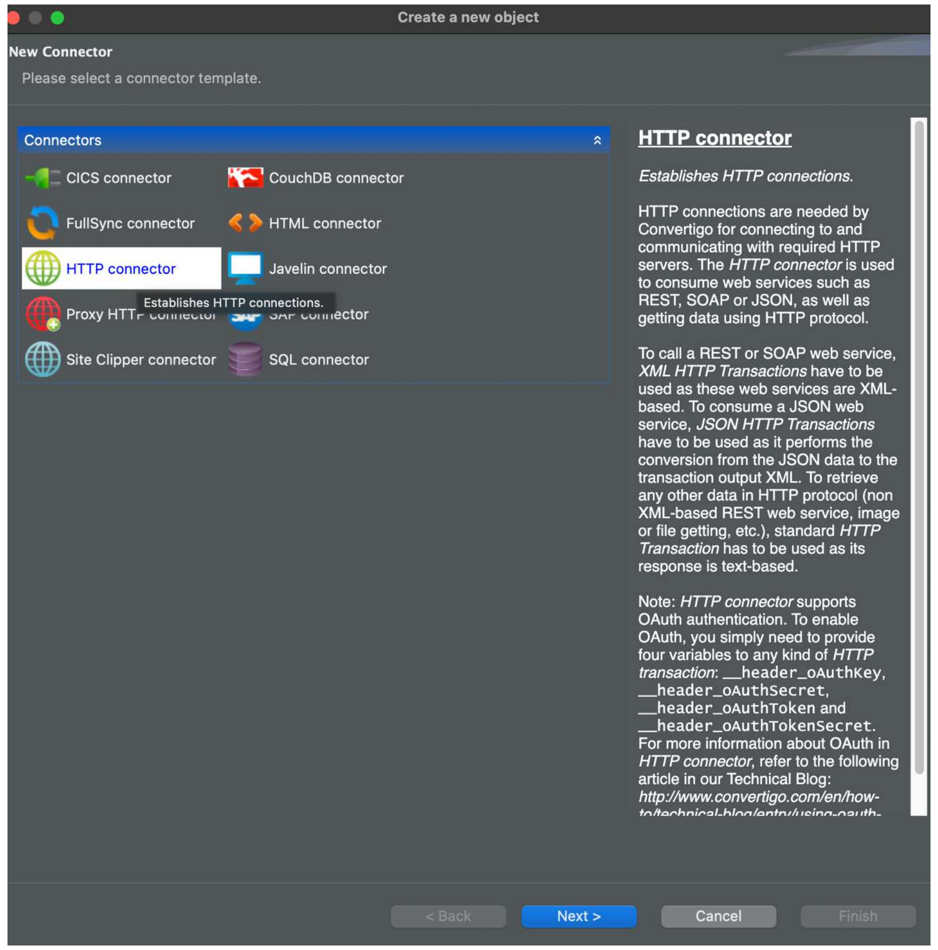
Second option:

Right-click on the **Connectors** folder, then select **New** and choose **Connector**.





In the **Create a new object** window, select **HTTP connector** and then click on **Next>**.



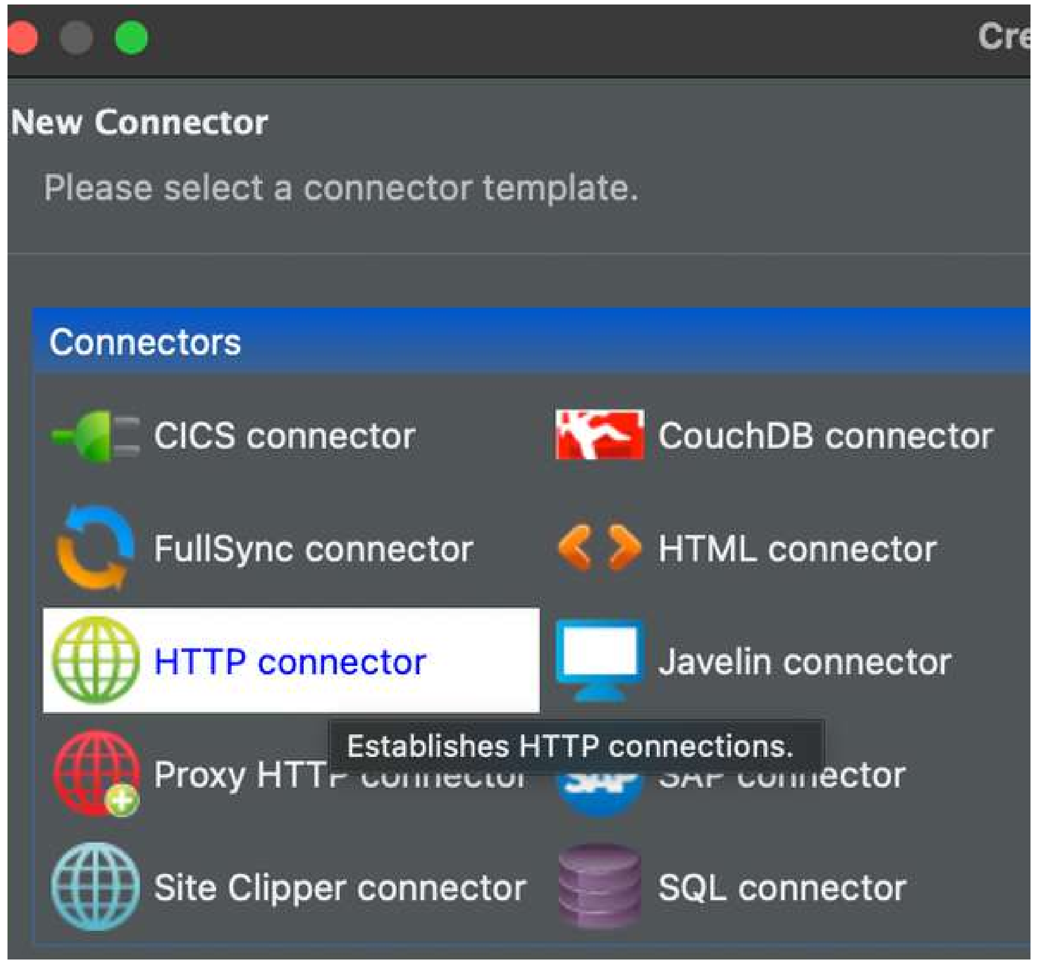
HTTP connector

Establishes HTTP connections.

HTTP connections are needed by Convertigo for connecting to and communicating with required HTTP servers. The *HTTP connector* is used to consume web services such as REST, SOAP or JSON, as well as getting data using HTTP protocol.

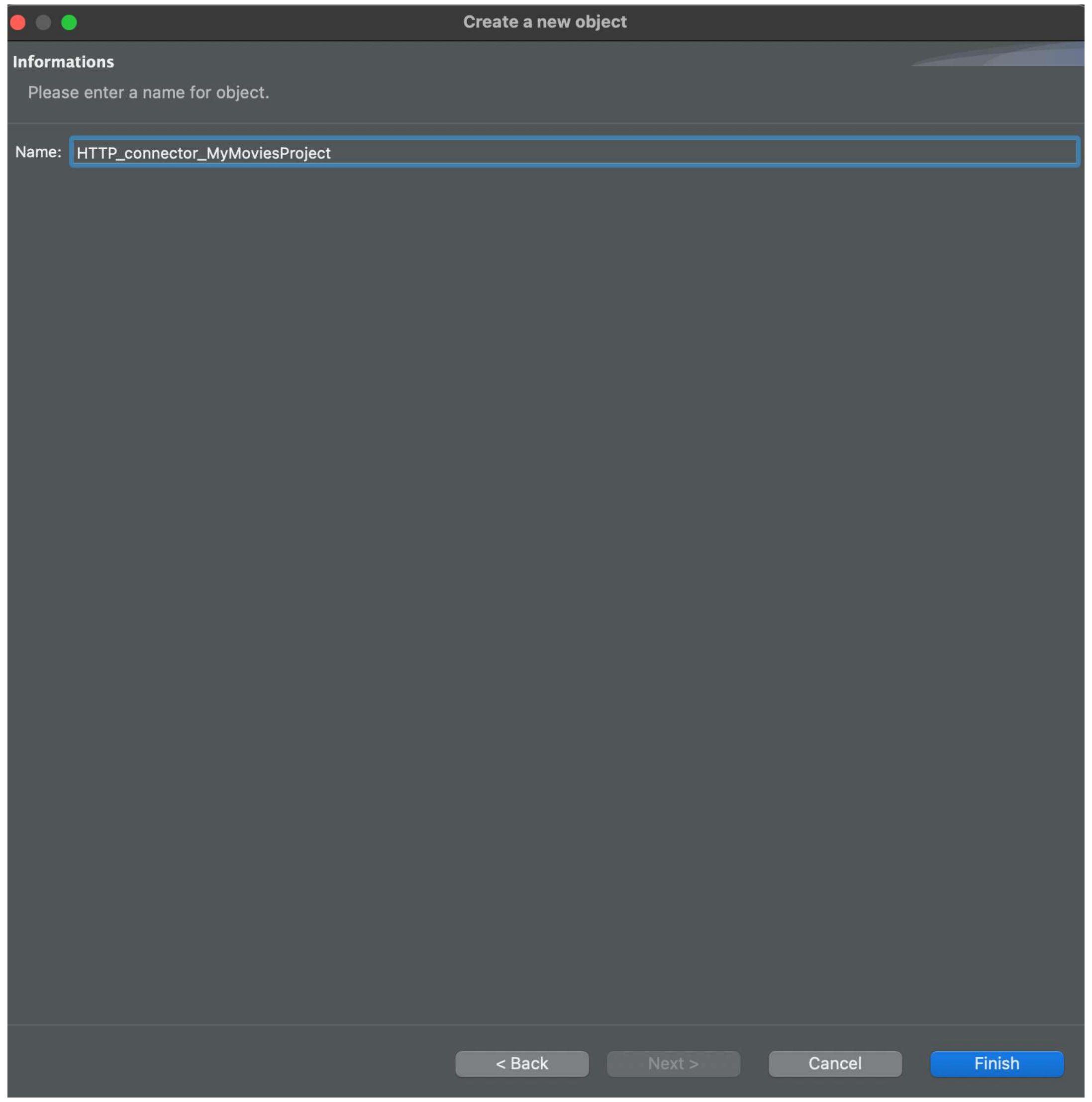
To call a REST or SOAP web service, *XML HTTP Transactions* have to be used as these web services are XML-based. To consume a JSON web service, *JSON HTTP Transactions* have to be used as it performs the conversion from the JSON data to the transaction output XML. To retrieve any other data in HTTP protocol (non XML-based REST web service, image or file getting, etc.), standard *HTTP Transaction* has to be used as its response is text-based.

Note: *HTTP connector* supports OAuth authentication. To enable OAuth, you simply need to provide four variables to any kind of *HTTP transaction*: `__header_oAuthKey`, `__header_oAuthSecret`, `__header_oAuthToken` and `__header_oAuthTokenSecret`. For more information about OAuth in *HTTP connector*, refer to the following article in our Technical Blog: *http://www.convertigo.com/en/how-to/technical-blog/entry/using-oauth-*



3.3 Create an HTTP connector

Choose a name for the connector, and click on **Finish**.

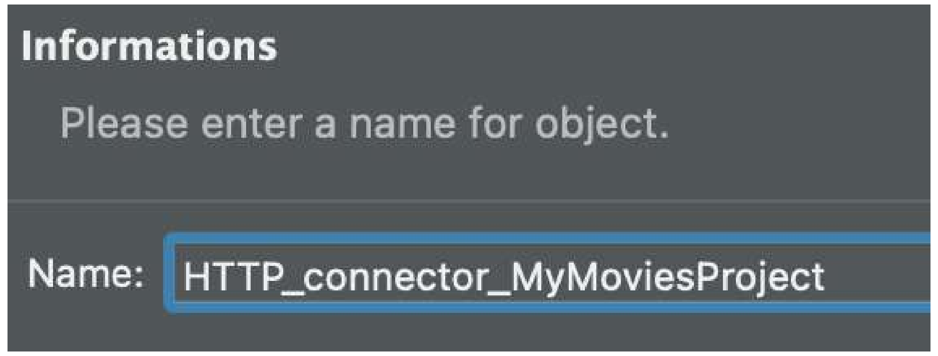


Create a new object

Informations

Please enter a name for object.

Name: HTTP_connector_MyMoviesProject

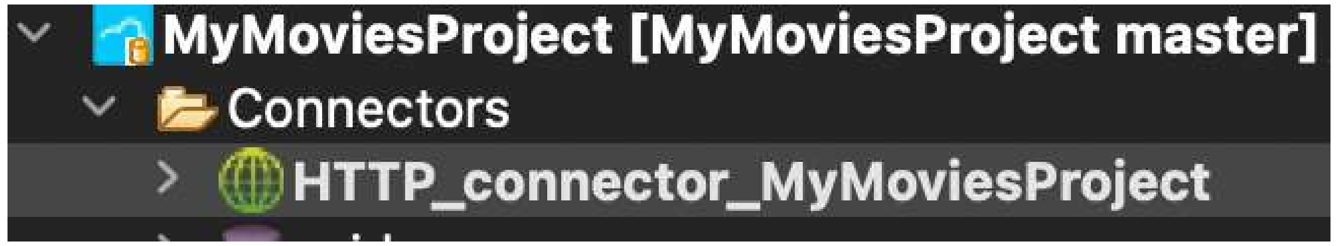


Informations

Please enter a name for object.

Name: HTTP_connector_MyMoviesProject

The new connector is created in the **Connectors** folder.



MyMoviesProject [MyMoviesProject master]

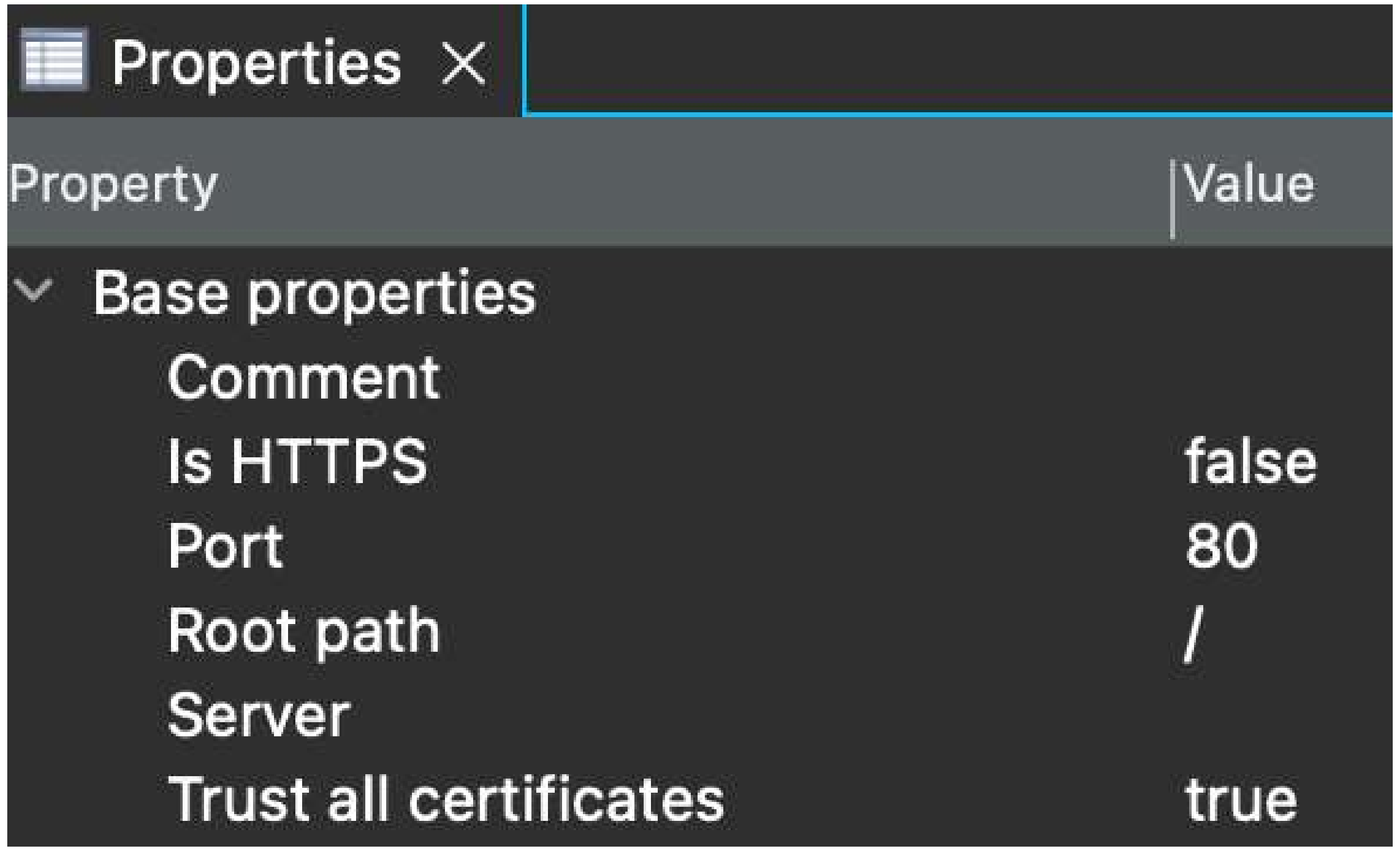
Connectors

HTTP_connector_MyMoviesProject

3.4 Configure the HTTP connector

In the **Properties window**,

you will find the **default properties** of the connector.



Property	Value
Base properties	
Comment	
Is HTTPS	false
Port	80
Root path	/
Server	
Trust all certificates	true

For **http requests**

- IsHTTPS : false
- Port : 80

For **https requests**

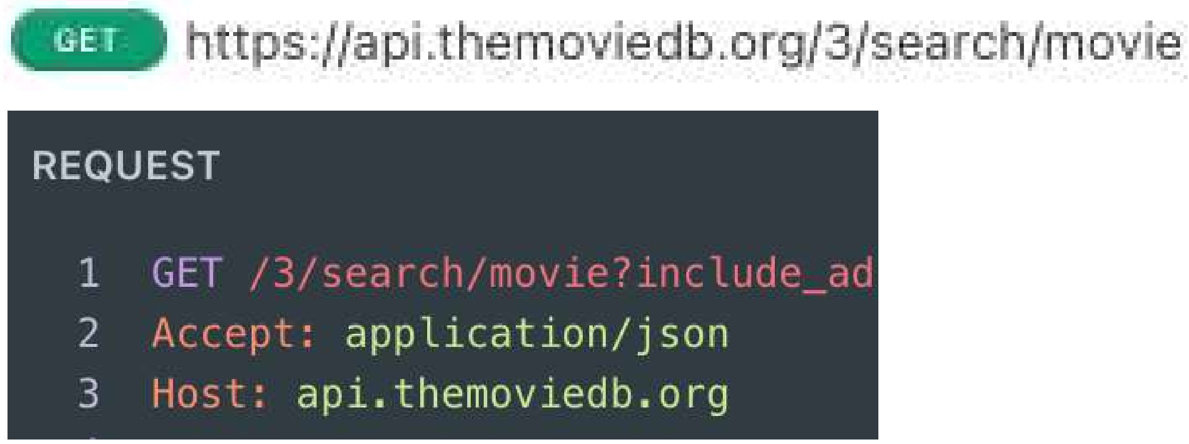
- IsHTTPS : true
- Port : 443

Root path : / (default path)

Server : => enter a server name

3.4 Configure the HTTP connector

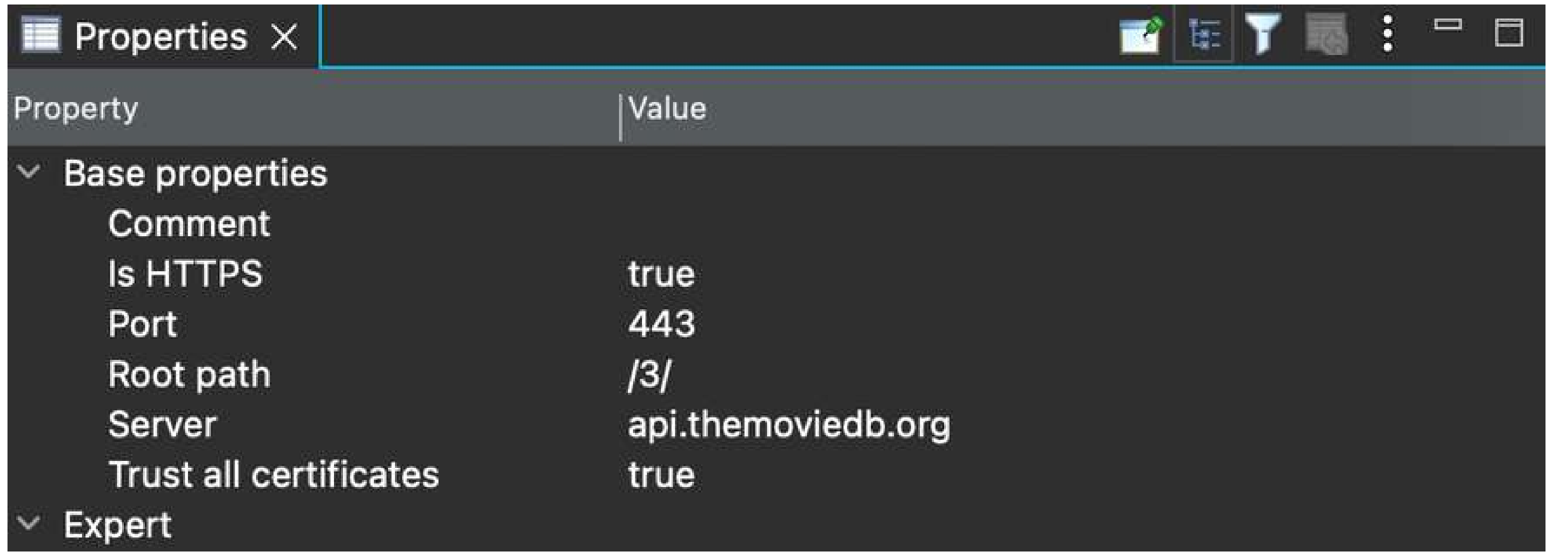
Now, we need to configure the connector
with the informations found in the TMDB API documentation.



GET https://api.themoviedb.org/3/search/movie

REQUEST

```
GET /3/search/movie?include_ad
Accept: application/json
Host: api.themoviedb.org
```



Property	Value
Base properties	
Comment	
Is HTTPS	true
Port	443
Root path	/3/
Server	api.themoviedb.org
Trust all certificates	true
Expert	

In the TMDB API documentation, we can see that:

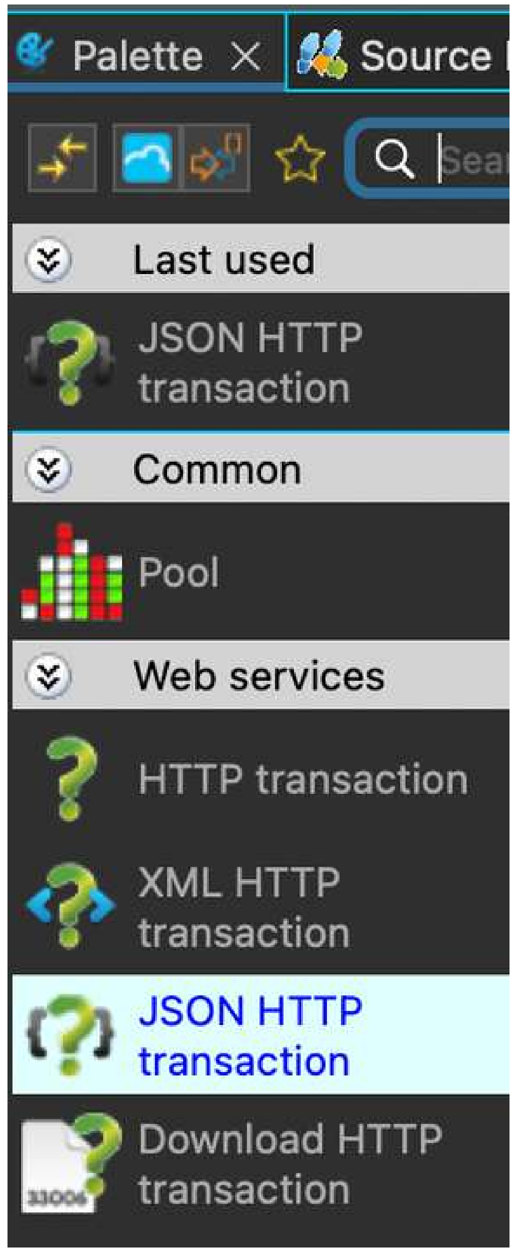
- the request is **https**
- the request has a root path : **/3/** (version 3 of the API)
- the domain name is **api.themoviedb.org**

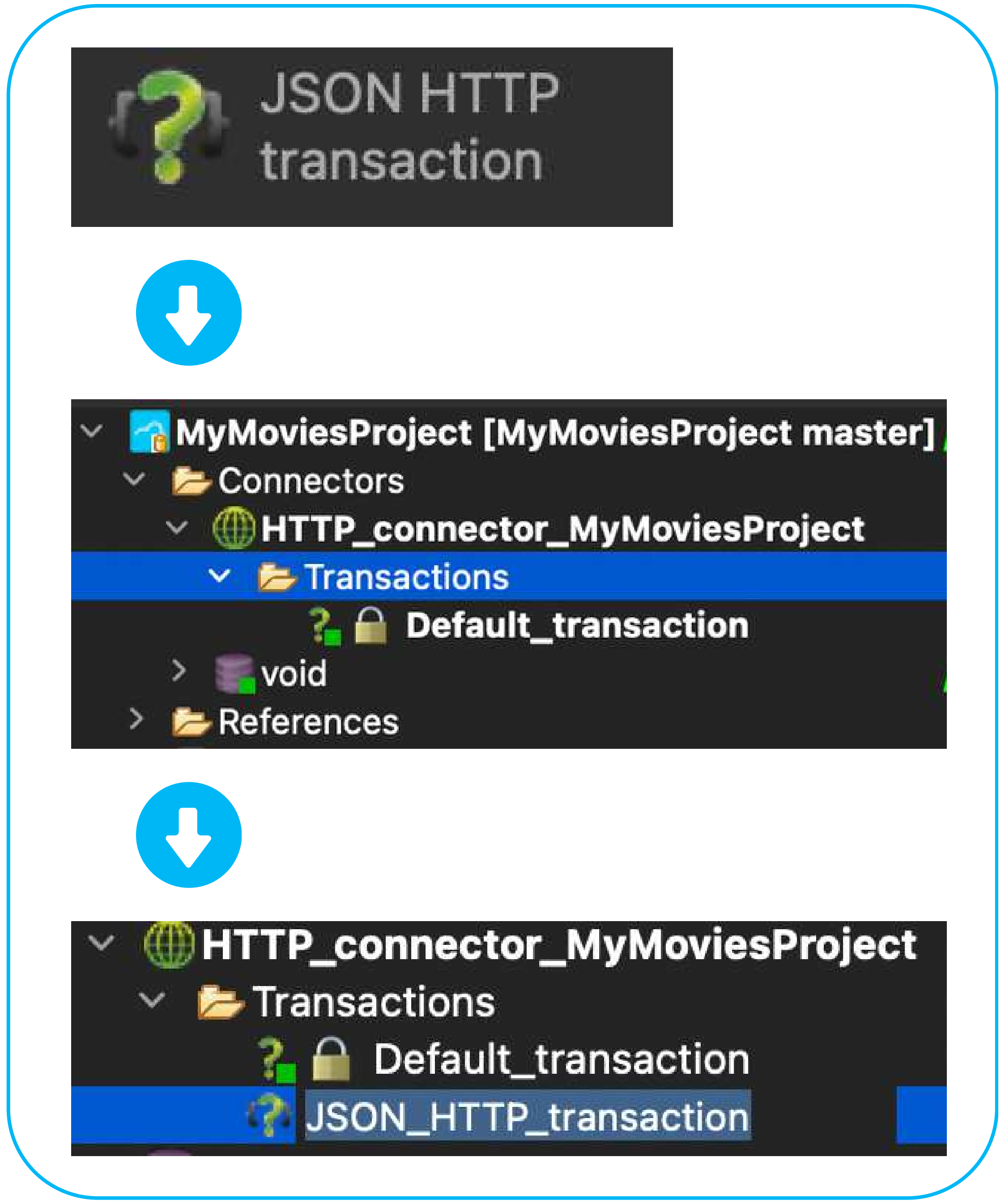
As a result, the **Connector configuration** is

- IsHTTPS : true
- Port : 443
- Root path : /3/
- Server : api.themoviedb.org

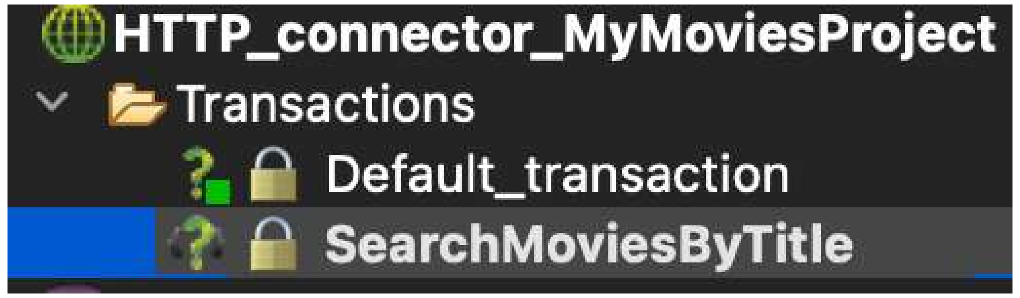
3.5 Create a transaction

First option : Drag and drop a **JSON HTTP transaction** from the palette into the **Connectors folder**.



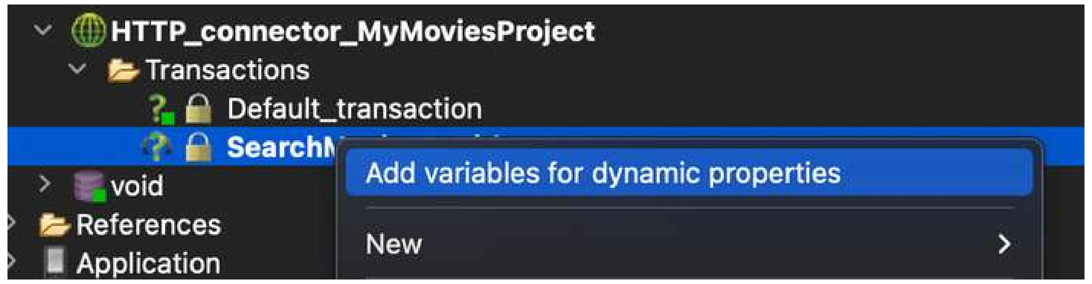


Rename the transaction to **SearchMoviesByTitle**.



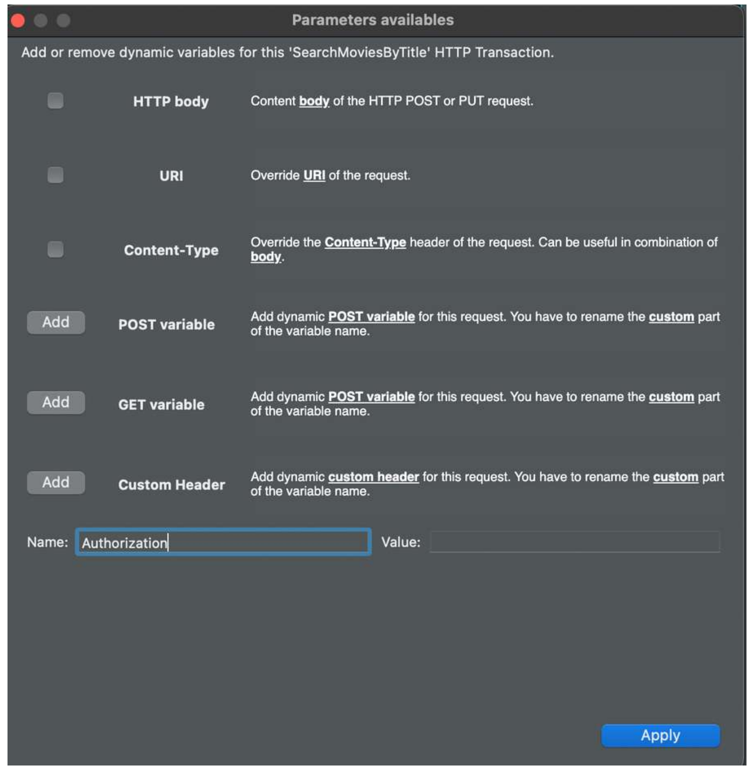
3.5 Create a transaction

Right-click on the transaction, and select **Add variables for dynamic properties**.



HTTP_connector_MyMoviesProject
Transactions
Default_transaction
SearchM
void
References
Application
Add variables for dynamic properties
New

The **Parameters available window** appears.



Parameters availables

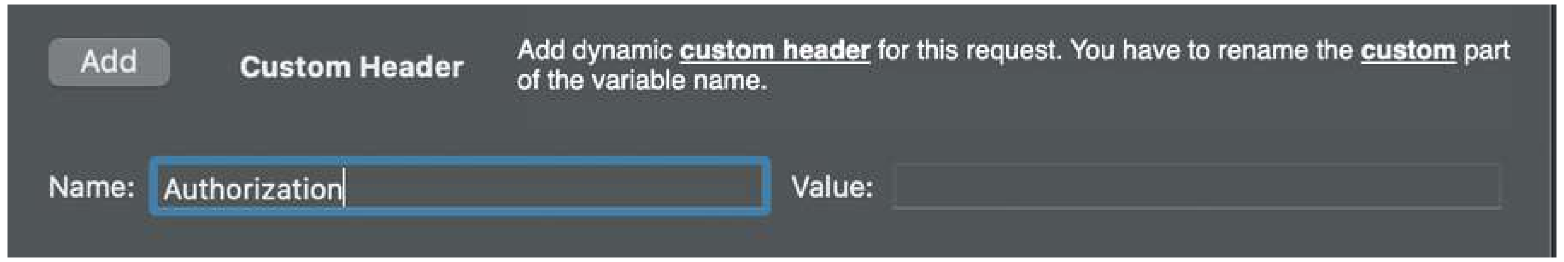
Add or remove dynamic variables for this 'SearchMoviesByTitle' HTTP Transaction.

□	HTTP body	Content **body** of the HTTP POST or PUT request.
□	URI	Override **URI** of the request.
□	Content-Type	Override the **Content-Type** header of the request. Can be useful in combination of **body**.
Add	POST variable	Add dynamic **POST variable** for this request. You have to rename the **custom** part of the variable name.
Add	GET variable	Add dynamic **POST variable** for this request. You have to rename the **custom** part of the variable name.
Add	Custom Header	Add dynamic **custom header** for this request. You have to rename the **custom** part of the variable name.

Name: Authorization Value:

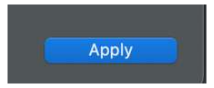
Apply

In the **Parameters available window**, click on **Add Custom Header** to add the **Authorization** header (which will allow sending the access token),

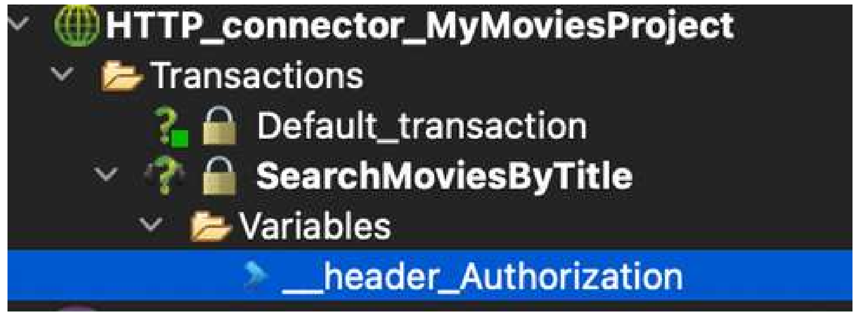


Add **Custom Header** Add dynamic **custom header** for this request. You have to rename the **custom** part of the variable name.

Name: Authorization Value:



Then click on **Apply**.



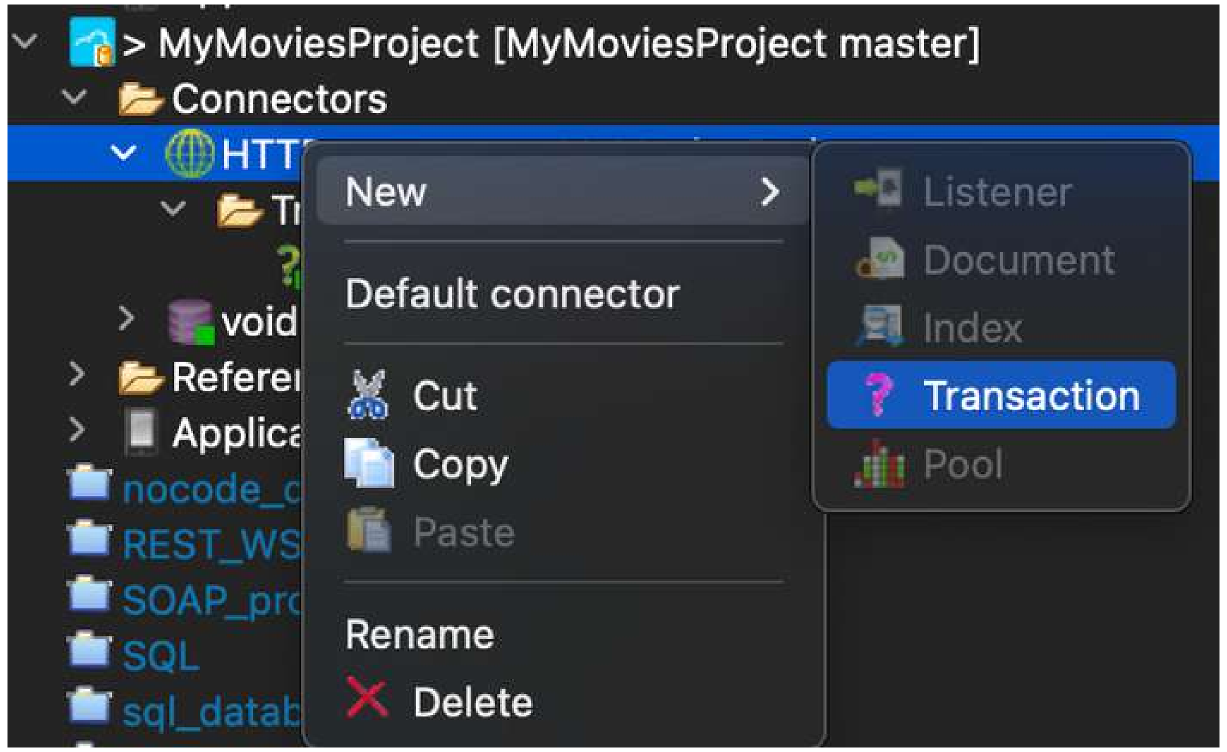
The **Authorization** header will appear in the folder **Variables** of the transaction

3.5 Create a transaction

Second Option :

Right-click on the connector,

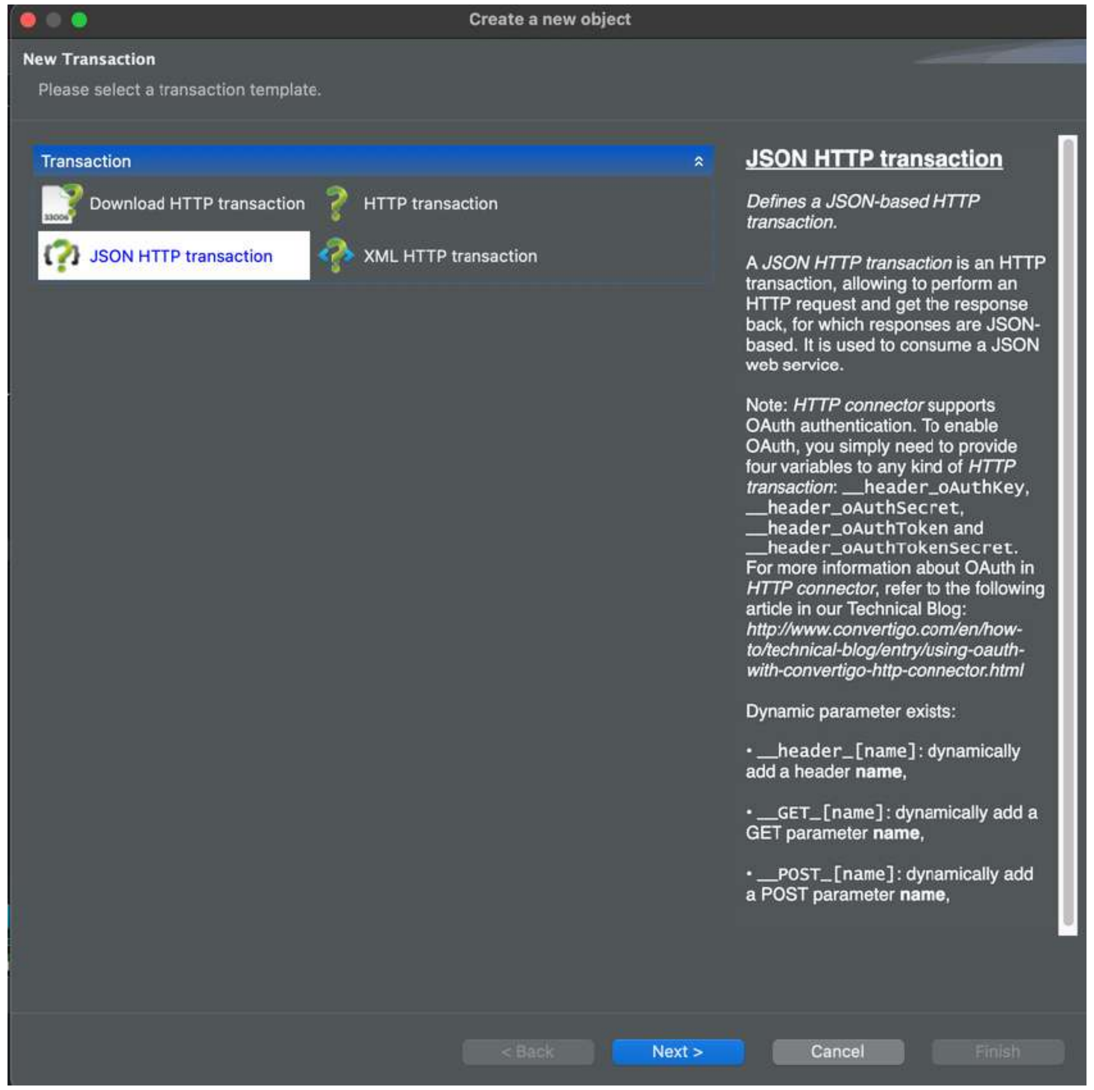
then select **New >**, then select **Transaction**.



In the **Create a new object window**,

choose **JSON HTTP transaction**,

then click on **Next >**.



JSON HTTP transaction

Defines a JSON-based HTTP transaction.

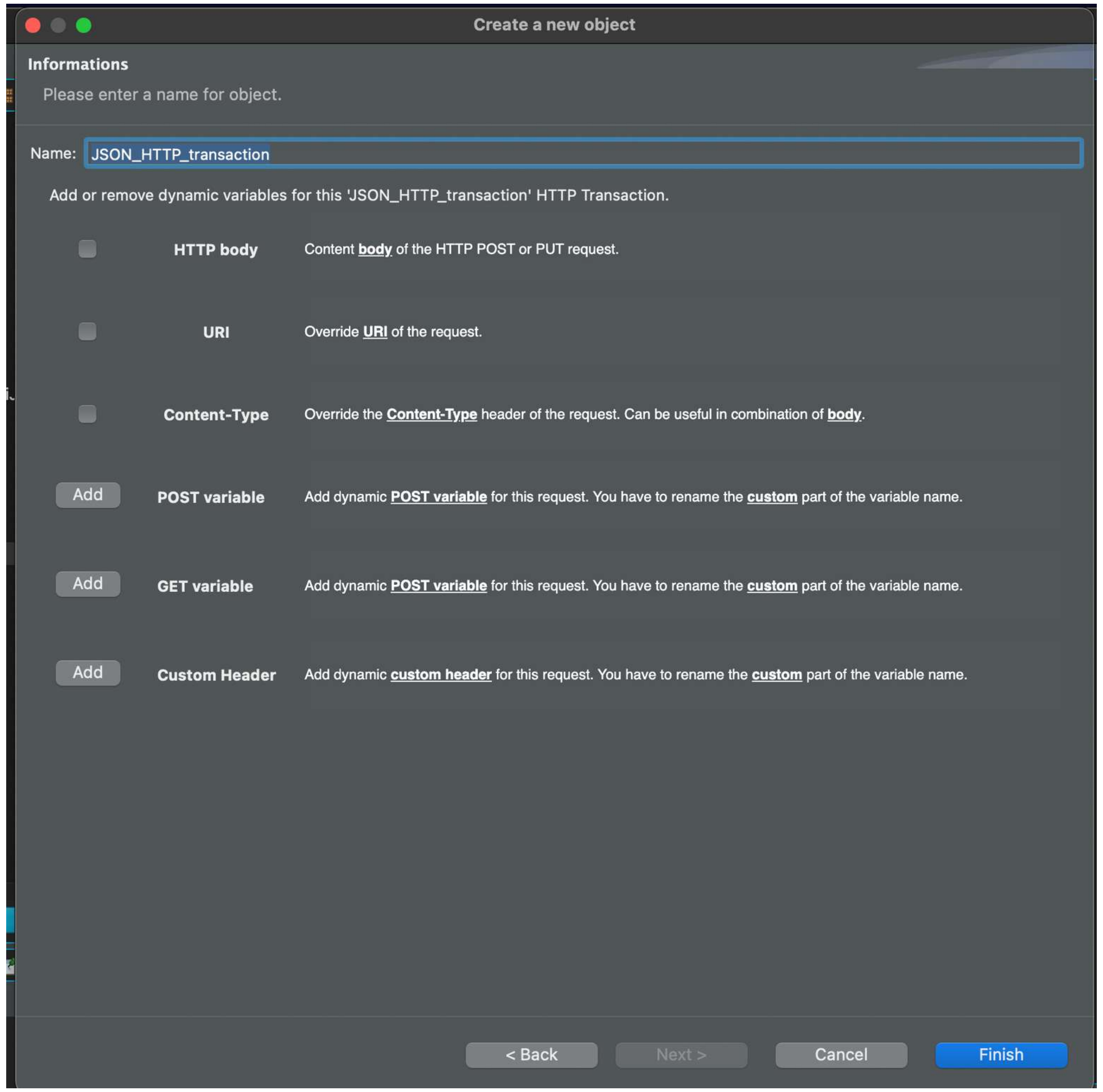
A *JSON HTTP transaction* is an HTTP transaction, allowing to perform an HTTP request and get the response back, for which responses are JSON-based. It is used to consume a JSON web service.

Note: *HTTP connector* supports OAuth authentication. To enable OAuth, you simply need to provide four variables to any kind of *HTTP transaction*: `__header_oAuthKey`, `__header_oAuthSecret`, `__header_oAuthToken` and `__header_oAuthTokenSecret`. For more information about OAuth in *HTTP connector*, refer to the following article in our Technical Blog: *http://www.convertigo.com/en/how-to/technical-blog/entry/using-oauth-with-convertigo-http-connector.html*

Dynamic parameter exists:

- `__header_[name]`: dynamically add a header **name**,
- `__GET_[name]`: dynamically add a GET parameter **name**,
- `__POST_[name]`: dynamically add a POST parameter **name**,

3.5 Create a transaction



Create a new object

Informations

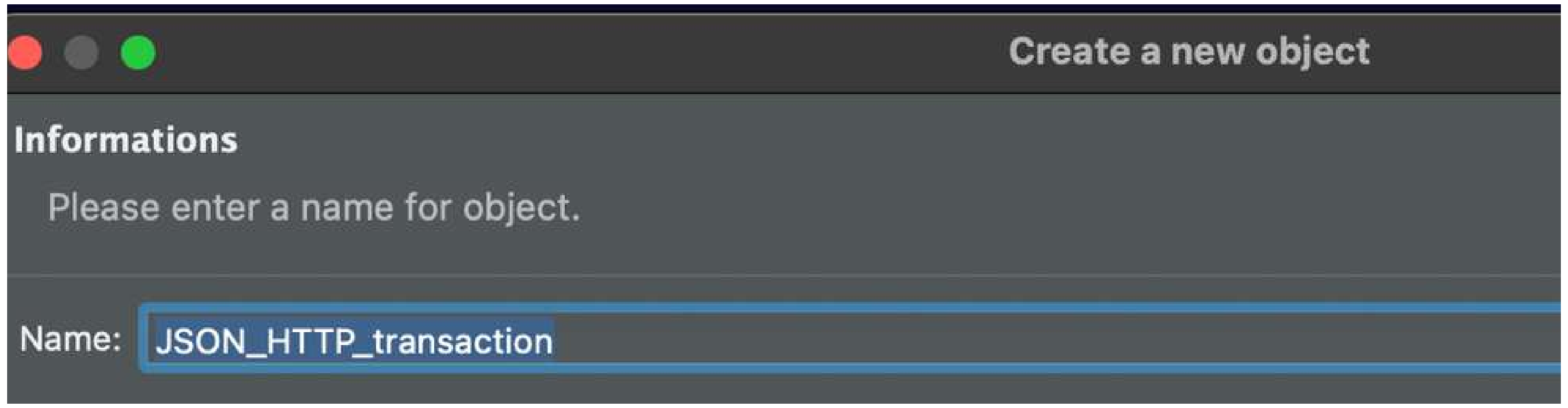
Please enter a name for object.

Name: JSON_HTTP_transaction

Add or remove dynamic variables for this 'JSON_HTTP_transaction' HTTP Transaction.

□	**HTTP body**	Content **body** of the HTTP POST or PUT request.
□	**URI**	Override **URI** of the request.
□	**Content-Type**	Override the **Content-Type** header of the request. Can be useful in combination of **body**.
Add	**POST variable**	Add dynamic **POST variable** for this request. You have to rename the **custom** part of the variable name.
Add	**GET variable**	Add dynamic **POST variable** for this request. You have to rename the **custom** part of the variable name.
Add	**Custom Header**	Add dynamic **custom header** for this request. You have to rename the **custom** part of the variable name.

Rename the transaction with the name of the request.



Create a new object

Informations

Please enter a name for object.

Name: JSON_HTTP_transaction

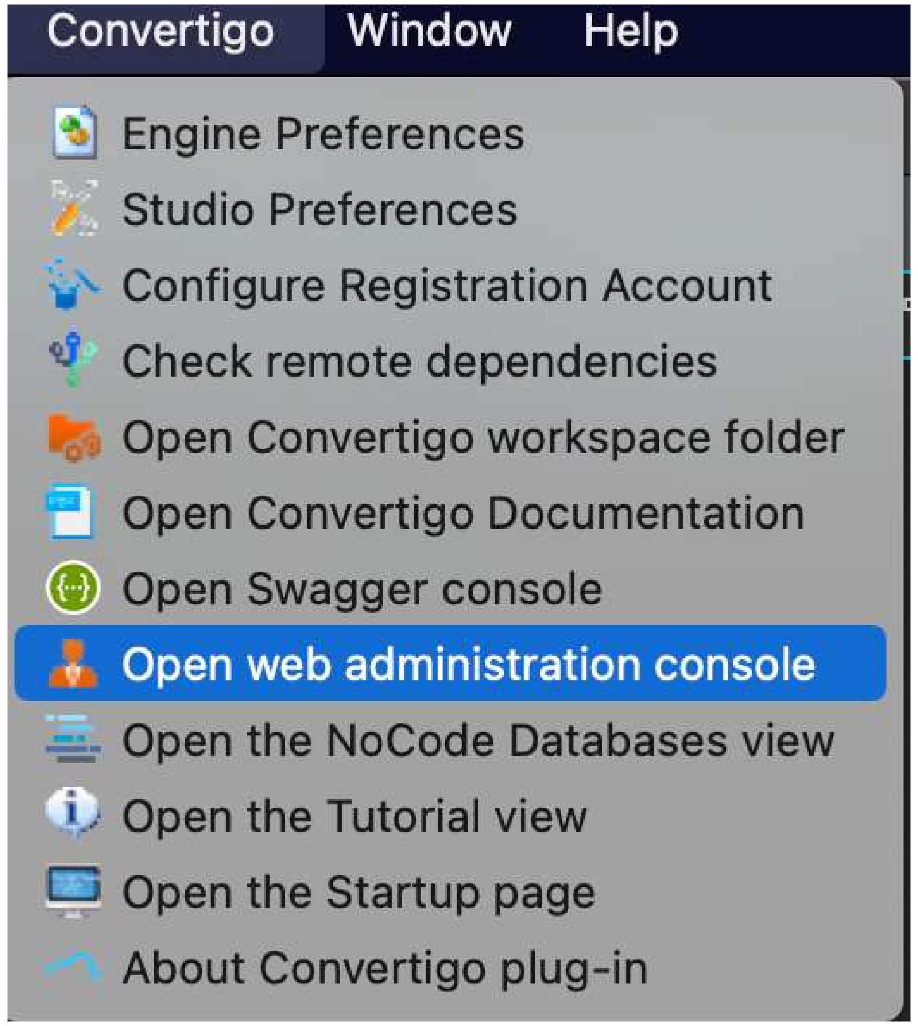
Then, follow the same steps as in the first option.

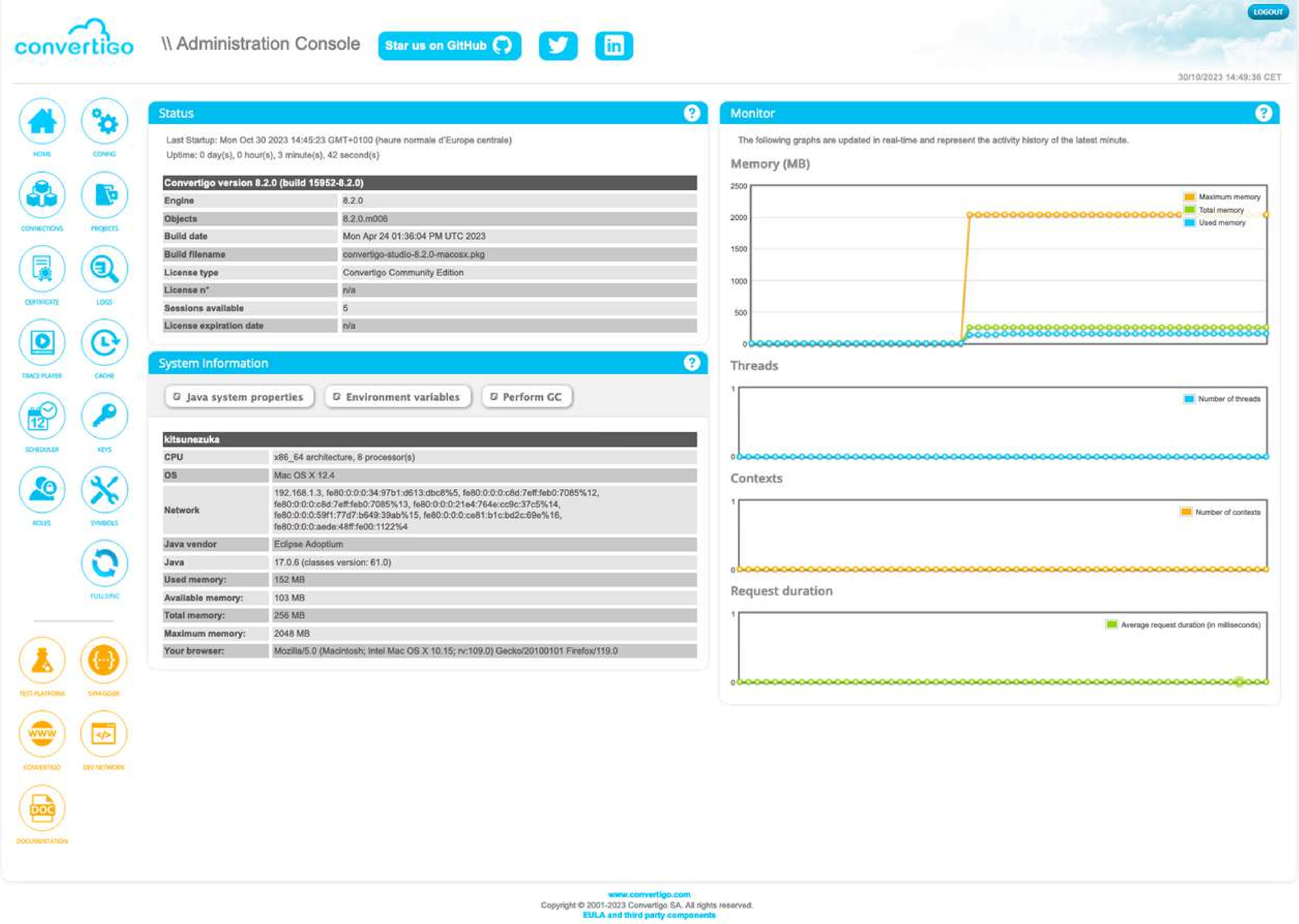
3.6 Add a token

To open the **web administration console**,

click on **Convertigo**,

then select **Open web administration console**.


Convertigo
Window
Help
Engine Preferences
Studio Preferences
Configure Registration Account
Check remote dependencies
Open Convertigo workspace folder
Open Convertigo Documentation
Open Swagger console
Open web administration console
Open the NoCode Databases view
Open the Tutorial view
Open the Startup page
About Convertigo plug-in



Status

Last Startup: Mon Oct 30 2023 14:45:23 GMT+0100 (heure normale d'Europe centrale)
Uptime: 0 day(s), 0 hour(s), 3 minute(s), 42 second(s)

Convertigo version 8.2.0 (build 15952-8.2.0)	
Engine	8.2.0
Objects	8.2.0.m006
Build date	Mon Apr 24 01:36:04 PM UTC 2023
Build filename	convertigo-studio-8.2.0-macosx.pkg
License type	Convertigo Community Edition
License n°	n/a
Sessions available	5
License expiration date	n/a

System Information

Java system properties | Environment variables | Perform GC

kitsunezuka	
CPU	x86_64 architecture, 8 processor(s)
OS	Mac OS X 12.4
Network	192.168.1.3, fe80:0:0:0:34:97b1:d613:dbc8%5, fe80:0:0:0:c8d:7eff:feb0:7085%12, fe80:0:0:0:c8d:7eff:feb0:7085%13, fe80:0:0:0:21e4:764e:cc9c:37c5%14, fe80:0:0:0:59f1:77d7:b649:39ab%15, fe80:0:0:0:ce81:b1c:bd2c:69e%16, fe80:0:0:0:aede:48ff:fe00:1122%4
Java vendor	Eclipse Adoptium
Java	17.0.6 (classes version: 61.0)
Used memory:	152 MB
Available memory:	103 MB
Total memory:	256 MB
Maximum memory:	2048 MB
Your browser:	Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:109.0) Gecko/20100101 Firefox/119.0

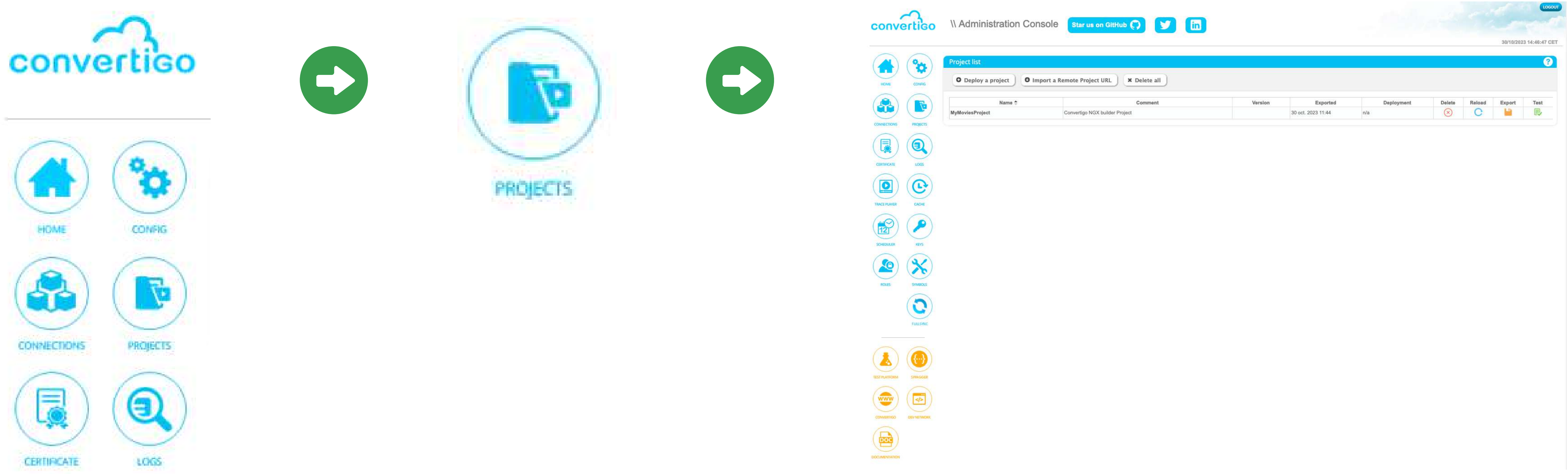
The following graphs are updated in real-time and represent the activity history of the latest minute.

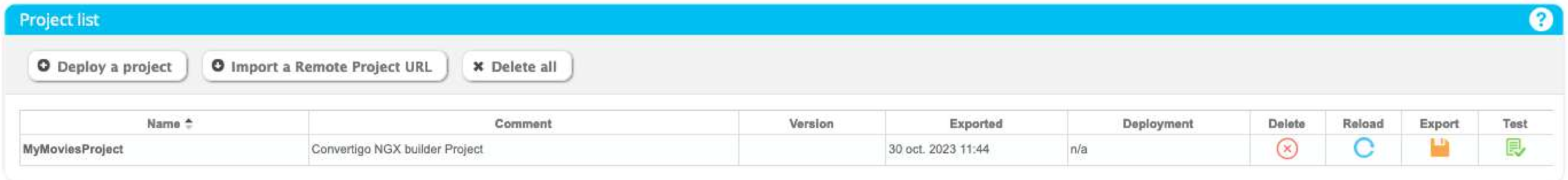
www.convertigo.com
Copyright © 2001-2023 Convertigo SA. All rights reserved.
EULA and third party components

3.6 Add a token

In the **web administration console**,

click on the icon **PROJECTS** to view the **projects currently opened** in the studio **workspace**.

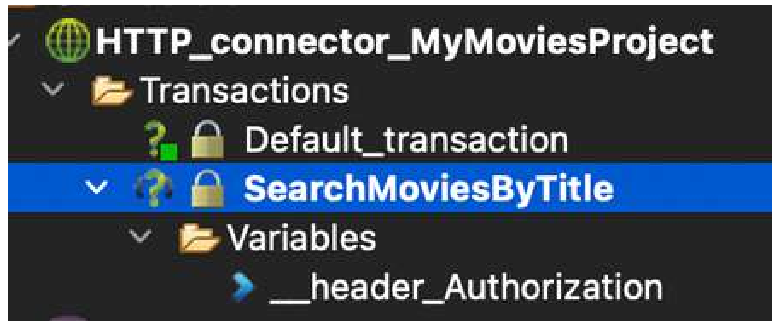




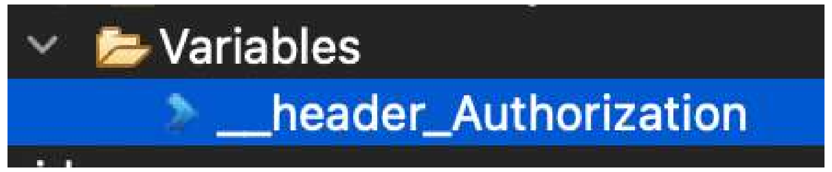
Name	Comment	Version	Exported	Deployment	Delete	Reload	Export	Test
MyMoviesProject	Convertigo NGX builder Project		30 oct. 2023 11:44	n/a	⊗	↻	💾	✓

3.6 Add a token

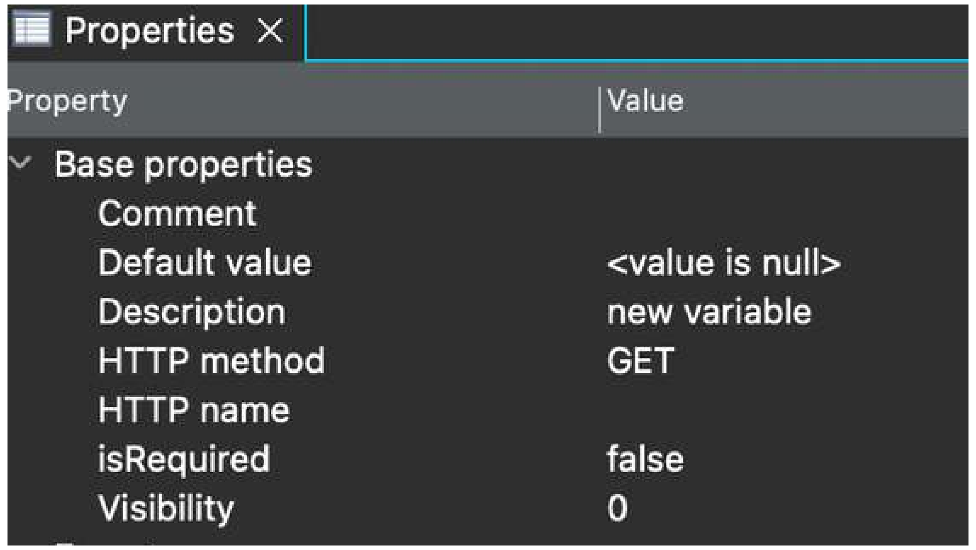
Let’s have a look on the properties of the variable **__header_Authorization.**



- HTTP_connector_MyMoviesProject
  - Transactions
    - Default_transaction
    - SearchMoviesByTitle
      - Variables
        - __header_Authorization



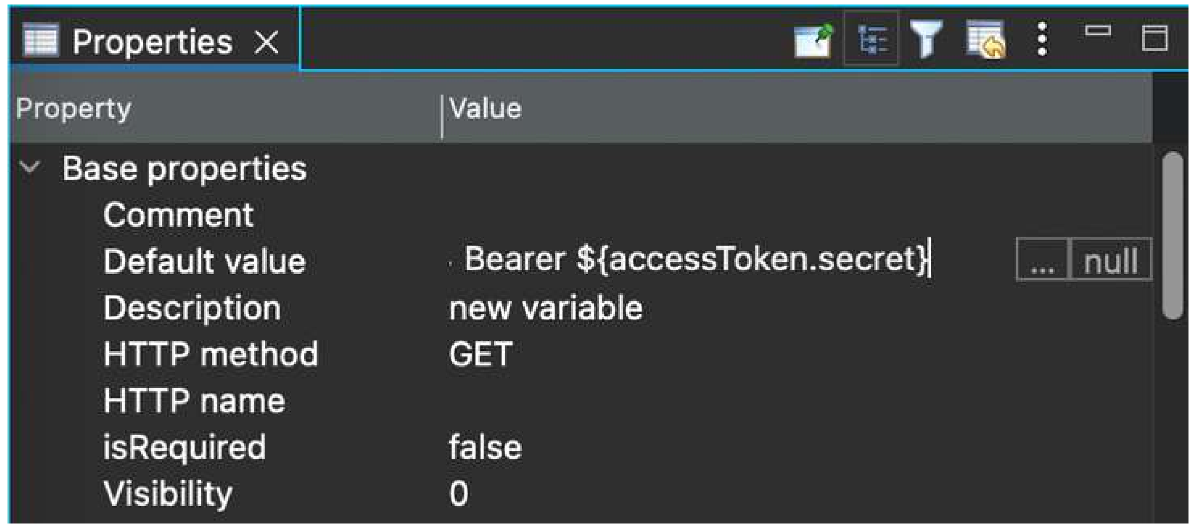
- Variables
  - __header_Authorization



Properties

Property	Value
Base properties	
Comment	
Default value	<value is null>
Description	new variable
HTTP method	GET
HTTP name	
isRequired	false
Visibility	0

As **Default value**, enter **Bearer ${accessToken.secret}**.

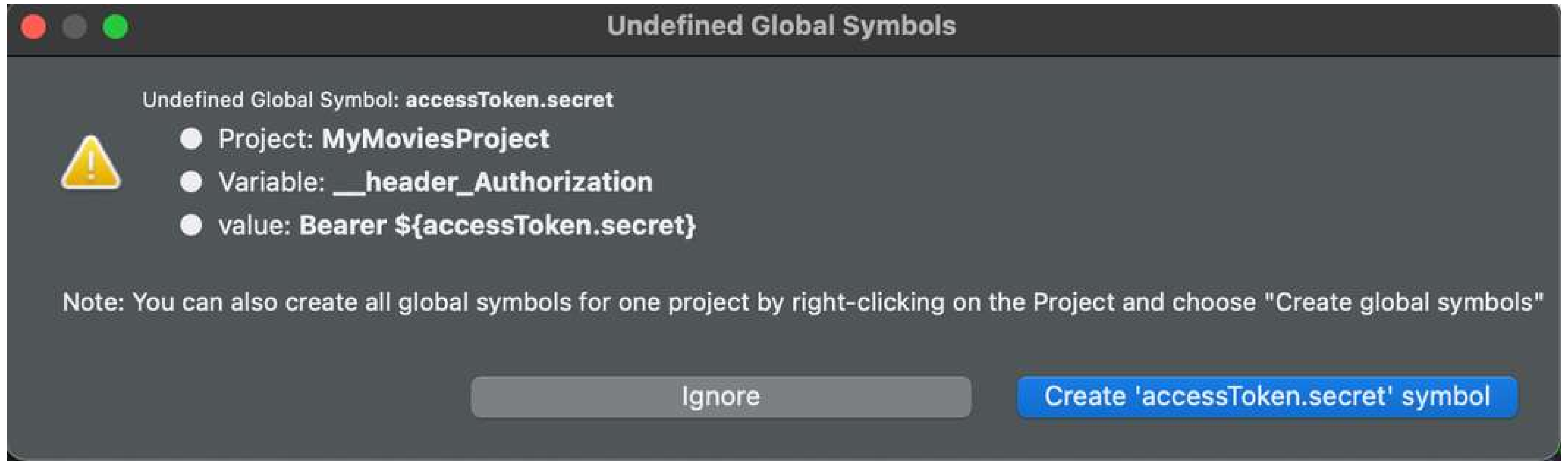


Properties

Property	Value
Base properties	
Comment	
Default value	Bearer ${accessToken.secret}
Description	new variable
HTTP method	GET
HTTP name	
isRequired	false
Visibility	0

Validating the value will open the **Undefined Global Symbols window.**

Click on **Create ‘accessToken.secret’ symbol**.



Undefined Global Symbols

Undefined Global Symbol: **accessToken.secret**

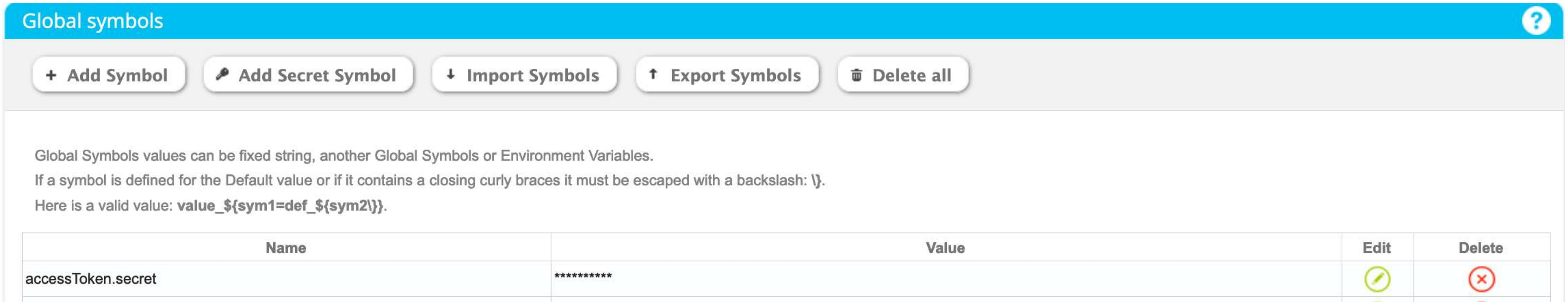
- Project: **MyMoviesProject**
- Variable: **__header_Authorization**
- value: **Bearer ${accessToken.secret}**

Note: You can also create all global symbols for one project by right-clicking on the Project and choose "Create global symbols"

3.6 Add a token

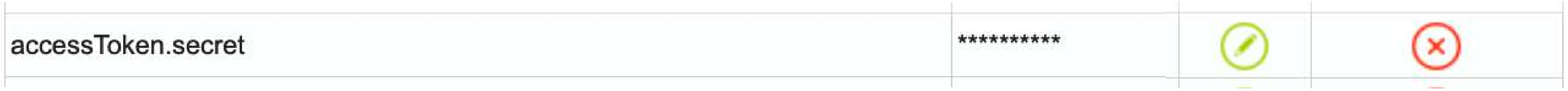
In the web administration console,

click on **Symbols** to access the Global symbols.

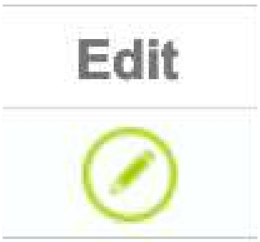


Global Symbols values can be fixed string, another Global Symbols or Environment Variables.
If a symbol is defined for the Default value or if it contains a closing curly braces it must be escaped with a backslash: \}.
Here is a valid value: **value_${sym1=def_${sym2\}}**.

Name	Value	Edit	Delete
accessToken.secret	**********	✎	⊗



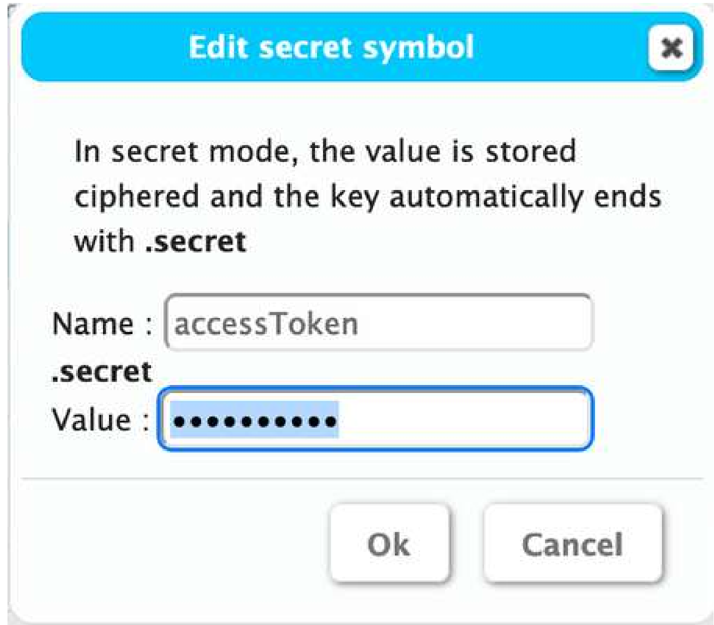
accessToken.secret	***********	✎	⊗



Click on **Edit** to open the **Edit symbol window**.

3.6 Add a token

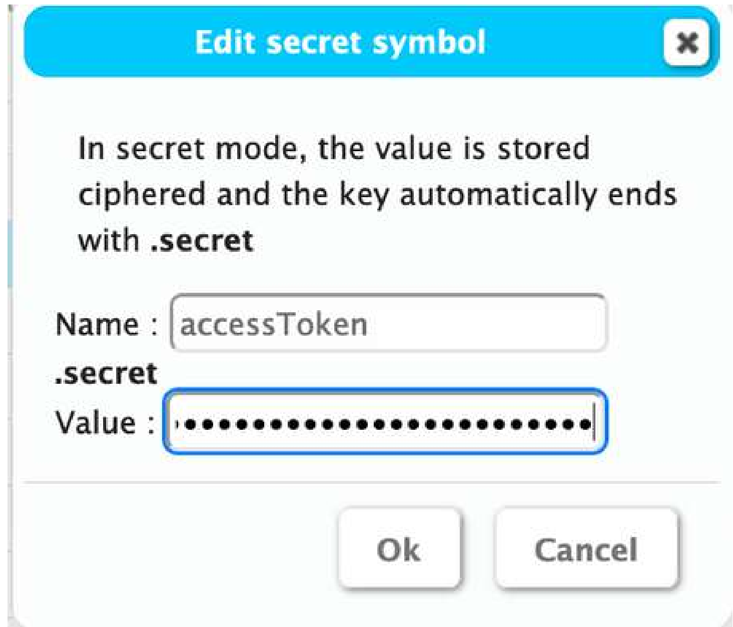
In the **Edit symbol window**, change the **accessToken.secret** value.



In secret mode, the value is stored ciphered and the key automatically ends with **.secret**

Name : accessToken .secret

Value : ••••••••••



In secret mode, the value is stored ciphered and the key automatically ends with **.secret**

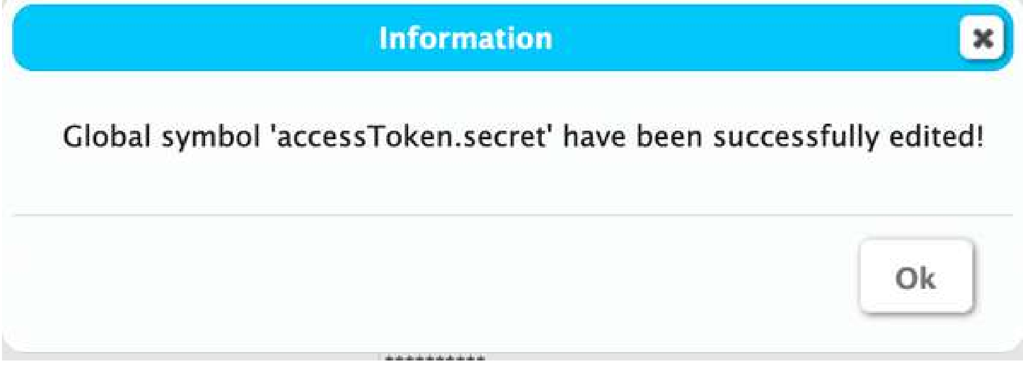
Name : accessToken .secret

Value : ••••••••••••••••••••••••

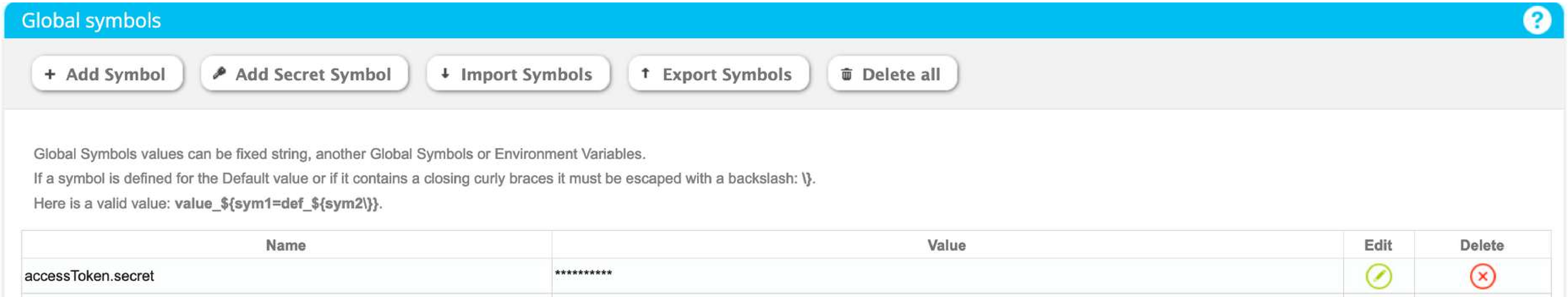
In the **Information window**,

a message confirms the changes

in the accessToken.secret value.



Global symbol 'accessToken.secret' have been successfully edited!



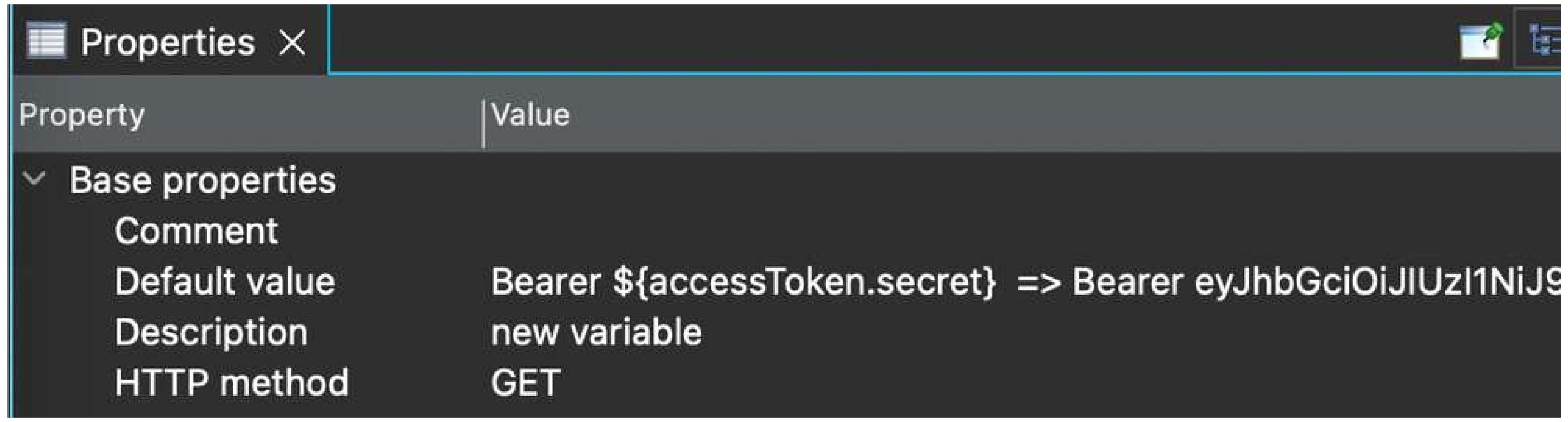
Global Symbols values can be fixed string, another Global Symbols or Environment Variables.
If a symbol is defined for the Default value or if it contains a closing curly braces it must be escaped with a backslash: **\}**.
Here is a valid value: **value_${sym1=def_${sym2\}}**.

Name	Value	Edit	Delete
accessToken.secret	***********	✎	⊗

3.6 Add a token

In the studio, the value of the symbol appears in clear

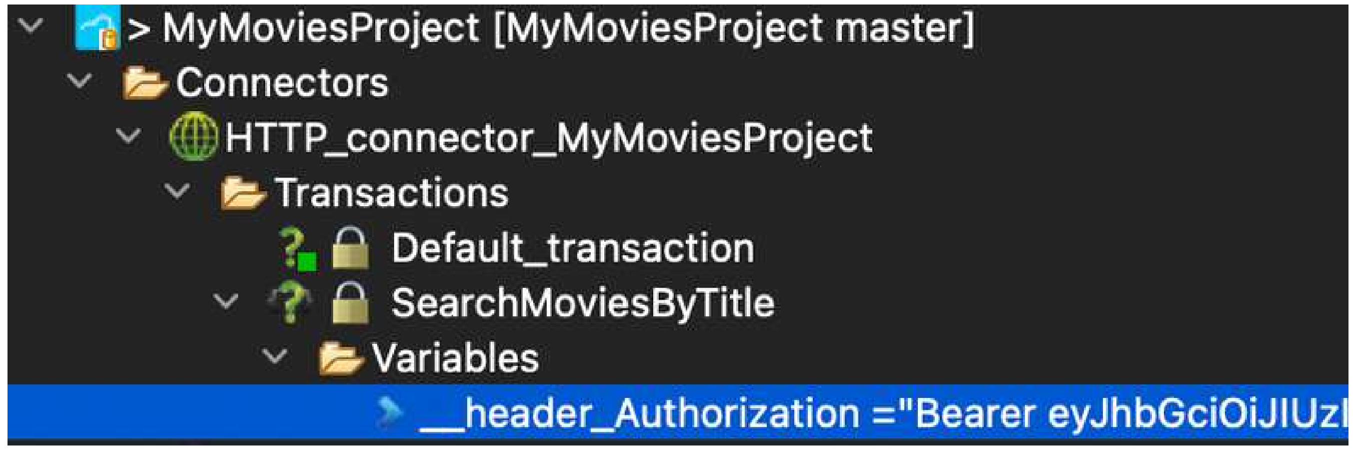
in the Properties of **__header_Authorization**



Properties

Property	Value
Base properties	
Comment	
Default value	Bearer ${accessToken.secret} => Bearer eyJhbGciOiJIUzI1NiJ9
Description	new variable
HTTP method	GET

and in the treeview's variable _header_Authorization.



```
> MyMoviesProject [MyMoviesProject master]
  Connectors
    HTTP_connector_MyMoviesProject
      Transactions
        Default_transaction
        SearchMoviesByTitle
          Variables
            __header_Authorization ="Bearer eyJhbGciOiJIUzI
```

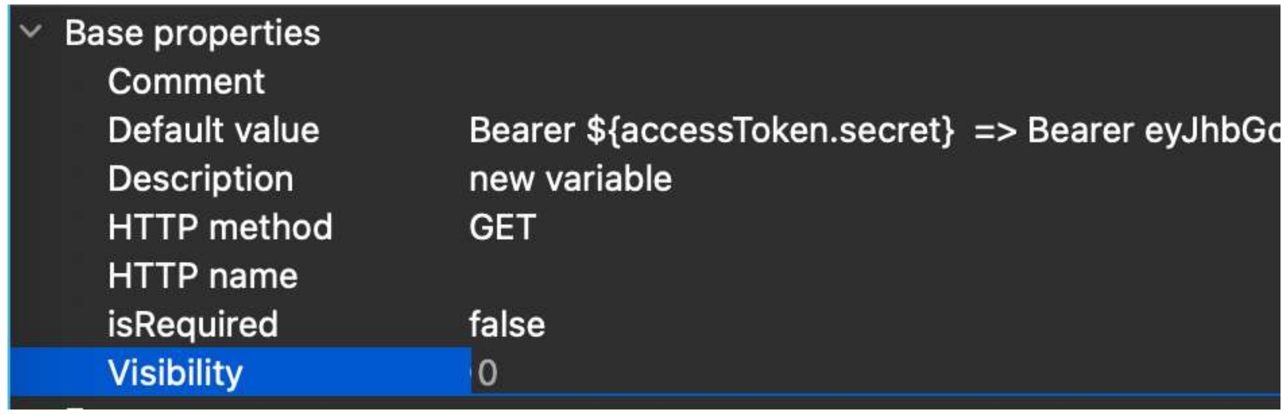
For security purposes,

the value of the symbol MUST BE HIDDEN.

3.6 Add a token

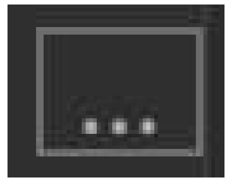
To hide the value of the symbol,

we need to change the **Visibility property**.



Base properties	
Comment	
Default value	Bearer ${accessToken.secret} => Bearer eyJhbGc
Description	new variable
HTTP method	GET
HTTP name	
isRequired	false
Visibility	0

Visibility 0



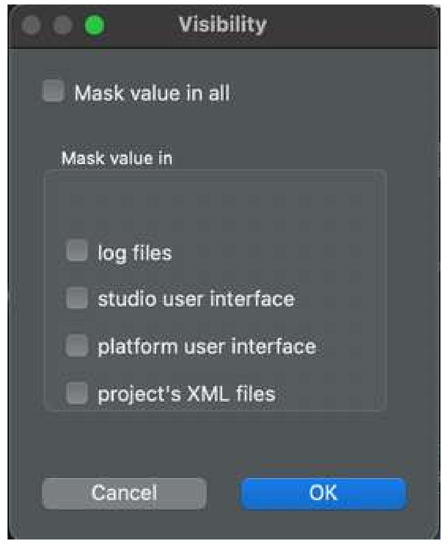
Click on the icon

at the end of

the **Visibility property** line.

The **Visibility windows** appears,

click on **Mask value in all**, then click on **OK**.



Visibility

□ Mask value in all

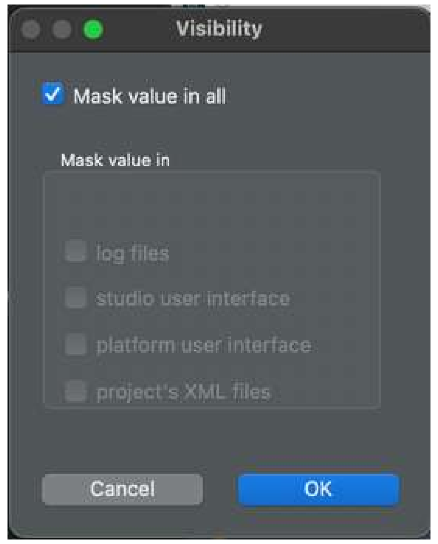
Mask value in

□ log files

□ studio user interface

□ platform user interface

□ project's XML files



Visibility

☑ Mask value in all

Mask value in

□ log files

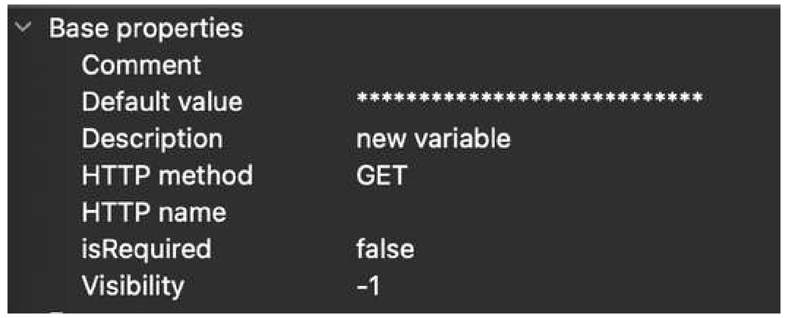
□ studio user interface

□ platform user interface

□ project's XML files

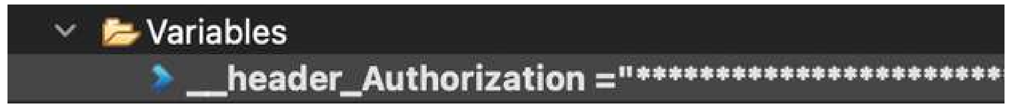
The value of **__header_Authorization** is hidden.

In Properties



Base properties	
Comment	
Default value	******************************
Description	new variable
HTTP method	GET
HTTP name	
isRequired	false
Visibility	-1

And in the variables folder



Variables

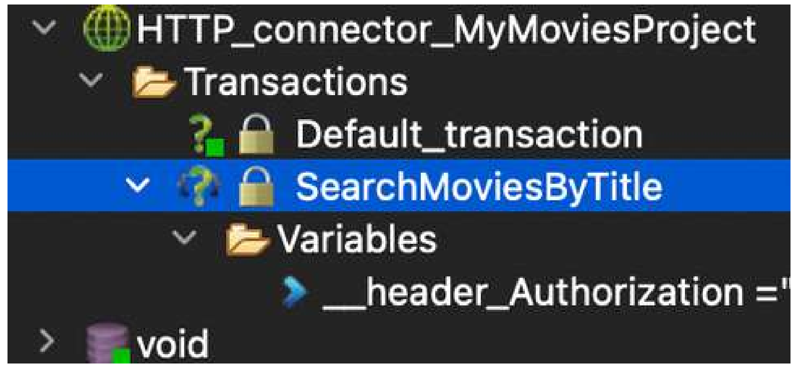
__header_Authorization ="***********************

3.7 Edit the request path

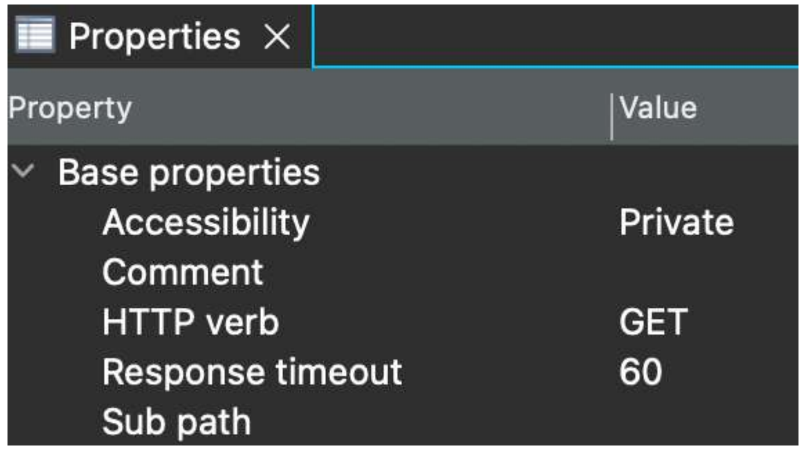
In the Properties of the transaction,

edit the Sub path to include the **request path**:

search/movie?query={movieTitle}&include_adult=false&language=en-US&page=1 (as seen in the TMDB API doc)

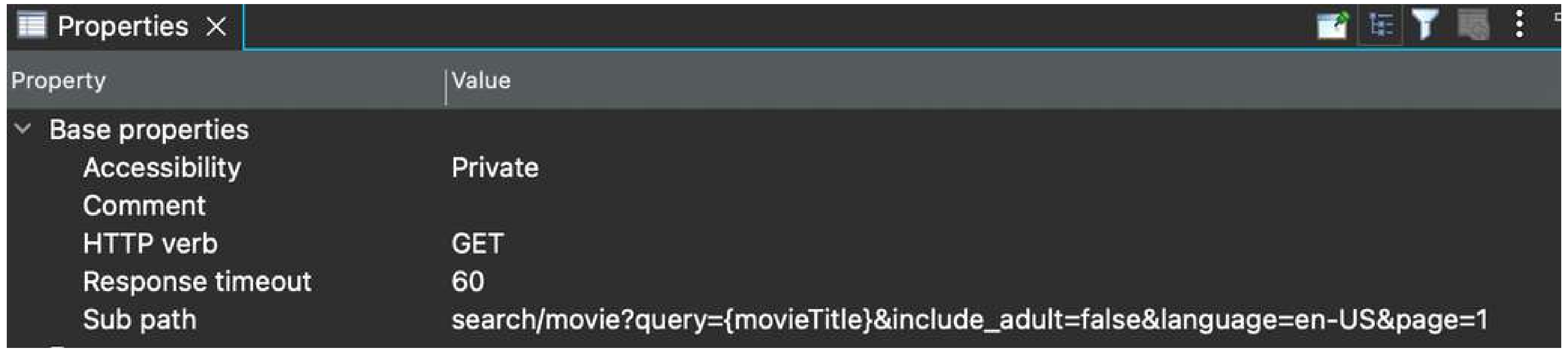


HTTP_connector_MyMoviesProject
Transactions
Default_transaction
SearchMoviesByTitle
Variables
__header_Authorization ="
void



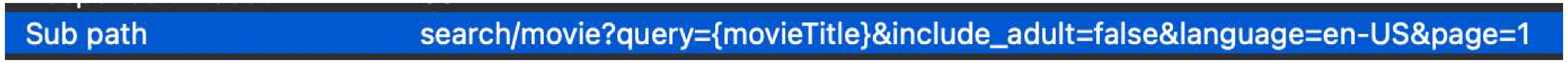
Property	Value
Base properties	
Accessibility	Private
Comment	
HTTP verb	GET
Response timeout	60
Sub path	

Sub path

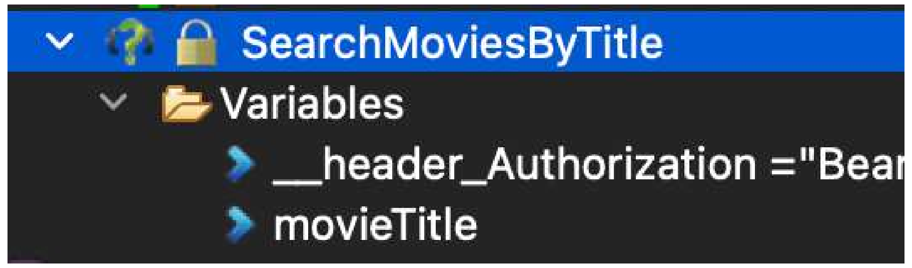


Property	Value
Base properties	
Accessibility	Private
Comment	
HTTP verb	GET
Response timeout	60
Sub path	search/movie?query={movieTitle}&include_adult=false&language=en-US&page=1

To add a **variable part**, enclose it in **curly braces within the path** (in our case, {movieTitle}).



Sub path search/movie?query={movieTitle}&include_adult=false&language=en-US&page=1



SearchMoviesByTitle
Variables
__header_Authorization ="Bear
movieTitle

This **automatically adds the variable** to the **Variables folder**.

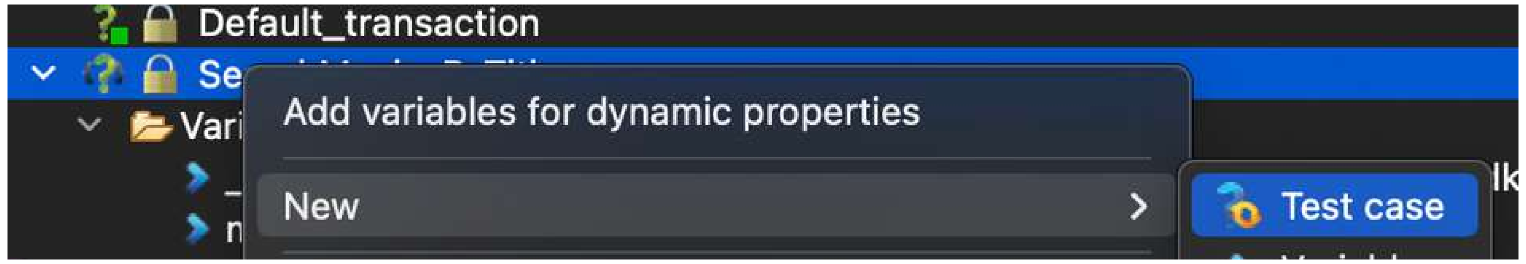
3.8 Test the request

To test the request, you need to create a test case.

Right-click on the transaction,

Select **New >**,

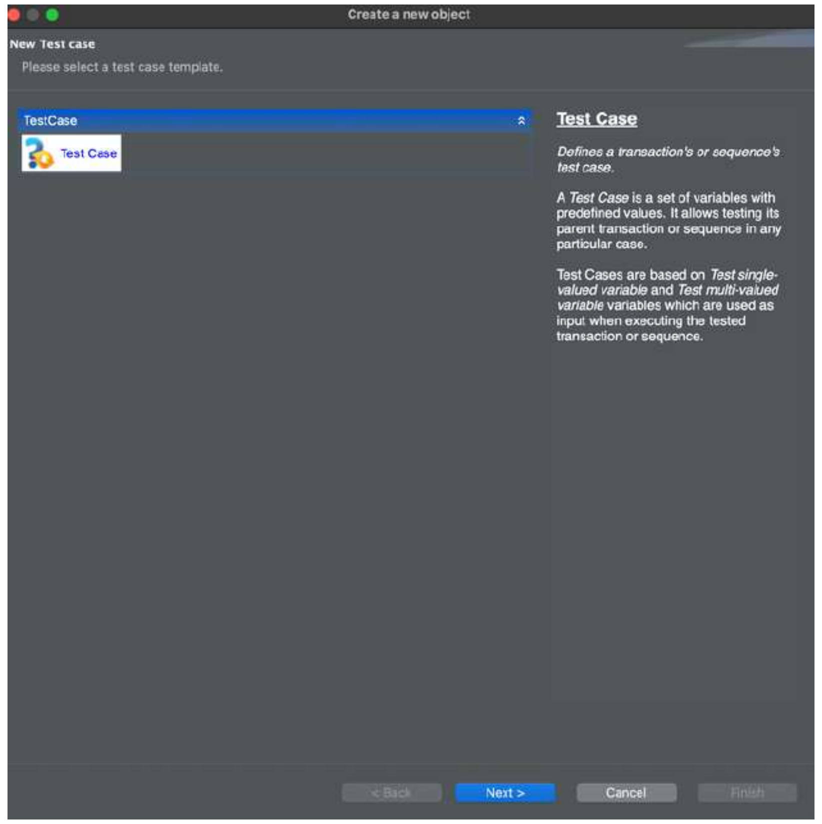
then click on **Test Case**.



In the **Create a new object window**,

select **Test Case**

and click **Next.**

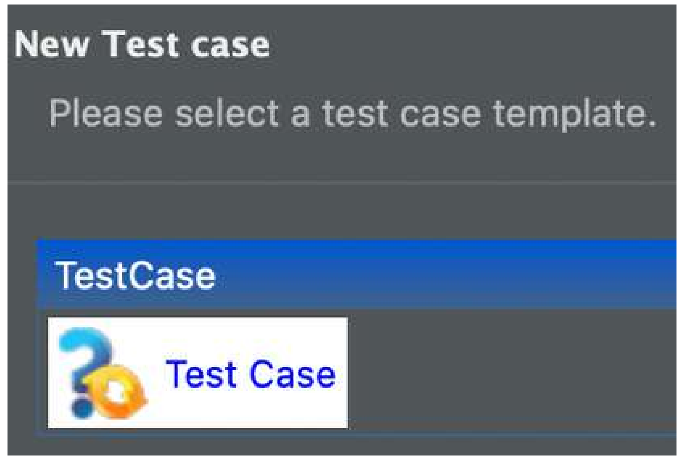


Test Case

Defines a transaction's or sequence's test case.

A *Test Case* is a set of variables with predefined values. It allows testing its parent transaction or sequence in any particular case.

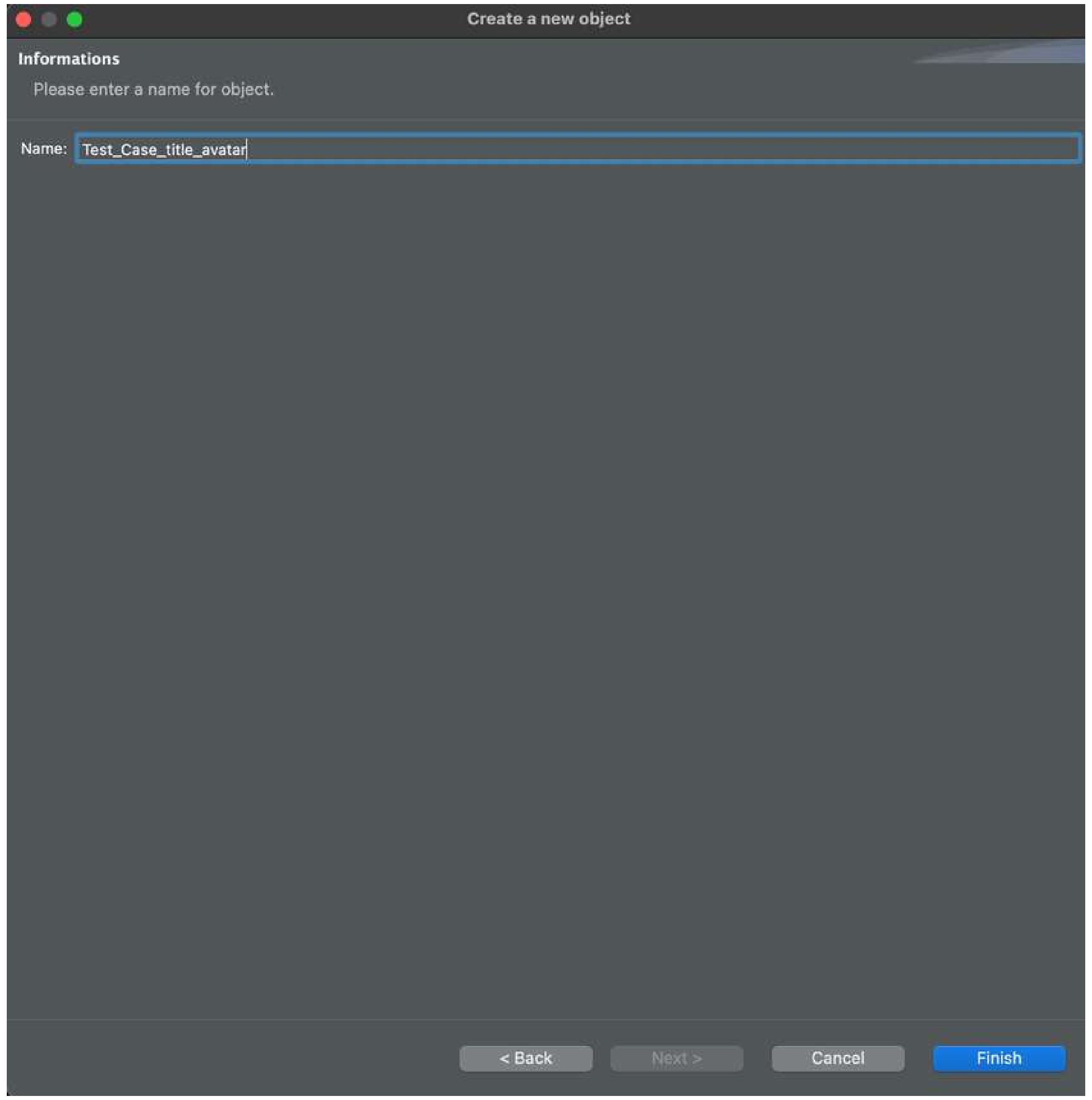
Test Cases are based on *Test single-valued variable* and *Test multi-valued variable* variables which are used as input when executing the tested transaction or sequence.



3.8 Test the request

Then, enter a name for the test case,

and click **Finish**.

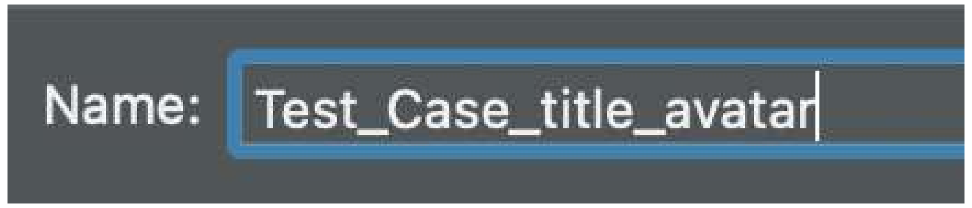


Create a new object

Informations

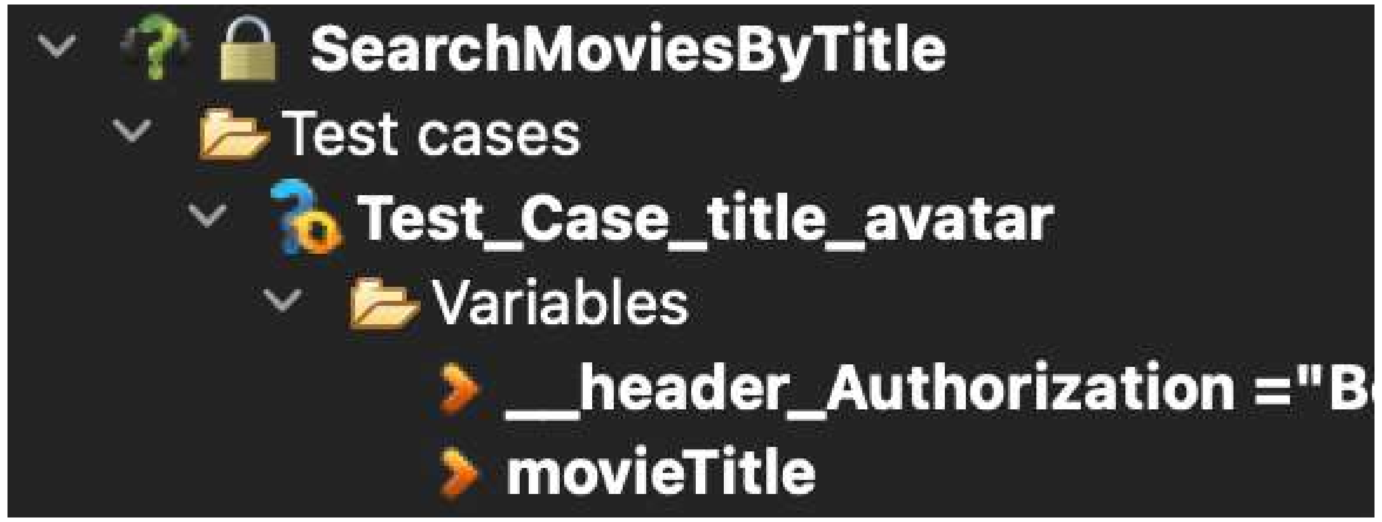
Please enter a name for object.

Name: Test_Case_title_avatar



Name: Test_Case_title_avatar

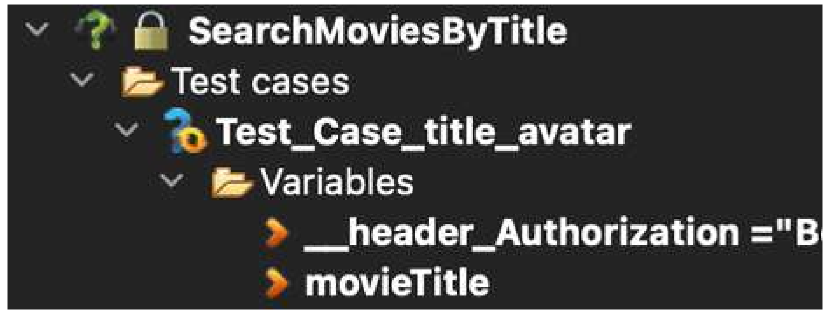
The test case is created in a **Test Cases folder**.



```
SearchMoviesByTitle
  Test cases
    Test_Case_title_avatar
      Variables
        __header_Authorization ="B
        movieTitle
```

3.8 Test the request

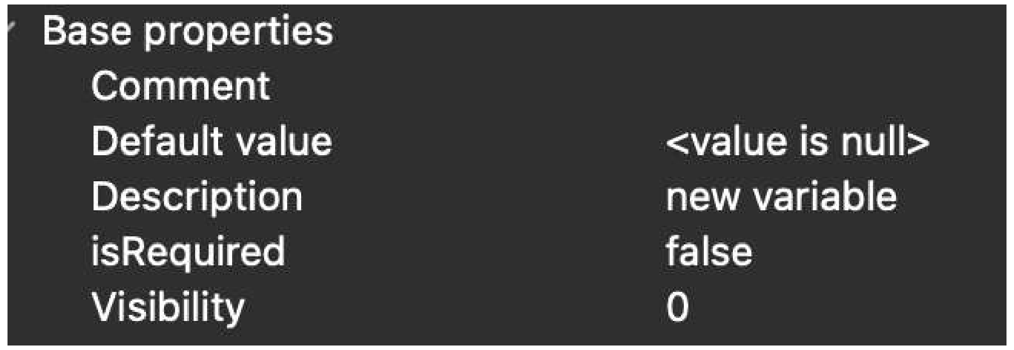
Select the **variable movieTitle** of the **test case.**



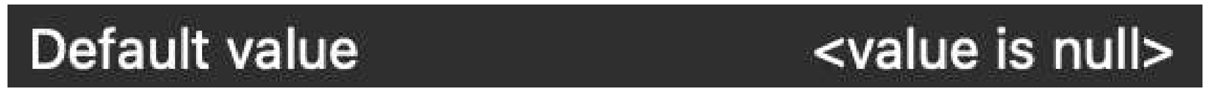
```
SearchMoviesByTitle
  Test cases
    Test_Case_title_avatar
      Variables
        __header_Authorization ="B
        movieTitle
```

In the Properties, edit the Default Value

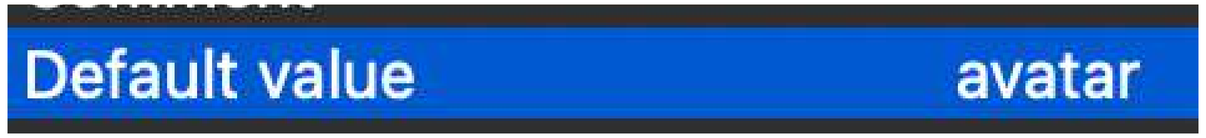
to enter a search term (in this case, 'avatar').



Base properties	
Comment	
Default value	<value is null>
Description	new variable
isRequired	false
Visibility	0



Default value <value is null>



Default value avatar

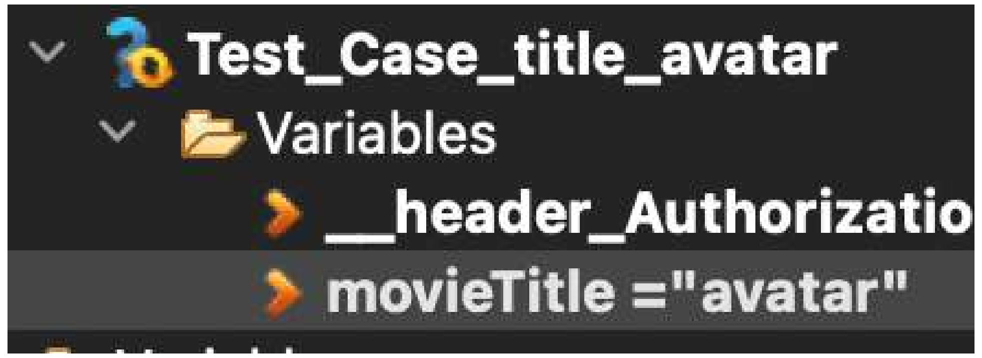
When we **edit** the **Default value**

of the **variable** in **properties**

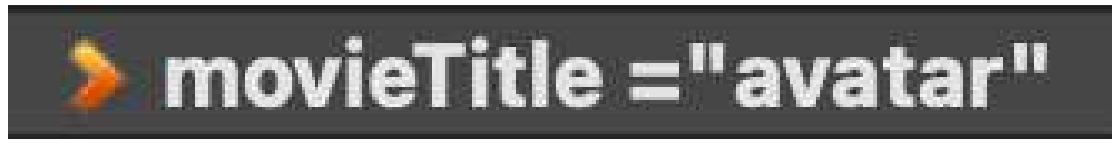
In the **treeview**,

the **value of the variable** 'movieTitle'

is **automatically modified**.



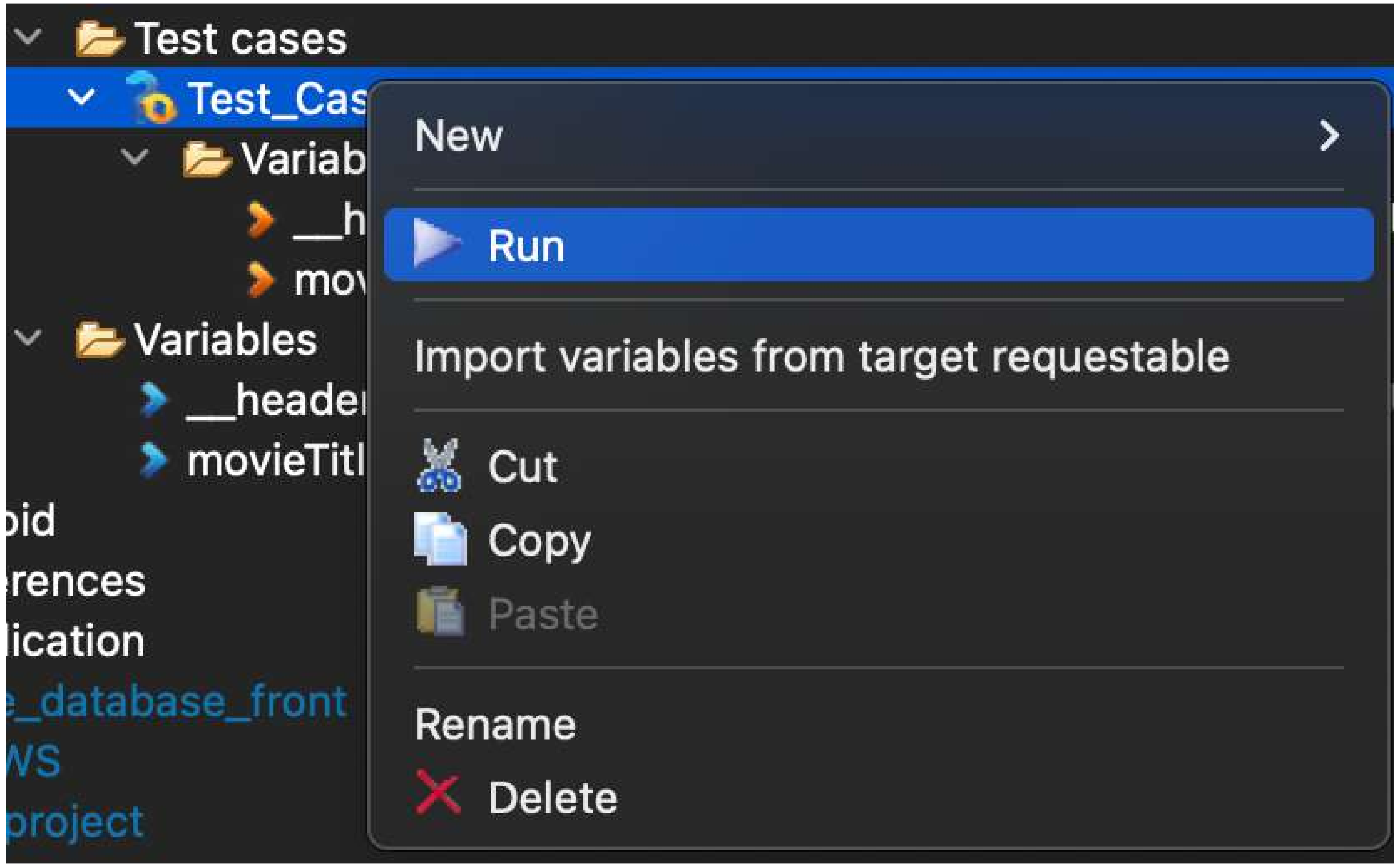
```
Test_Case_title_avatar
  Variables
    __header_Authorizatio
    movieTitle ="avatar"
```



movieTitle ="avatar"

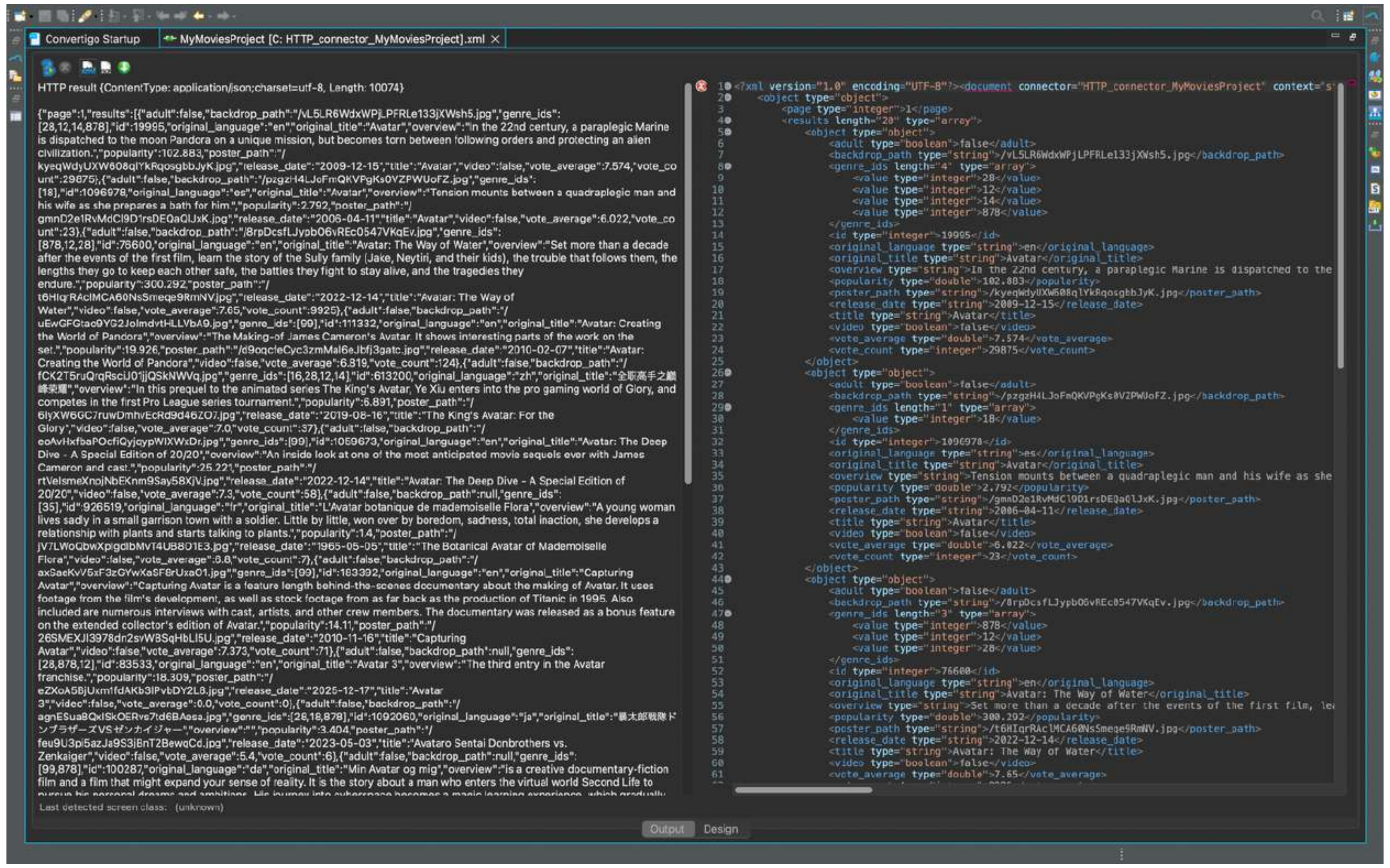
3.8 Test the request

To run the test,
right-click on the test case,
and click on **Run**.



3.8 Test the request

The **API response** is displayed in **XML** by default.

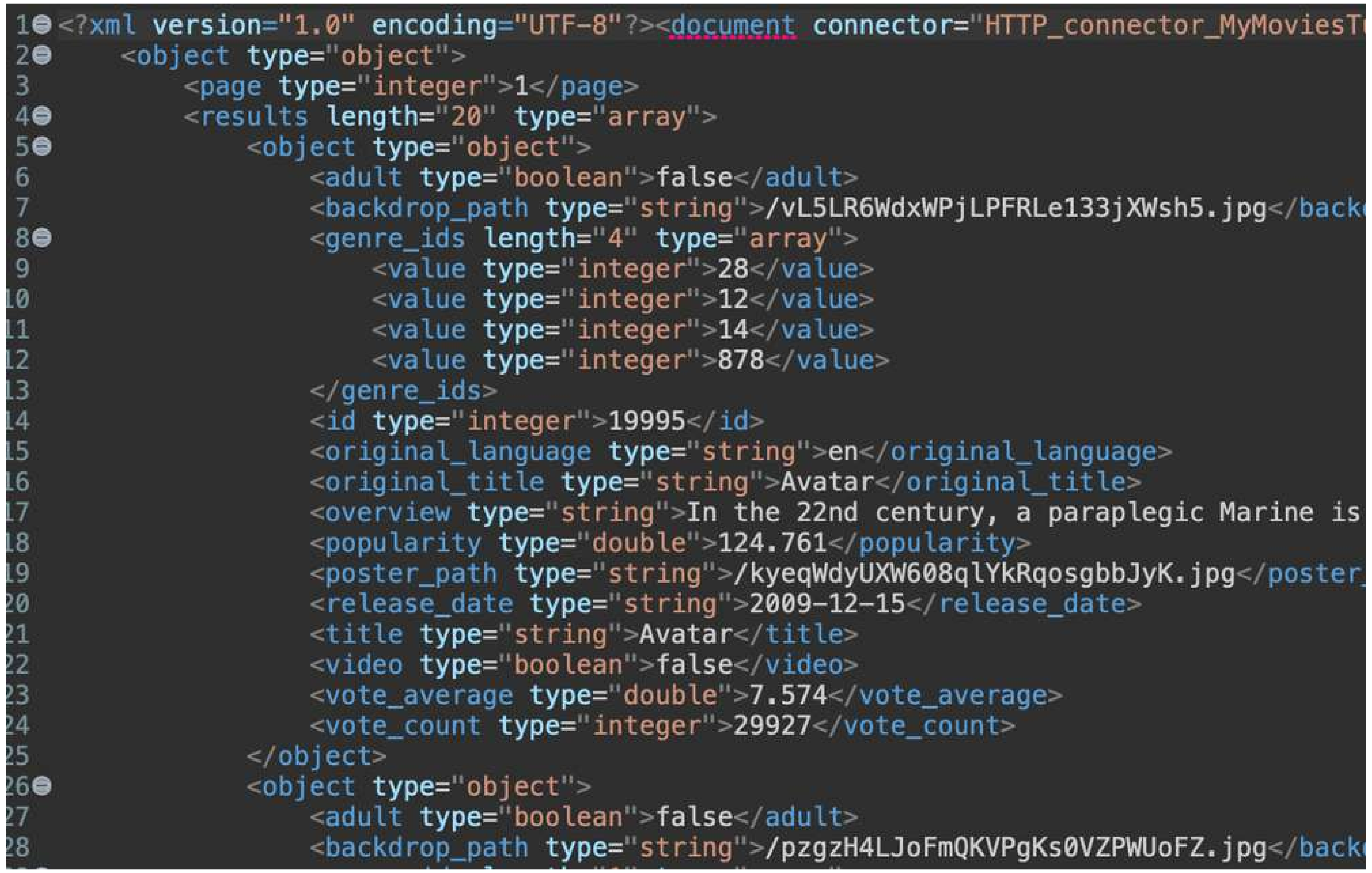


Convertigo Startup
MyMoviesProject [C: HTTP_connector_MyMoviesProject].xml
Last detected screen class: (unknown)
Output
Design

HTTP result {ContentType: application/json;charset=utf-8, Length: 10074}

```
{"page":1,"results":[{"adult":false,"backdrop_path":"/vL5LR6WdxWPjLPFRLe133jXWsh5.jpg","genre_ids":
[28,12,14,878],"id":19995,"original_language":"en","original_title":"Avatar","overview":"In the 22nd century, a paraplegic Marine
is dispatched to the moon Pandora on a unique mission, but becomes torn between following orders and protecting an alien
civilization.","popularity":102.883,"poster_path":"/
kyeqWdyUXW608qlYkRqosgbbJyK.jpg","release_date":"2009-12-15","title":"Avatar","video":false,"vote_average":7.574,"vote_co
unt":29875},{"adult":false,"backdrop_path":"/pzgzH4LJoFmQKVPgKs0VZPWUoFZ.jpg","genre_ids":
[18],"id":1096978,"original_language":"es","original_title":"Avatar","overview":"Tension mounts between a quadraplegic man and
his wife as she prepares a bath for him.","popularity":2.792,"poster_path":"/
gmnD2e1RvMdCl9D1rsDEQaQlJxK.jpg","release_date":"2006-04-11","title":"Avatar","video":false,"vote_average":6.022,"vote_co
unt":23},{"adult":false,"backdrop_path":"/8rpDcsfLJypbO6vREc0547VKqEv.jpg","genre_ids":
[878,12,28],"id":76600,"original_language":"en","original_title":"Avatar: The Way of Water","overview":"Set more than a decade
after the events of the first film, learn the story of the Sully family (Jake, Neytiri, and their kids), the trouble that follows them, the
lengths they go to keep each other safe, the battles they fight to stay alive, and the tragedies they
endure.","popularity":300.292,"poster_path":"/
t6HIqrRAclMCA60NsSmeqe9RmNV.jpg","release_date":"2022-12-14","title":"Avatar: The Way of
Water","video":false,"vote_average":7.65,"vote_count":9925},{"adult":false,"backdrop_path":"/
uEwGFGtao9YG2JolmdvtHLLVbA9.jpg","genre_ids":[99],"id":111332,"original_language":"en","original_title":"Avatar: Creating
the World of Pandora","overview":"The Making-of James Cameron's Avatar. It shows interesting parts of the work on the
set.","popularity":19.926,"poster_path":"/d9oqcfeCyc3zmMal6eJbfj3gatc.jpg","release_date":"2010-02-07","title":"Avatar:
Creating the World of Pandora","video":false,"vote_average":6.819,"vote_count":124},{"adult":false,"backdrop_path":"/
fCK2T5ruQrqRsciJ01jjQSkNWVq.jpg","genre_ids":[16,28,12,14],"id":613200,"original_language":"zh","original_title":"全職高手之巔
峰榮耀","overview":"In this prequel to the animated series The King's Avatar, Ye Xiu enters into the pro gaming world of Glory, and
competes in the first Pro League series tournament.","popularity":6.891,"poster_path":"/
6lyXW6GC7ruwDmhvEcRd9d46ZO7.jpg","release_date":"2019-08-16","title":"The King's Avatar: For the
Glory","video":false,"vote_average":7.0,"vote_count":37},{"adult":false,"backdrop_path":"/
eoAvHxfbaPOcfiQyjqypWIXWxDr.jpg","genre_ids":[99],"id":1059673,"original_language":"en","original_title":"Avatar: The Deep
Dive - A Special Edition of 20/20","overview":"An inside look at one of the most anticipated movie sequels ever with James
Cameron and cast.","popularity":25.221,"poster_path":"/
rtVelsmeXnojNbEKnm9Say58XjV.jpg","release_date":"2022-12-14","title":"Avatar: The Deep Dive - A Special Edition of
20/20","video":false,"vote_average":7.3,"vote_count":58},{"adult":false,"backdrop_path":null,"genre_ids":
[35],"id":926519,"original_language":"fr","original_title":"L'Avatar botanique de mademoiselle Flora","overview":"A young woman
lives sadly in a small garrison town with a soldier. Little by little, won over by boredom, sadness, total inaction, she develops a
relationship with plants and starts talking to plants.","popularity":1.4,"poster_path":"/
jV7LWoQbwXplgdlbMvT4UB8O1E3.jpg","release_date":"1965-05-05","title":"The Botanical Avatar of Mademoiselle
Flora","video":false,"vote_average":6.8,"vote_count":7},{"adult":false,"backdrop_path":"/
axSaeKvV5xF3zGYwXaSF8rUxaO1.jpg","genre_ids":[99],"id":183392,"original_language":"en","original_title":"Capturing
Avatar","overview":"Capturing Avatar is a feature length behind-the-scenes documentary about the making of Avatar. It uses
footage from the film's development, as well as stock footage from as far back as the production of Titanic in 1995. Also
included are numerous interviews with cast, artists, and other crew members. The documentary was released as a bonus feature
on the extended collector's edition of Avatar.","popularity":14.11,"poster_path":"/
26SMEXJI3978dn2svWBSqHbLI5U.jpg","release_date":"2010-11-16","title":"Capturing
Avatar","video":false,"vote_average":7.373,"vote_count":71},{"adult":false,"backdrop_path":null,"genre_ids":
[28,878,12],"id":83533,"original_language":"en","original_title":"Avatar 3","overview":"The third entry in the Avatar
franchise.","popularity":18.309,"poster_path":"/
eZXoA5BjUxm1fdAKb3IPvbDY2L8.jpg","release_date":"2025-12-17","title":"Avatar
3","video":false,"vote_average":0.0,"vote_count":0},{"adult":false,"backdrop_path":"/
agnESuaBQxISkOERvs7td6BAesa.jpg","genre_ids":[28,18,878],"id":1092060,"original_language":"ja","original_title":"暴太郎戦隊ド
ンブラザーズVSゼンカイジャー","overview":"","popularity":3.404,"poster_path":"/
feu9U3pi5azJa9S3jBnT2BewqCd.jpg","release_date":"2023-05-03","title":"Avataro Sentai Donbrothers vs.
Zenkaiger","video":false,"vote_average":5.4,"vote_count":6},{"adult":false,"backdrop_path":null,"genre_ids":
[99,878],"id":100287,"original_language":"da","original_title":"Min Avatar og mig","overview":"is a creative documentary-fiction
film and a film that might expand your sense of reality. It is the story about a man who enters the virtual world Second Life to
pursue his personal dreams and ambitions. His journey into cyberspace becomes a magic learning experience, which gradually
```

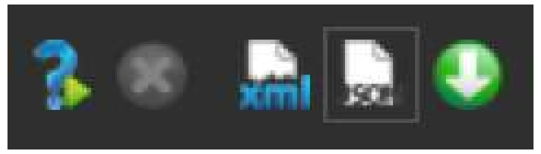
```
<?xml version="1.0" encoding="UTF-8"?><document connector="HTTP_connector_MyMoviesProject" context="s
    <object type="object">
        <page type="integer">1</page>
        <results length="20" type="array">
            <object type="object">
                <adult type="boolean">false</adult>
                <backdrop_path type="string">/vL5LR6WdxWPjLPFRLe133jXWsh5.jpg</backdrop_path>
                <genre_ids length="4" type="array">
                    <value type="integer">28</value>
                    <value type="integer">12</value>
                    <value type="integer">14</value>
                    <value type="integer">878</value>
                </genre_ids>
                <id type="integer">19995</id>
                <original_language type="string">en</original_language>
                <original_title type="string">Avatar</original_title>
                <overview type="string">In the 22nd century, a paraplegic Marine is dispatched to the
                <popularity type="double">102.883</popularity>
                <poster_path type="string">/kyeqWdyUXW608qlYkRqosgbbJyK.jpg</poster_path>
                <release_date type="string">2009-12-15</release_date>
                <title type="string">Avatar</title>
                <video type="boolean">false</video>
                <vote_average type="double">7.574</vote_average>
                <vote_count type="integer">29875</vote_count>
            </object>
            <object type="object">
                <adult type="boolean">false</adult>
                <backdrop_path type="string">/pzgzH4LJoFmQKVPgKs0VZPWUoFZ.jpg</backdrop_path>
                <genre_ids length="1" type="array">
                    <value type="integer">18</value>
                </genre_ids>
                <id type="integer">1096978</id>
                <original_language type="string">es</original_language>
                <original_title type="string">Avatar</original_title>
                <overview type="string">Tension mounts between a quadraplegic man and his wife as she
                <popularity type="double">2.792</popularity>
                <poster_path type="string">/gmnD2e1RvMdCl9D1rsDEQaQlJxK.jpg</poster_path>
                <release_date type="string">2006-04-11</release_date>
                <title type="string">Avatar</title>
                <video type="boolean">false</video>
                <vote_average type="double">6.022</vote_average>
                <vote_count type="integer">23</vote_count>
            </object>
            <object type="object">
                <adult type="boolean">false</adult>
                <backdrop_path type="string">/8rpDcsfLJypbO6vREc0547VKqEv.jpg</backdrop_path>
                <genre_ids length="3" type="array">
                    <value type="integer">878</value>
                    <value type="integer">12</value>
                    <value type="integer">28</value>
                </genre_ids>
                <id type="integer">76600</id>
                <original_language type="string">en</original_language>
                <original_title type="string">Avatar: The Way of Water</original_title>
                <overview type="string">Set more than a decade after the events of the first film, lea
                <popularity type="double">300.292</popularity>
                <poster_path type="string">/t6HIqrRAclMCA60NsSmeqe9RmNV.jpg</poster_path>
                <release_date type="string">2022-12-14</release_date>
                <title type="string">Avatar: The Way of Water</title>
                <video type="boolean">false</video>
                <vote_average type="double">7.65</vote_average>
```

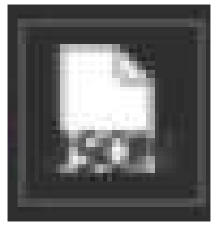


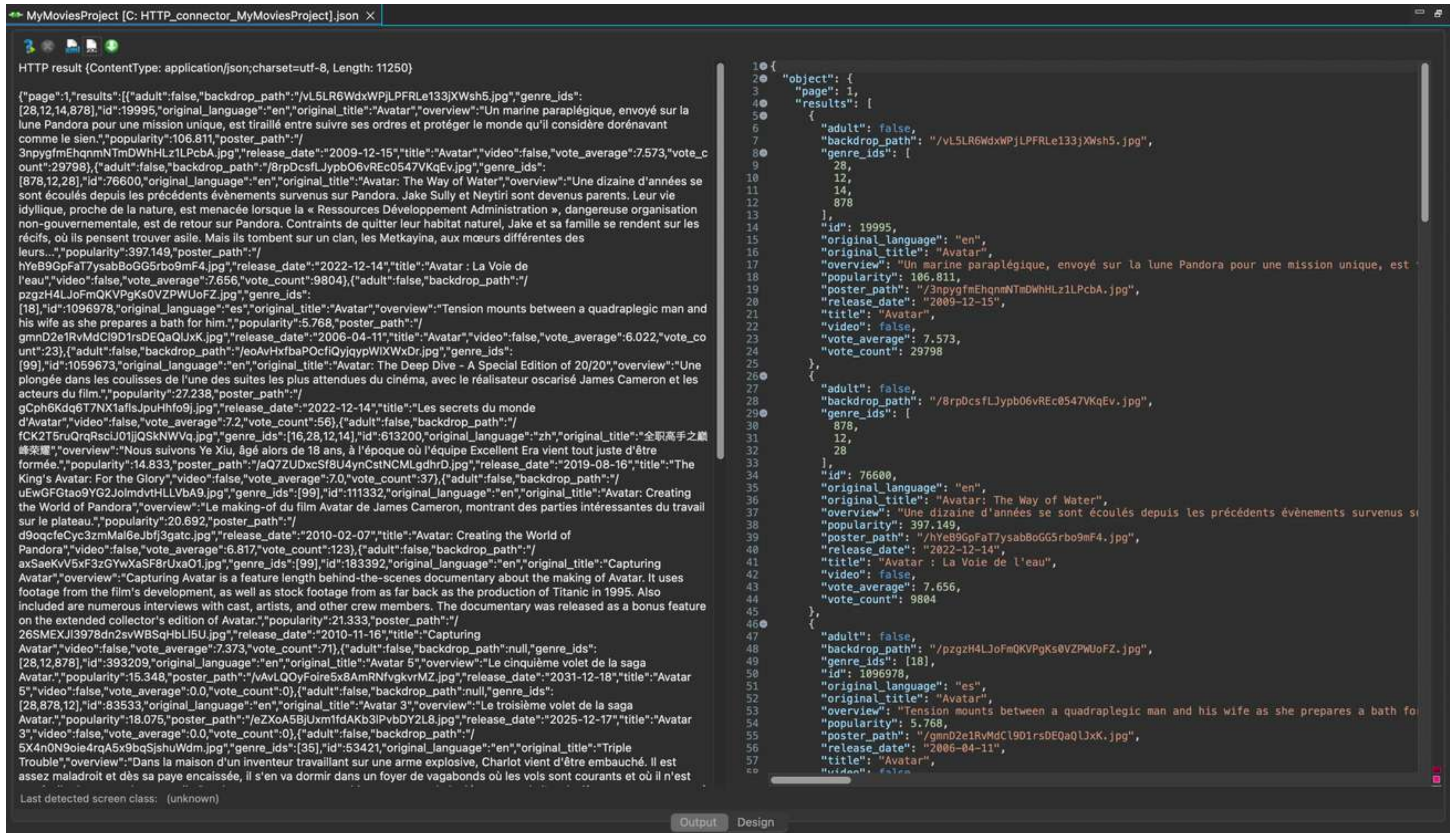
```
<?xml version="1.0" encoding="UTF-8"?><document connector="HTTP_connector_MyMoviesT
    <object type="object">
        <page type="integer">1</page>
        <results length="20" type="array">
            <object type="object">
                <adult type="boolean">false</adult>
                <backdrop_path type="string">/vL5LR6WdxWPjLPFRLe133jXWsh5.jpg</back
                <genre_ids length="4" type="array">
                    <value type="integer">28</value>
                    <value type="integer">12</value>
                    <value type="integer">14</value>
                    <value type="integer">878</value>
                </genre_ids>
                <id type="integer">19995</id>
                <original_language type="string">en</original_language>
                <original_title type="string">Avatar</original_title>
                <overview type="string">In the 22nd century, a paraplegic Marine is
                <popularity type="double">124.761</popularity>
                <poster_path type="string">/kyeqWdyUXW608qlYkRqosgbbJyK.jpg</poster_
                <release_date type="string">2009-12-15</release_date>
                <title type="string">Avatar</title>
                <video type="boolean">false</video>
                <vote_average type="double">7.574</vote_average>
                <vote_count type="integer">29927</vote_count>
            </object>
            <object type="object">
                <adult type="boolean">false</adult>
                <backdrop_path type="string">/pzgzH4LJoFmQKVPgKs0VZPWUoFZ.jpg</back
```

3.8 Test the request

Click on the **JSON button** to display the **API response** in **JSON**.





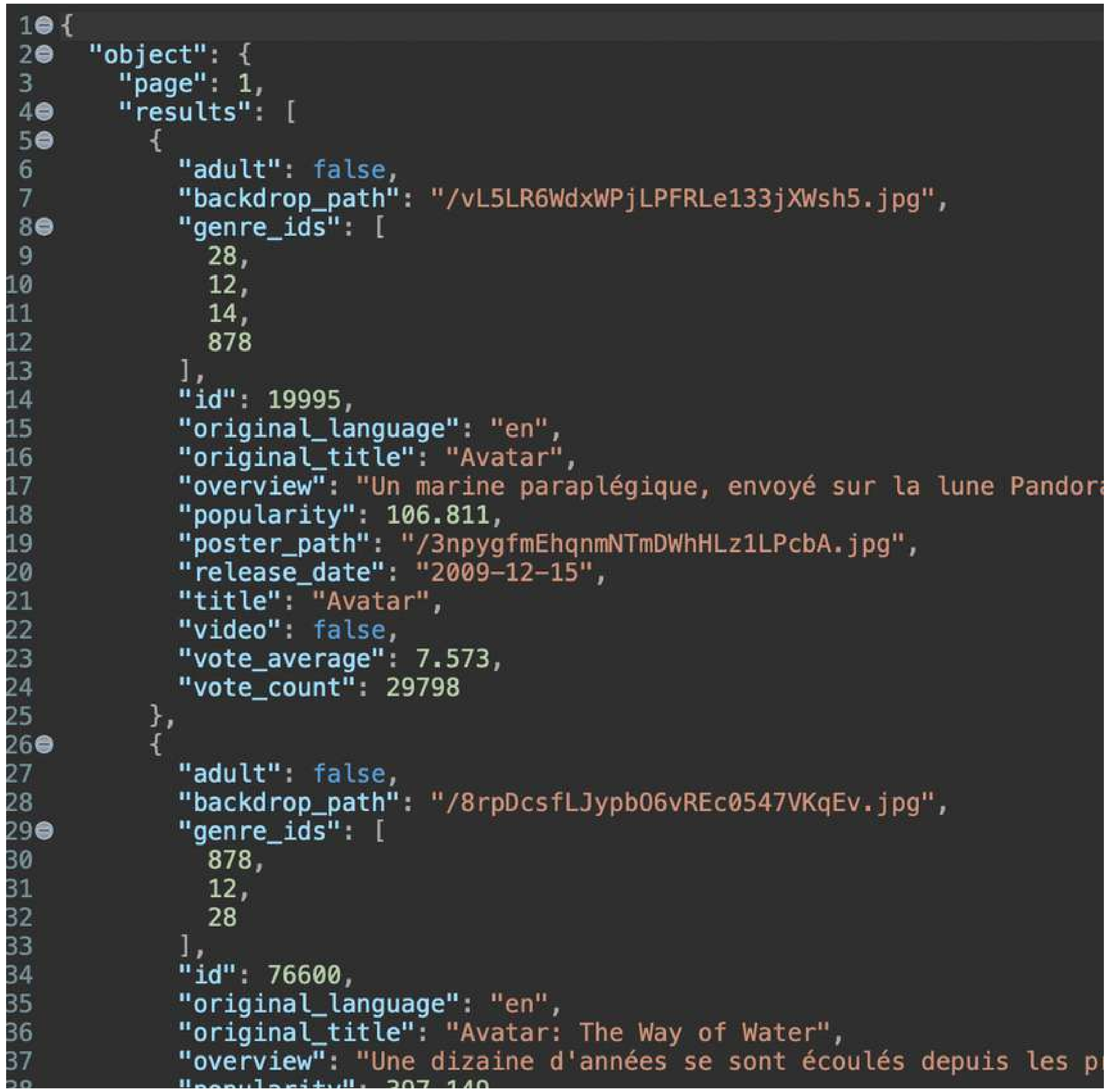


HTTP result {ContentType: application/json;charset=utf-8, Length: 11250}

```
{"page":1,"results":[{"adult":false,"backdrop_path":"/vL5LR6WdxWPjLPFRLe133jXWsh5.jpg","genre_ids":
[28,12,14,878],"id":19995,"original_language":"en","original_title":"Avatar","overview":"Un marine paraplégique, envoyé sur la
lune Pandora pour une mission unique, est tiraillé entre suivre ses ordres et protéger le monde qu'il considère dorénavant
comme le sien.","popularity":106.811,"poster_path":"/
3npygfmEhqnmNTmDWhHLz1LPcbA.jpg","release_date":"2009-12-15","title":"Avatar","video":false,"vote_average":7.573,"vote_c
ount":29798},{"adult":false,"backdrop_path":"/8rpDcsfLJypbO6vREc0547VKqEv.jpg","genre_ids":
[878,12,28],"id":76600,"original_language":"en","original_title":"Avatar: The Way of Water","overview":"Une dizaine d'années se
sont écoulés depuis les précédents évènements survenus sur Pandora. Jake Sully et Neytiri sont devenus parents. Leur vie
idyllique, proche de la nature, est menacée lorsque la « Ressources Développement Administration », dangereuse organisation
non-gouvernementale, est de retour sur Pandora. Contraints de quitter leur habitat naturel, Jake et sa famille se rendent sur les
récifs, où ils pensent trouver asile. Mais ils tombent sur un clan, les Metkayina, aux mœurs différentes des
leurs...","popularity":397.149,"poster_path":"/
hYeB9GpFaT7ysabBoGG5rbo9mF4.jpg","release_date":"2022-12-14","title":"Avatar : La Voie de
l'eau","video":false,"vote_average":7.656,"vote_count":9804},{"adult":false,"backdrop_path":"/
pzgzH4LJoFmQKVPgKs0VZPWUoFZ.jpg","genre_ids":
[18],"id":1096978,"original_language":"es","original_title":"Avatar","overview":"Tension mounts between a quadraplegic man and
his wife as she prepares a bath for him.","popularity":5.768,"poster_path":"/
gmnD2e1RvMdCl9D1rsDEQaQlJxK.jpg","release_date":"2006-04-11","title":"Avatar","video":false,"vote_average":6.022,"vote_co
unt":23},{"adult":false,"backdrop_path":"/eoAvHxfbaPOcfiQyjqypWIXWxDr.jpg","genre_ids":
[99],"id":1059673,"original_language":"en","original_title":"Avatar: The Deep Dive - A Special Edition of 20/20","overview":"Une
plongée dans les coulisses de l'une des suites les plus attendues du cinéma, avec le réalisateur oscarisé James Cameron et les
acteurs du film.","popularity":27.238,"poster_path":"/
gCph6Kdq6T7NX1aflsJpuHhfo9j.jpg","release_date":"2022-12-14","title":"Les secrets du monde
d'Avatar","video":false,"vote_average":7.2,"vote_count":56},{"adult":false,"backdrop_path":"/
fCK2T5ruQrqRsciJ01jjQSkNWVq.jpg","genre_ids":[16,28,12,14],"id":613200,"original_language":"zh","original_title":"全职高手之巅
峰荣耀","overview":"Nous suivons Ye Xiu, âgé alors de 18 ans, à l'époque où l'équipe Excellent Era vient tout juste d'être
formée.","popularity":14.833,"poster_path":"/aQ7ZUDxcSf8U4ynCstNCMLgdhrD.jpg","release_date":"2019-08-16","title":"The
King's Avatar: For the Glory","video":false,"vote_average":7.0,"vote_count":37},{"adult":false,"backdrop_path":"/
uEwGFGtao9YG2JolmdvtHLLVbA9.jpg","genre_ids":[99],"id":111332,"original_language":"en","original_title":"Avatar: Creating
the World of Pandora","overview":"Le making-of du film Avatar de James Cameron, montrant des parties intéressantes du travail
sur le plateau.","popularity":20.692,"poster_path":"/
d9oqcfeCyc3zmMal6eJbfj3gatc.jpg","release_date":"2010-02-07","title":"Avatar: Creating the World of
Pandora","video":false,"vote_average":6.817,"vote_count":123},{"adult":false,"backdrop_path":"/
axSaeKvV5xF3zGYwXaSF8rUxaO1.jpg","genre_ids":[99],"id":183392,"original_language":"en","original_title":"Capturing
Avatar","overview":"Capturing Avatar is a feature length behind-the-scenes documentary about the making of Avatar. It uses
footage from the film's development, as well as stock footage from as far back as the production of Titanic in 1995. Also
included are numerous interviews with cast, artists, and other crew members. The documentary was released as a bonus feature
on the extended collector's edition of Avatar.","popularity":21.333,"poster_path":"/
26SMEXJl3978dn2svWBSqHbLl5U.jpg","release_date":"2010-11-16","title":"Capturing
Avatar","video":false,"vote_average":7.373,"vote_count":71},{"adult":false,"backdrop_path":null,"genre_ids":
[28,12,878],"id":393209,"original_language":"en","original_title":"Avatar 5","overview":"Le cinquième volet de la saga
Avatar.","popularity":15.348,"poster_path":"/vAvLQOyFoire5x8AmRNfvgkvrMZ.jpg","release_date":"2031-12-18","title":"Avatar
5","video":false,"vote_average":0.0,"vote_count":0},{"adult":false,"backdrop_path":null,"genre_ids":
[28,878,12],"id":83533,"original_language":"en","original_title":"Avatar 3","overview":"Le troisième volet de la saga
Avatar.","popularity":18.075,"poster_path":"/eZXoA5BjUxm1fdAKb3lPvbDY2L8.jpg","release_date":"2025-12-17","title":"Avatar
3","video":false,"vote_average":0.0,"vote_count":0},{"adult":false,"backdrop_path":"/
5X4n0N9oie4rqA5x9bqSjshuWdm.jpg","genre_ids":[35],"id":53421,"original_language":"en","original_title":"Triple
Trouble","overview":"Dans la maison d'un inventeur travaillant sur une arme explosive, Charlot vient d'être embauché. Il est
assez maladroit et dès sa paye encaissée, il s'en va dormir dans un foyer de vagabonds où les vols sont courants et où il n'est
```

Last detected screen class: (unknown)

```
{
  "object": {
    "page": 1,
    "results": [
      {
        "adult": false,
        "backdrop_path": "/vL5LR6WdxWPjLPFRLe133jXWsh5.jpg",
        "genre_ids": [
          28,
          12,
          14,
          878
        ],
        "id": 19995,
        "original_language": "en",
        "original_title": "Avatar",
        "overview": "Un marine paraplégique, envoyé sur la lune Pandora pour une mission unique, est 
        "popularity": 106.811,
        "poster_path": "/3npygfmEhqnmNTmDWhHLz1LPcbA.jpg",
        "release_date": "2009-12-15",
        "title": "Avatar",
        "video": false,
        "vote_average": 7.573,
        "vote_count": 29798
      },
      {
        "adult": false,
        "backdrop_path": "/8rpDcsfLJypb06vREc0547VKqEv.jpg",
        "genre_ids": [
          878,
          12,
          28
        ],
        "id": 76600,
        "original_language": "en",
        "original_title": "Avatar: The Way of Water",
        "overview": "Une dizaine d'années se sont écoulés depuis les précédents évènements survenus s
        "popularity": 397.149,
        "poster_path": "/hYeB9GpFaT7ysabBoGG5rbo9mF4.jpg",
        "release_date": "2022-12-14",
        "title": "Avatar : La Voie de l'eau",
        "video": false,
        "vote_average": 7.656,
        "vote_count": 9804
      },
      {
        "adult": false,
        "backdrop_path": "/pzgzH4LJoFmQKVPgKs0VZPWUoFZ.jpg",
        "genre_ids": [18],
        "id": 1096978,
        "original_language": "es",
        "original_title": "Avatar",
        "overview": "Tension mounts between a quadraplegic man and his wife as she prepares a bath fo
        "popularity": 5.768,
        "poster_path": "/gmnD2e1RvMdCl9D1rsDEQaQlJxK.jpg",
        "release_date": "2006-04-11",
        "title": "Avatar",
```

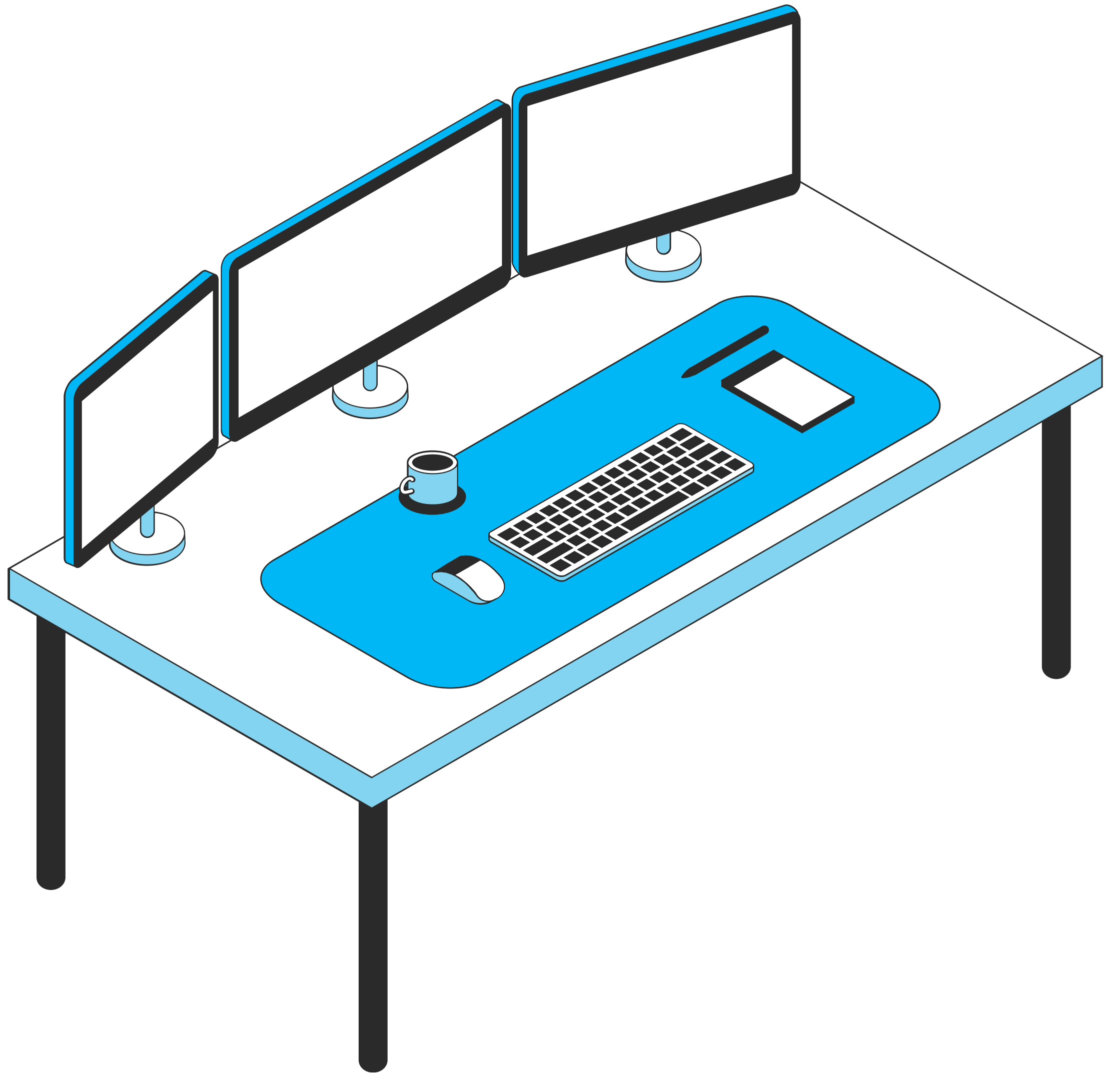


```
{
  "object": {
    "page": 1,
    "results": [
      {
        "adult": false,
        "backdrop_path": "/vL5LR6WdxWPjLPFRLe133jXWsh5.jpg",
        "genre_ids": [
          28,
          12,
          14,
          878
        ],
        "id": 19995,
        "original_language": "en",
        "original_title": "Avatar",
        "overview": "Un marine paraplégique, envoyé sur la lune Pandora
        "popularity": 106.811,
        "poster_path": "/3npygfmEhqnmNTmDWhHLz1LPcbA.jpg",
        "release_date": "2009-12-15",
        "title": "Avatar",
        "video": false,
        "vote_average": 7.573,
        "vote_count": 29798
      },
      {
        "adult": false,
        "backdrop_path": "/8rpDcsfLJypb06vREc0547VKqEv.jpg",
        "genre_ids": [
          878,
          12,
          28
        ],
        "id": 76600,
        "original_language": "en",
        "original_title": "Avatar: The Way of Water",
        "overview": "Une dizaine d'années se sont écoulés depuis les p
```

convertigo

4 – Sequences

How to create a flow of actions.



4.1 Sequences

4.2 Steps

4.3 XML & XPath

4.4 Source Picker

4.5 Create a sequence

4.6 Call a transaction from a sequence

4.7 Create a custom data structure

4.8 Test the sequence

4.1 Sequences

The **Sequence** is a very important **backend object**.

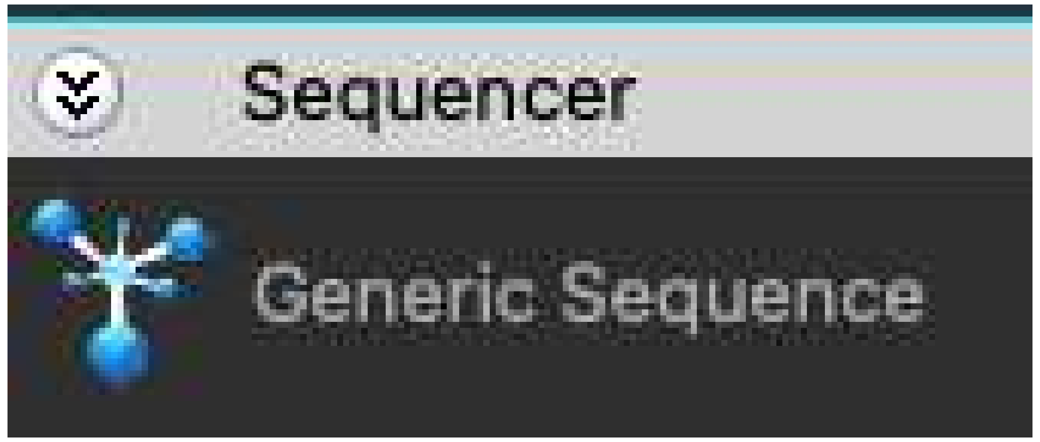
It is labelled as **Generic Sequence** in the palette.

In Convertigo Low Code Studio, **Sequences** are used to design the **logical flow** and **behavior** of the **backend** of your application by specifying **what actions should occur** and **in what order**.

Sequences allows you to

- create **sequences of actions**
- define **conditions and decision points**
- manage **the order** in which these **actions are executed**
- **define and manage the flow of actions with a series of successive steps**

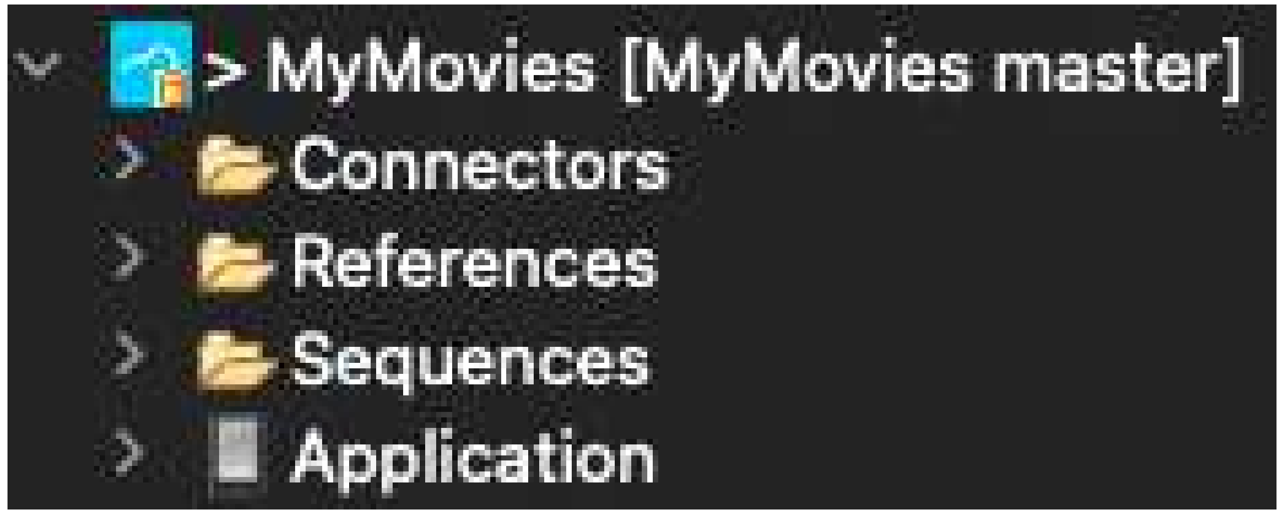
Object Sequence in the palette



Sequencer

Generic Sequence

Sequences folder in a project



> MyMovies [MyMovies master]

Connectors

References

Sequences

Application

4.2 Steps

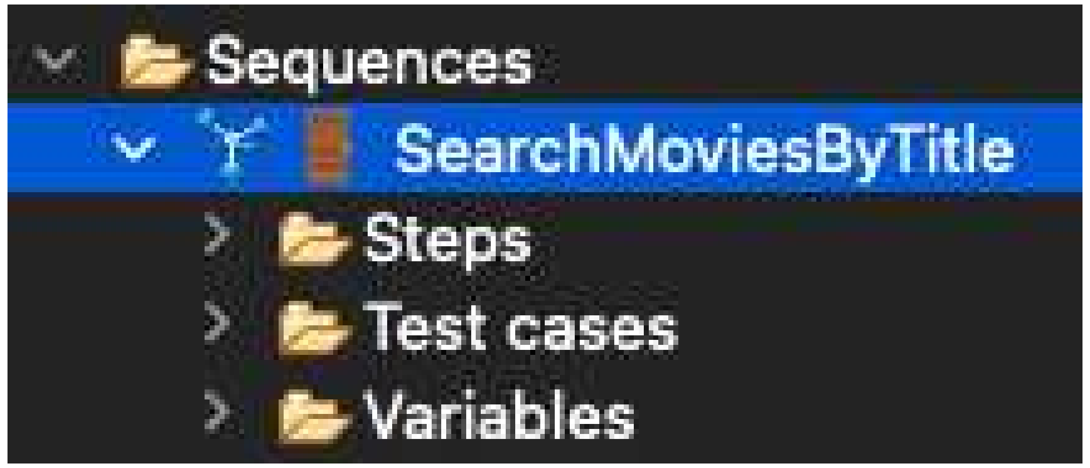
Steps are **back-end objects**.

A step is a **fundamental building block** used to **define a specific task, action, or operation** within a sequence.
For example, making an API request, showing a message, performing data manipulation…

Steps are **organized** to create **a sequence of actions** that the application should perform **in response** to **certain events** or **user interactions**.

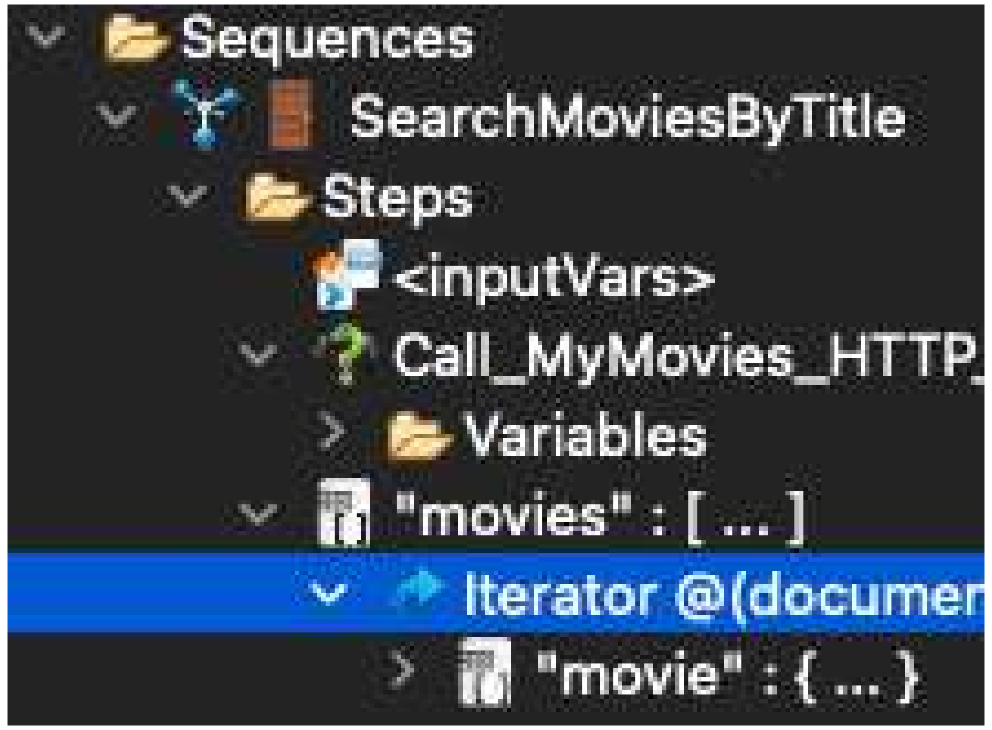
It allows developers to **define the logic and behavior** of the application in a **structured and modular manner**.

Example of a steps folder in a sequence



Sequences
SearchMoviesByTitle
Steps
Test cases
Variables

Example of a series of steps in a sequence

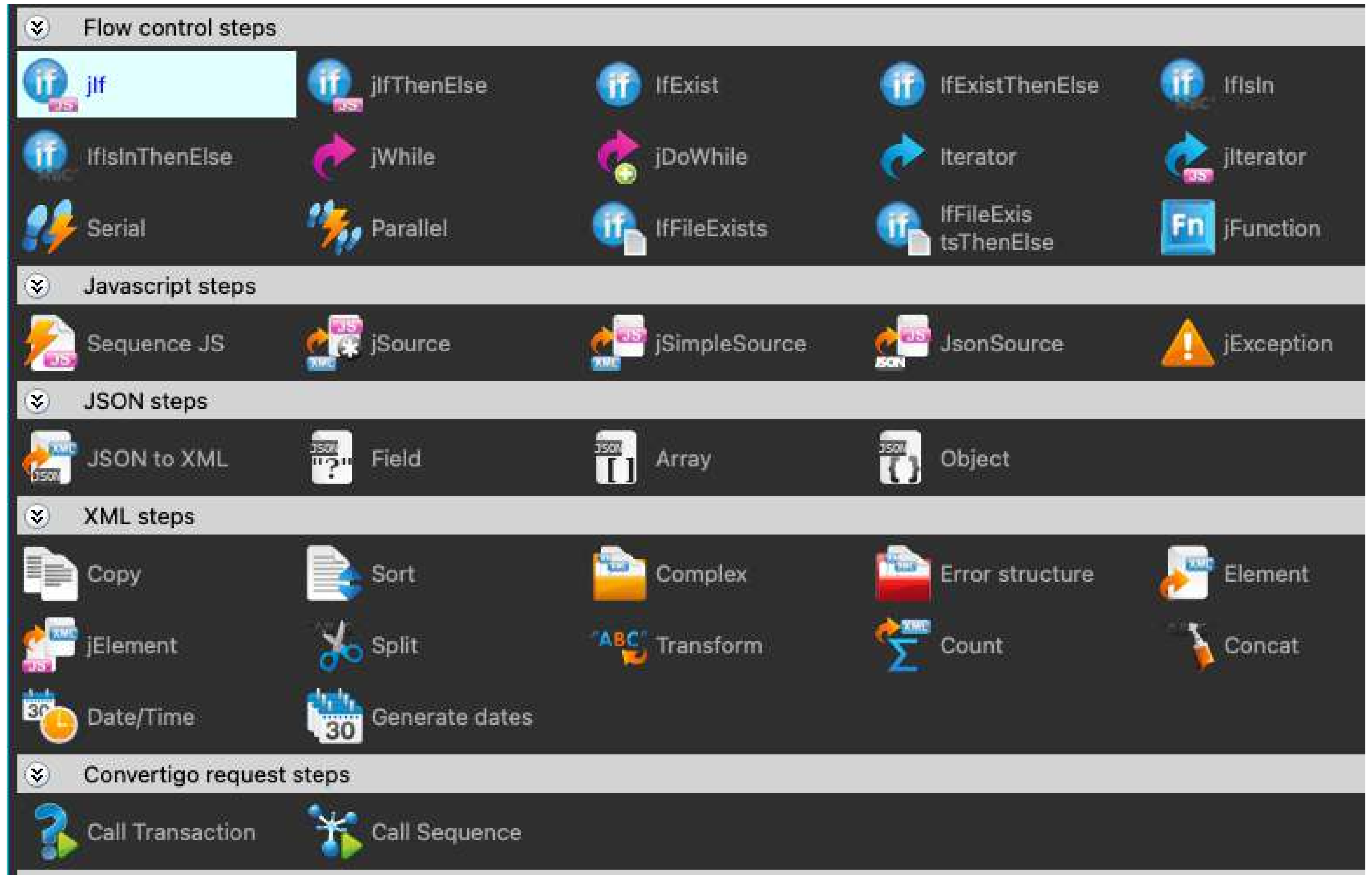


Sequences
SearchMoviesByTitle
Steps
<inputVars>
Call_MyMovies_HTTP_
Variables
"movies" : [...]
Iterator @(documen
"movie" : { ... }

4.2 Steps

Categories of Steps

Examples of Steps in the palette



There are different categories of steps :

- **Convertigo request steps** => to call a sequence or transaction
- **Flow Control Steps** => to control the sequence of actions and logic within a sequence
- **Javascript steps** => to incorporate custom JavaScript code in sequences
- **XML steps** => to work with XML data in sequences
- **JSON Steps** => to work with JSON data in sequences
- **HTTP session management** => to manage user sessions in web applications
- **File management steps** => to handle and manipulate files on the local system or server
- Others

4.2 Steps

JSON Steps

Convertigo provides **JSON steps** to **manipulate and interact with JSON data** in sequences.

Array – JSON step

When added to a sequence, this step creates an **XML element** (element node) ready to output a JSON Array

- First, you **drag-and-drop the step into a sequence**
- then, you **drag-and-drop the data** you want to manipulate from the Source Picker into the step in the sequence.

Object – JSON step

When added to a sequence, this step creates a JSON Object.

Field – JSON Step

When added to a sequence, this step creates a JSON string, number, boolean or null.

4.3 XML & XPath

XML Data Structure

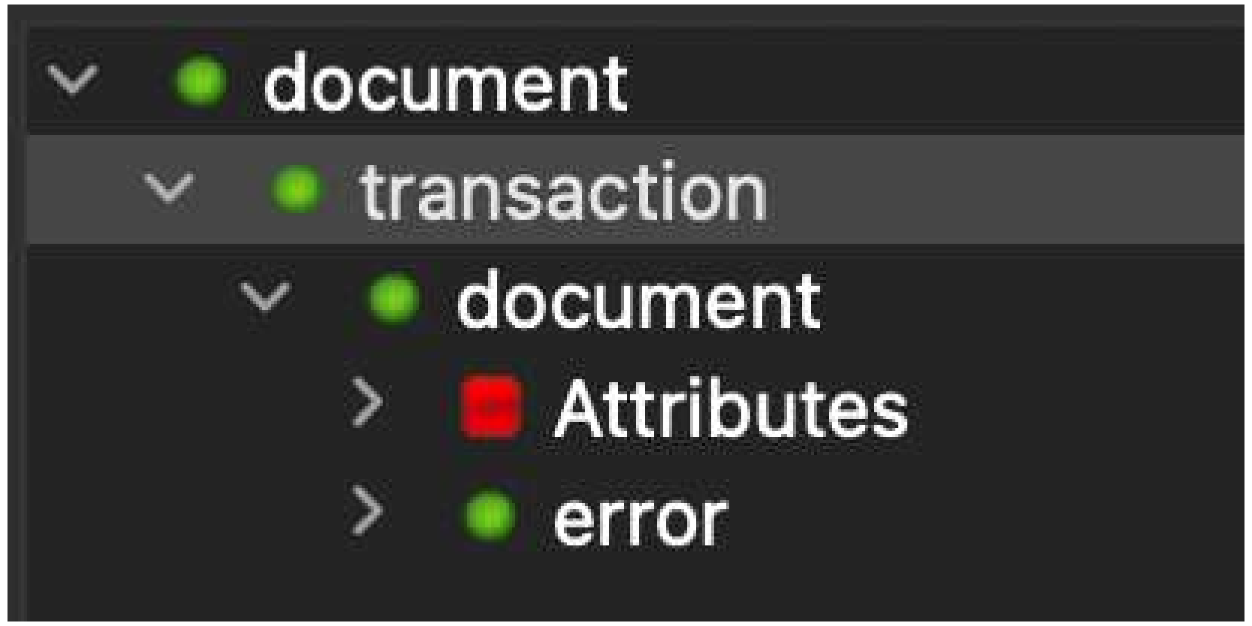
In Convertigo,
the data structure is **based on XML regardless of its source**.

The XML data structure follows the standard XML format.

It is organized hierarchically in a **tree structure**
with one **root element**, the **document**,
that is the **parent of all other elements**.

Each element has **attributes** and **text content**

Example of XML Data structure in Convertigo



```
document
  transaction
    document
      Attributes
      error
```

4.3 XML & XPath

XPath

XPath is a **language**

used for **navigating and querying XML documents**

XPath provides a way

to pinpoint **specific elements** and **data**

within an **XML structure**

by using **path expressions** that **define the location of nodes**.

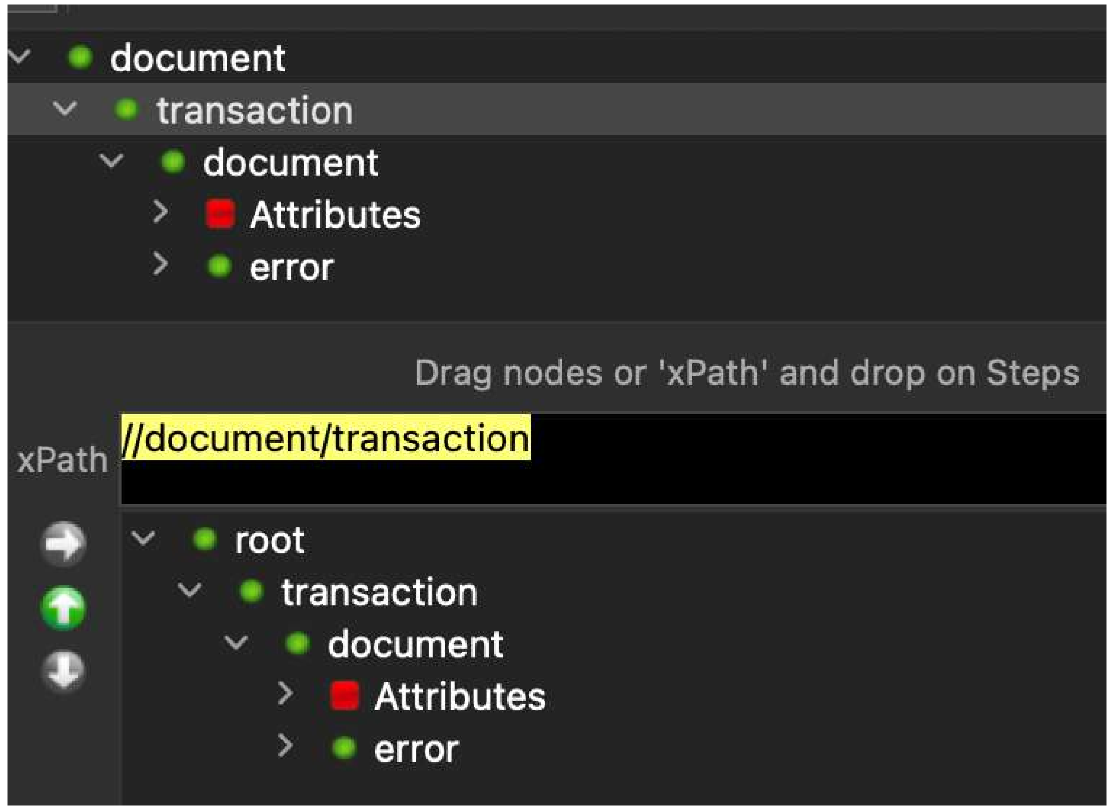
XPath expressions are used

to **identify and traverse these nodes**

within an **XML document**,

allowing for **data extraction and manipulation**.

Example of XML Data structure & XPath in Convertigo

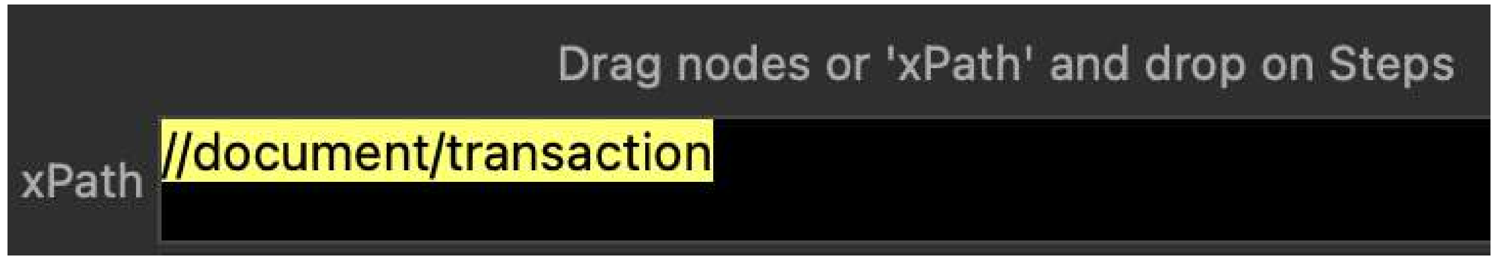


```
document
  transaction
    document
      Attributes
      error
```

Drag nodes or 'xPath' and drop on Steps

xPath //document/transaction

```
root
  transaction
    document
      Attributes
      error
```



Drag nodes or 'xPath' and drop on Steps

xPath //document/transaction

4.3 XML & XPath

Nodes

In XML and XPath,

Nodes are the **individual components of an XML document**.

There are several types of nodes :

- **element nodes** representing XML elements

  –> marked by a green dot in the XML Data structure in Convertigo

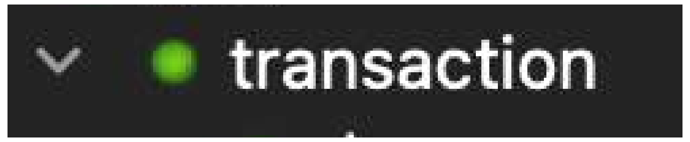
- **attribute nodes** representing attributes of elements

  –> marked by a red square in the XML Data structure

- **text nodes** containing textual content within elements

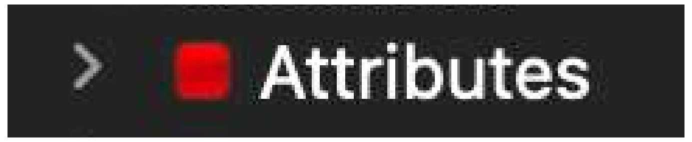
  –> marked by **TxT** in the XML Data structure

Element node



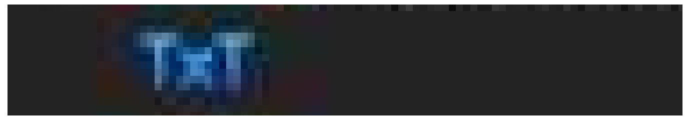
● transaction

Attribute node



🟥 Attributes

Text node



TxT

4.4 Source Picker

Sources

Each transaction, sequence, and step

- is a **data source** for the next step
- has a property called “output”
- emits data in the source picker

A **source** is defined as a **reference on a step**
previously existing in the parent sequence,
associated with an XPath applied on the step’s result DOM.

At runtime, the **XPath** is applied on the step’s **current execution result XML** and **extracts a list of XML nodes** resulting from this execution.

4.4 Source Picker

Each transaction, sequence, and step

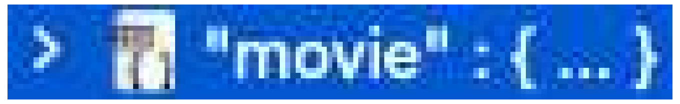
- is a **data source** for the next step
- **emits data** in the **source picker**

The source picker

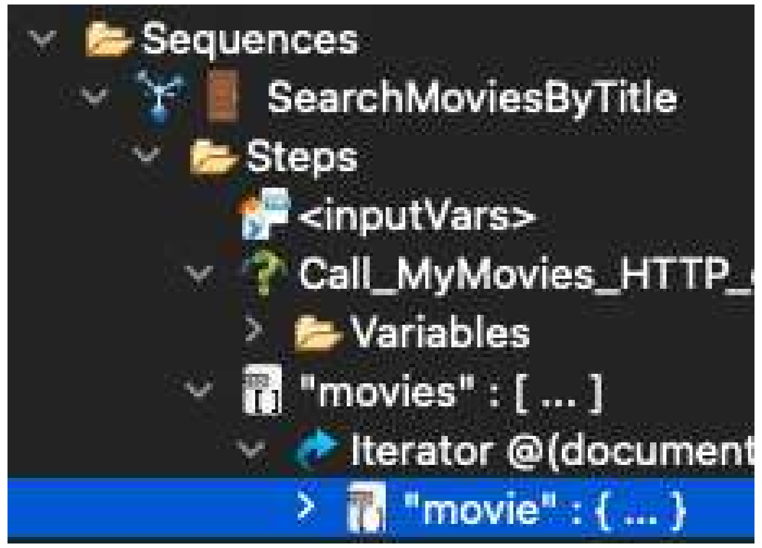
- displays the **structure of the data** emitted by a step.
- allows you to **select the XPath** without typing it by **dragging and dropping the node directly into a step**.

The **XPath** is used as **data path for accessing data**.

Example : step “movie” in sequence

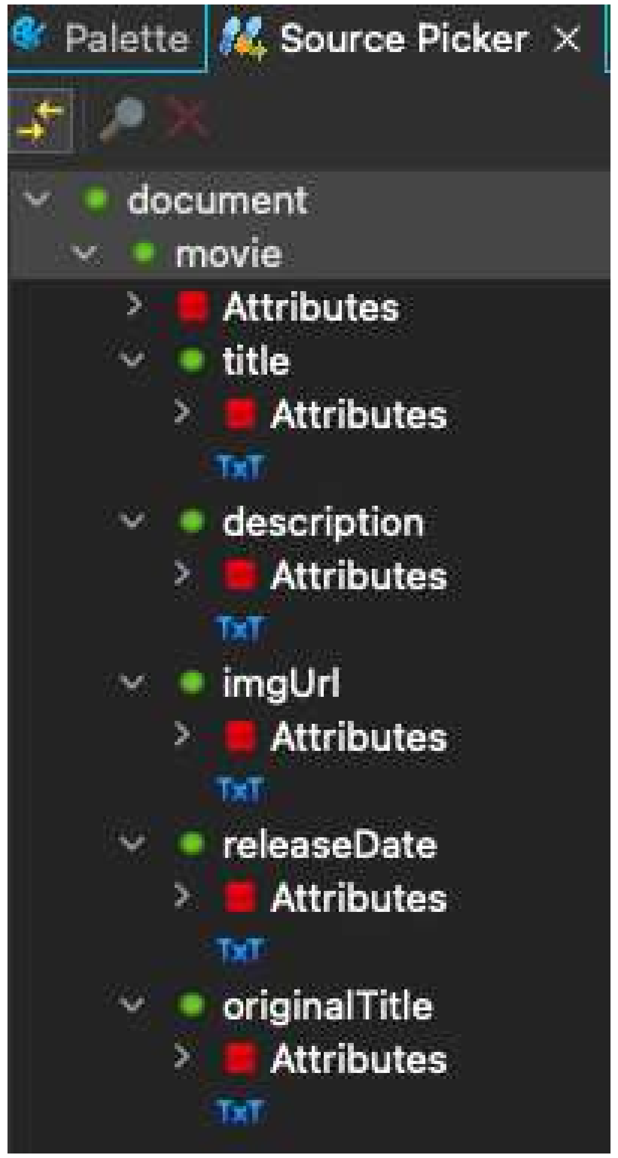


"movie" : { ... }



```
Sequences
  SearchMoviesByTitle
    Steps
      <inputVars>
      Call_MyMovies_HTTP_
        Variables
      "movies" : [ ... ]
        Iterator @(document
          "movie" : { ... }
```

Data structure of step “movie” in source picker



Palette | Source Picker ×

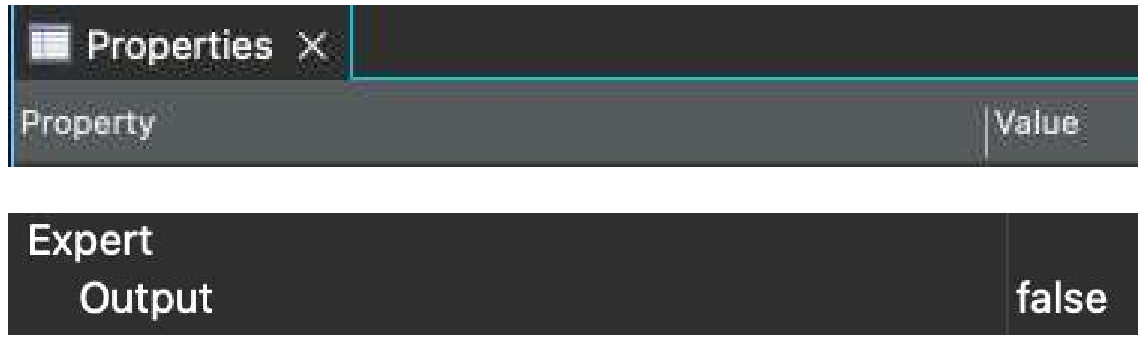
```
document
  movie
    Attributes
    title
      Attributes
      TxT
    description
      Attributes
      TxT
    imgUrl
      Attributes
      TxT
    releaseDate
      Attributes
      TxT
    originalTitle
      Attributes
      TxT
```

4.4 Source Picker

Output Property

Each transaction, sequence, and step

- has a **property** called “**output**”



Properties

Property	Value
Expert	
Output	false

The **Output property** defines whether the **XML generated by this step** should be **appended to the resulting XML**.

- Set this property to **true** to add the step’s resulting XML to the **sequence’s output XML** (default value for steps generating XML).

- Set this property to **false** to **prevent the steps’s XML result to appear** in the sequence’s output XML. Setting this property to false **does not prevent** the step’s generated XML from being **used as a source by other steps**.

4.4 Source Picker

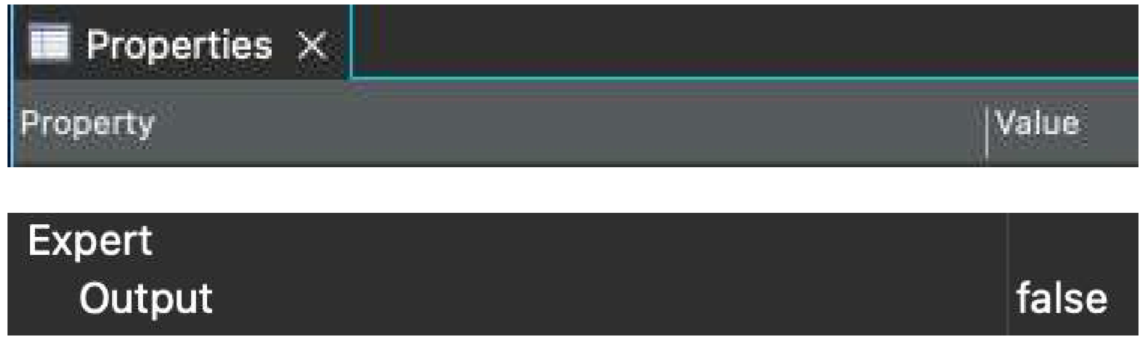
Output Property

To handle the **data emitted by a step**,

there are 2 options :

First option : If you need the **whole data emitted by a step**

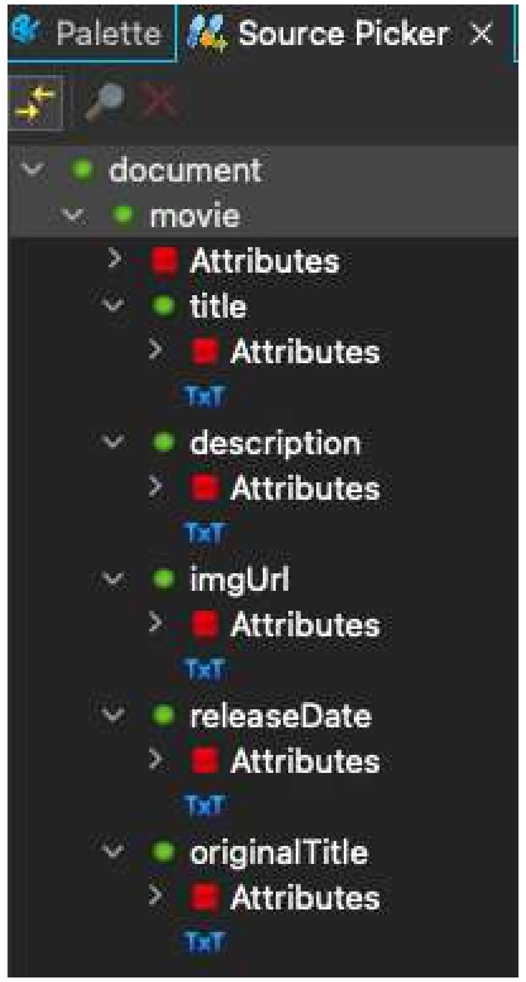
1. Put '**output**' on '**true**'
2. The step emits data in the response



Property	Value
Expert	
Output	false

Second option : If you need to **filter the data** and keep only specific data

1. Put '**output**' on '**false**'
2. The step doesn't emit in the response but still **emits in the source picker**.
3. You **select the data** you need **in the source picker** by **drag-and-dropping it in a sourceable step**
4. The following step can connect to this source through it.



```
document
  movie
    Attributes
    title
      Attributes
      TxT
    description
      Attributes
      TxT
    imgUrl
      Attributes
      TxT
    releaseDate
      Attributes
      TxT
    originalTitle
      Attributes
      TxT
```

4.4 Source Picker

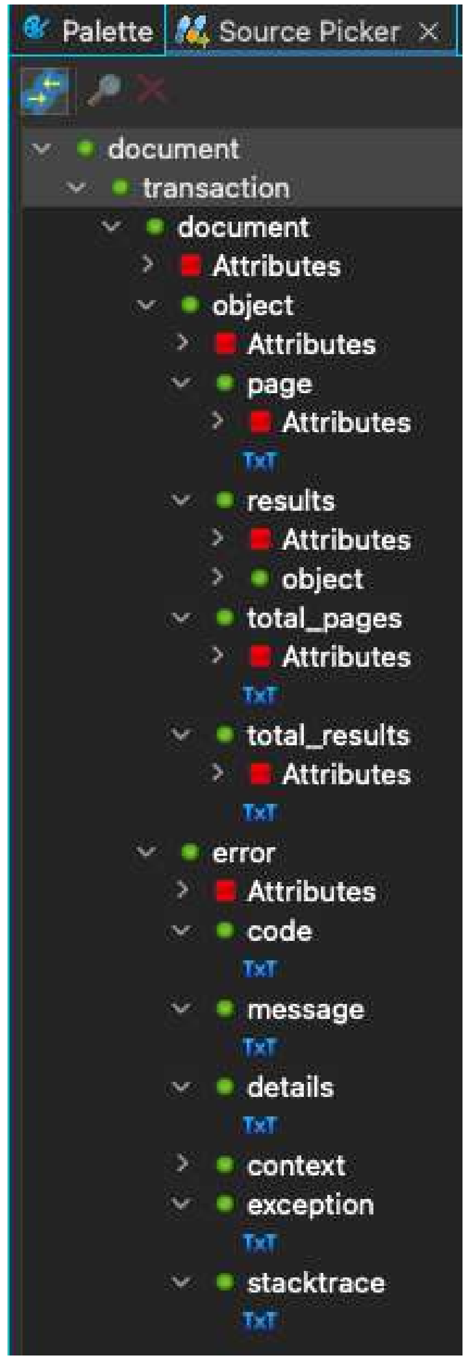
Transaction data structure

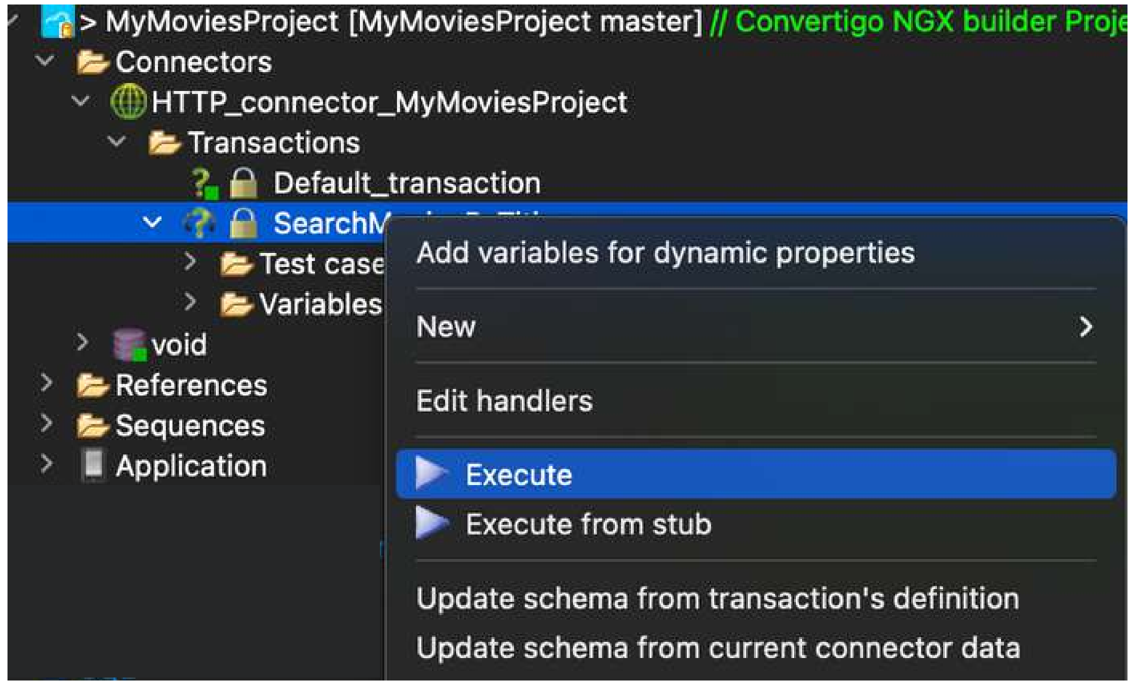
Calling a **transaction** brings back **data** with a **structure described in the source picker**.

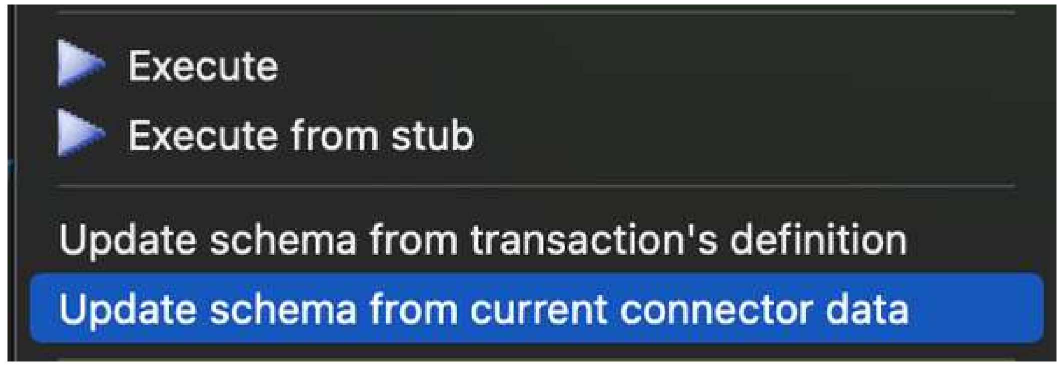
In other steps, the **output structure** is always the same.
In a **transaction call**, the **output structure** is unknown.

→ To discover it, you need to **execute the transaction once**.
→ Then retrieve the structure
with **Import data structure from current connector data**..

As good practice, this should be done when the transaction is created, before creating the sequence.




> MyMoviesProject [MyMoviesProject master] // Convertigo NGX builder Proje
Connectors
HTTP_connector_MyMoviesProject
Transactions
Default_transaction
SearchM
Test case
Variables
void
References
Sequences
Application
Add variables for dynamic properties
New
Edit handlers
Execute
Execute from stub
Update schema from transaction's definition
Update schema from current connector data

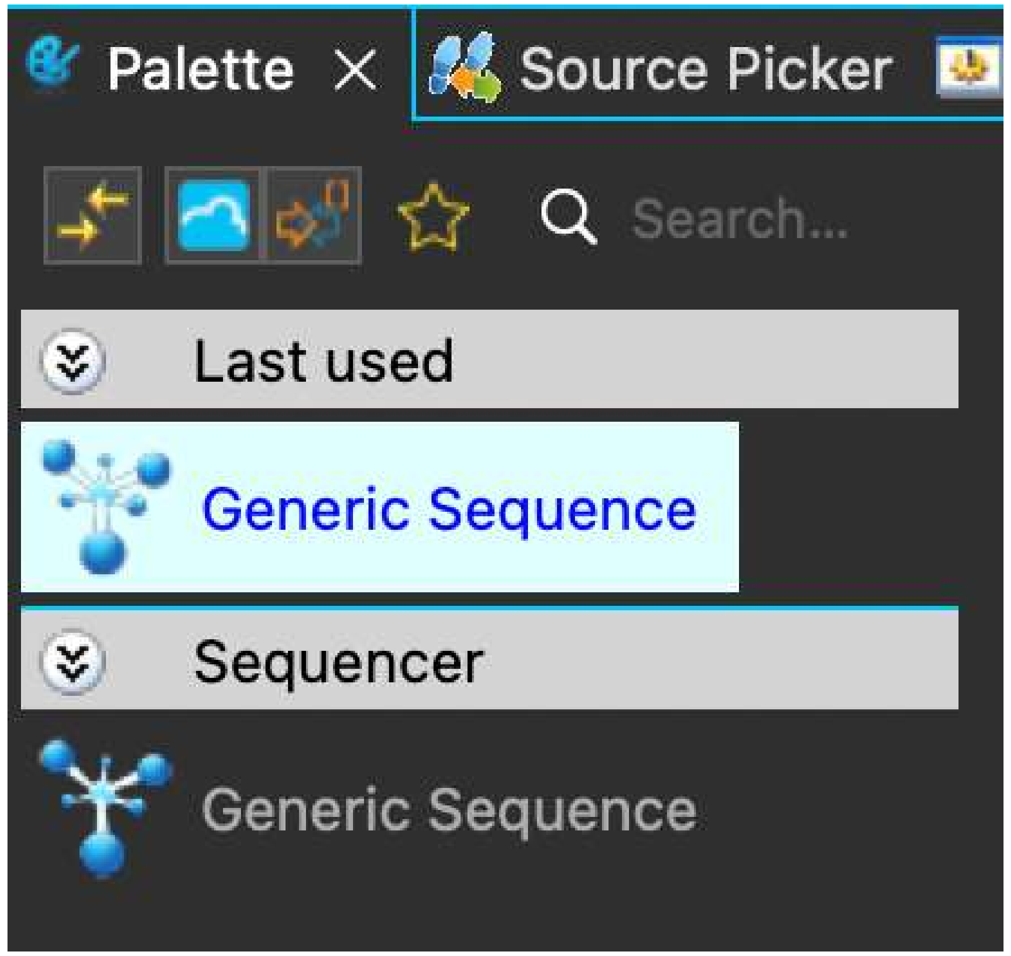

Execute
Execute from stub
Update schema from transaction's definition
Update schema from current connector data

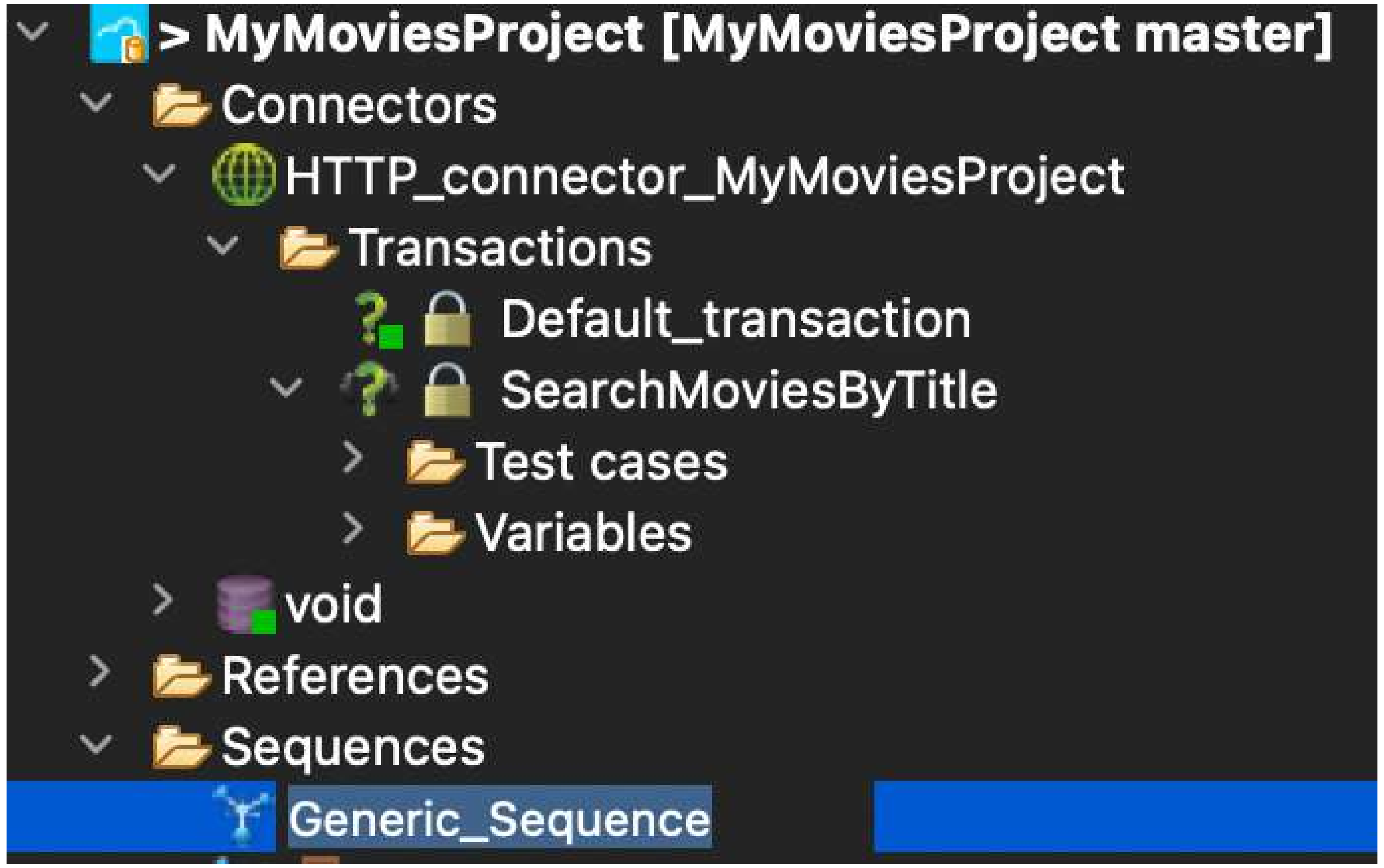
4.5 Create a sequence

To create a sequence, you have several options.

First option:

You can **drag and drop a Generic Sequence** from the palette in the tree structure and rename it.





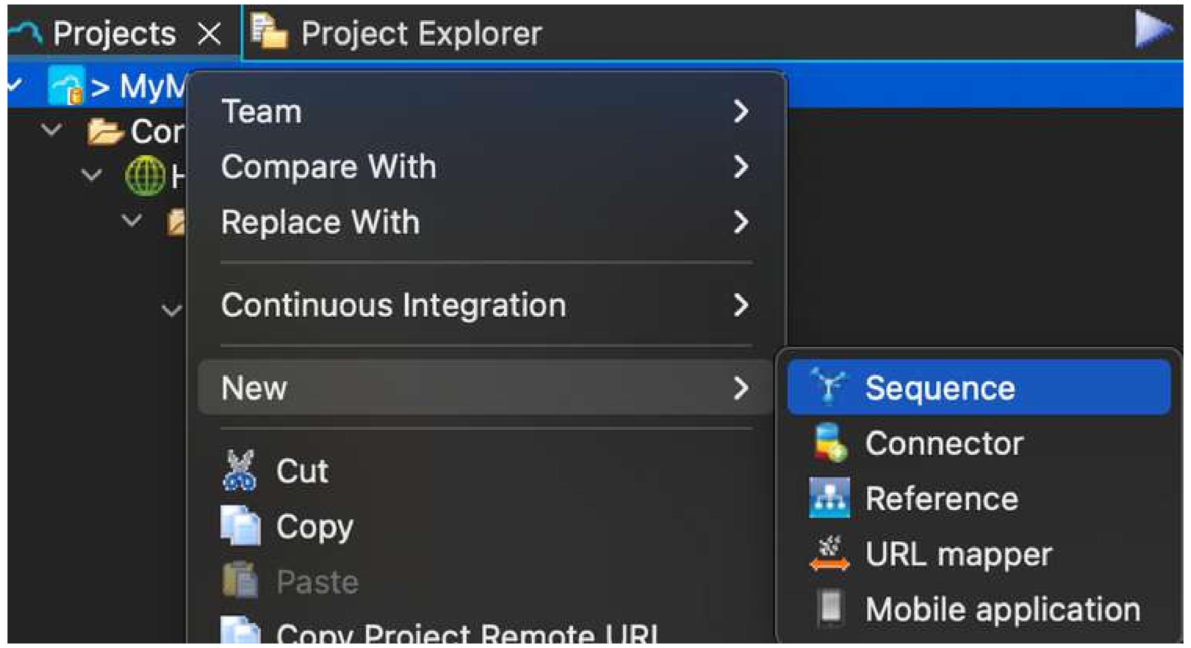
4.5 Create a sequence

Second option:

To create a sequence,

you can **right-click on the project**,

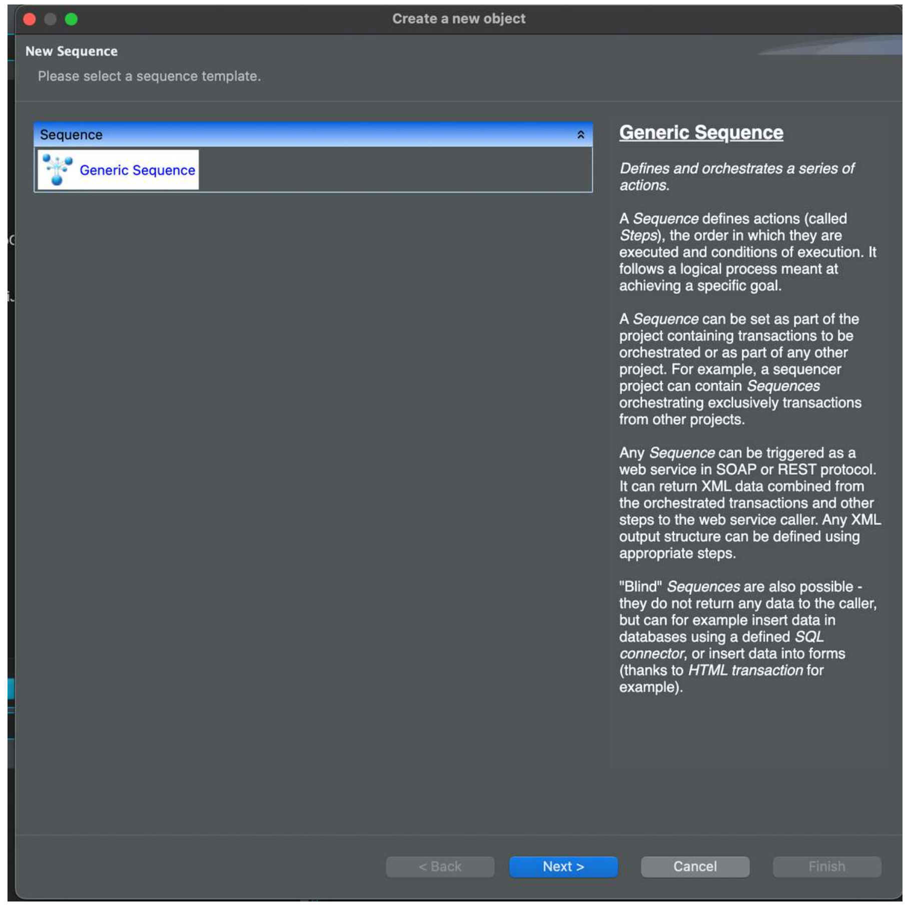
select **New >**, then click on **Sequence**.



The **Create a new object window** appears.

In the **Create a new object window**,

select **Generic Sequence**, then click on **Next >**



Generic Sequence

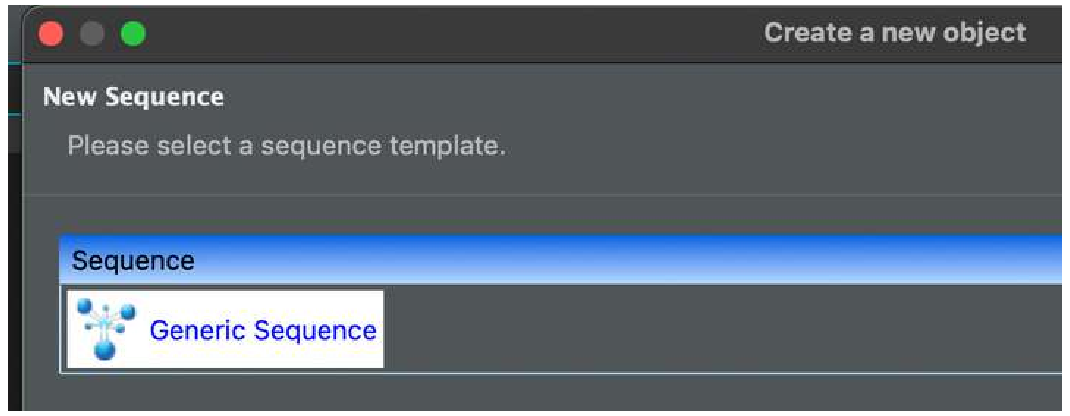
Defines and orchestrates a series of actions.

A *Sequence* defines actions (called *Steps*), the order in which they are executed and conditions of execution. It follows a logical process meant at achieving a specific goal.

A *Sequence* can be set as part of the project containing transactions to be orchestrated or as part of any other project. For example, a sequencer project can contain *Sequences* orchestrating exclusively transactions from other projects.

Any *Sequence* can be triggered as a web service in SOAP or REST protocol. It can return XML data combined from the orchestrated transactions and other steps to the web service caller. Any XML output structure can be defined using appropriate steps.

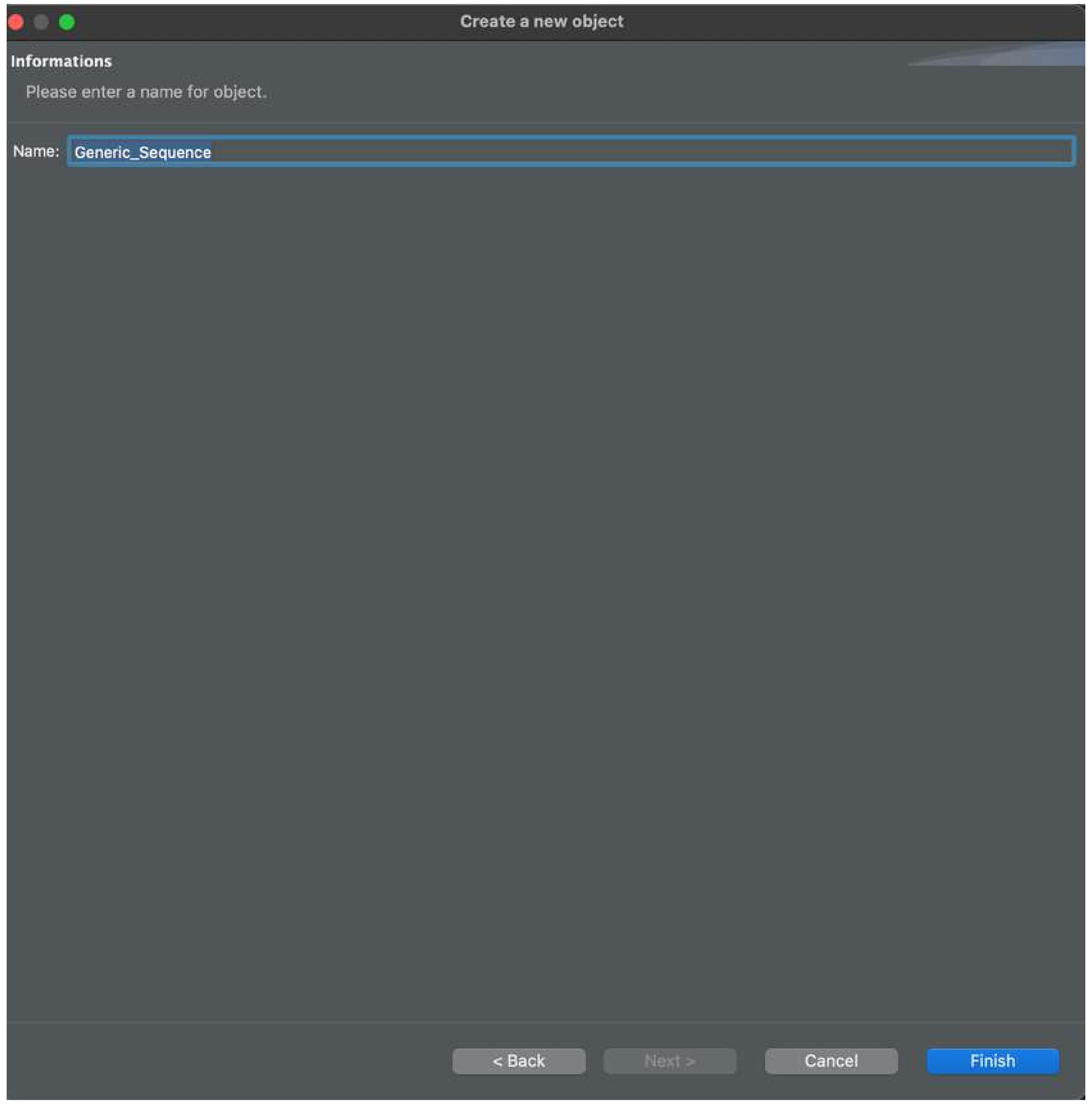
"Blind" *Sequences* are also possible - they do not return any data to the caller, but can for example insert data in databases using a defined *SQL connector*, or insert data into forms (thanks to *HTML transaction* for example).



4.5 Create a sequence

In the **Create a new object window**,

rename the sequence and click on **Finish**.

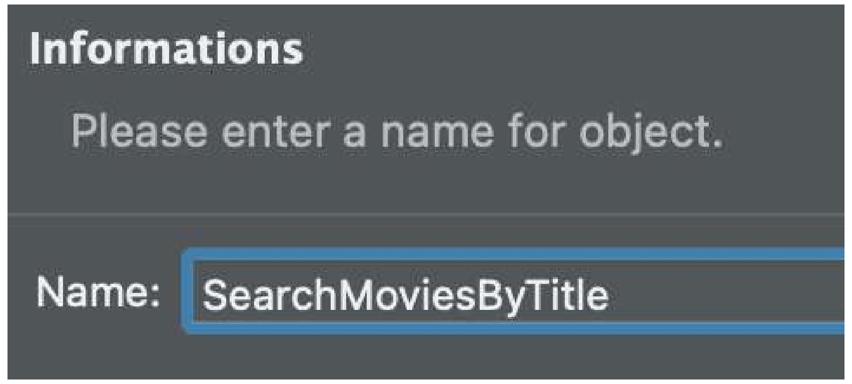


Create a new object

Informations

Please enter a name for object.

Name: Generic_Sequence



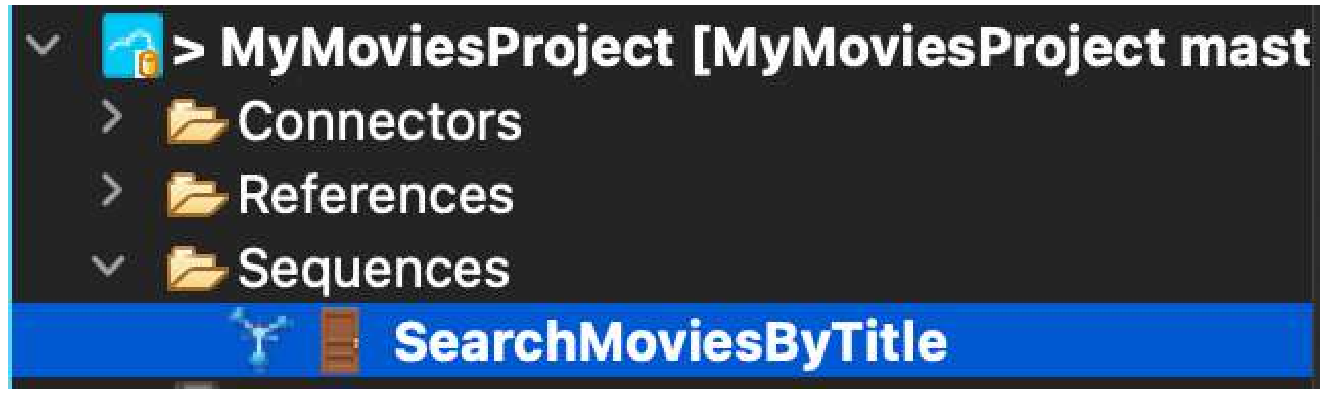
Informations

Please enter a name for object.

Name: SearchMoviesByTitle

A **Sequences folder** and the **created sequence**

appear in the tree structure.



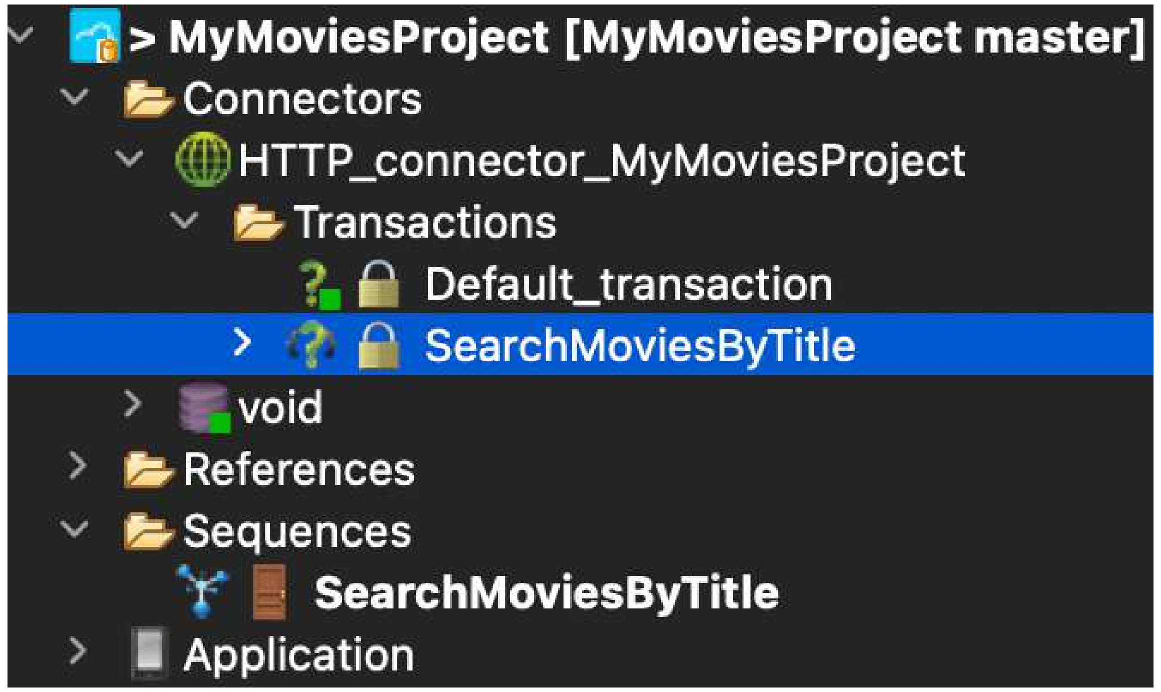
4.6 Call a transaction from a sequence

Import the transaction in the sequence

Once the sequence is created,

you need to

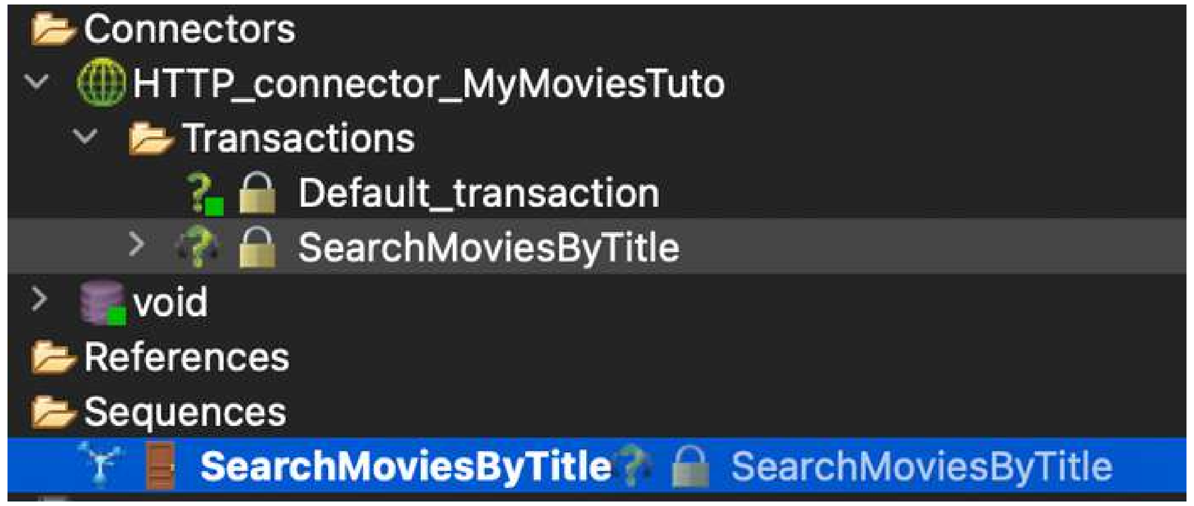
import the transaction in the sequence.



```
> MyMoviesProject [MyMoviesProject master]
  Connectors
    HTTP_connector_MyMoviesProject
      Transactions
        Default_transaction
        SearchMoviesByTitle
    void
  References
  Sequences
    SearchMoviesByTitle
  Application
```

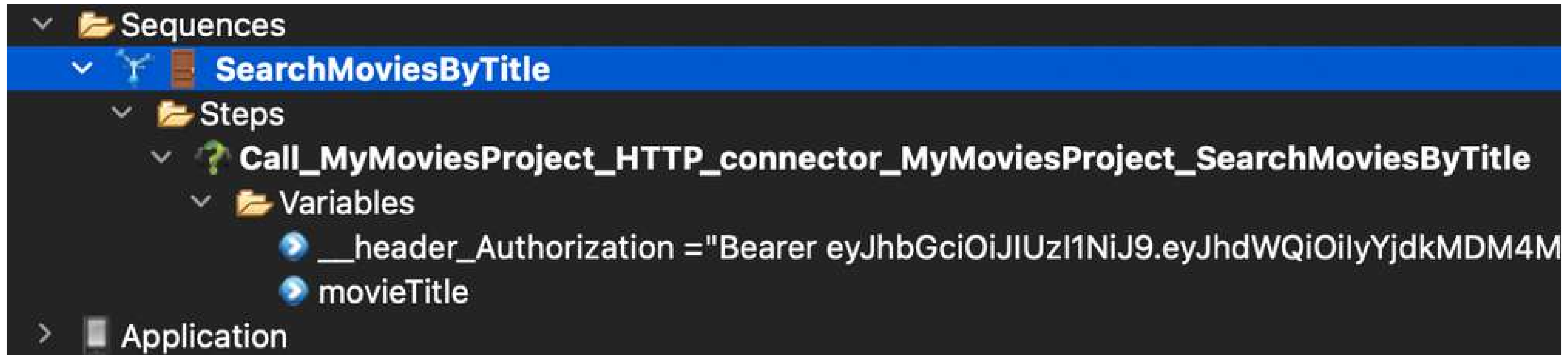
Drag-and-drop the transaction in the sequence

while clicking on **Option** in MacOs or **Control** in Windows.



```
Connectors
  HTTP_connector_MyMoviesTuto
    Transactions
      Default_transaction
      SearchMoviesByTitle
  void
References
Sequences
  SearchMoviesByTitle SearchMoviesByTitle
```

This creates a **Steps folder** where a **call to the transaction** appears.



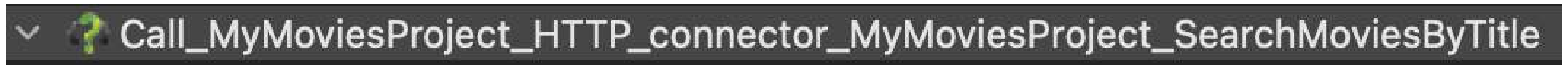
```
Sequences
  SearchMoviesByTitle
    Steps
      Call_MyMoviesProject_HTTP_connector_MyMoviesProject_SearchMoviesByTitle
        Variables
          __header_Authorization ="Bearer eyJhbGciOiJIUzI1NiJ9.eyJhdWQiOilyYjdkMDM4M
          movieTitle
Application
```

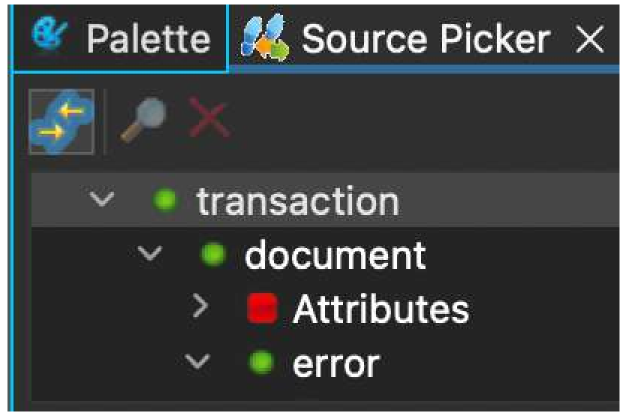
4.6 Call a transaction from a sequence

Update the transaction schema

Reminder : This step can and usually should be done just after creating the transaction.

Double-click on the Call Transaction step to display the source picker.

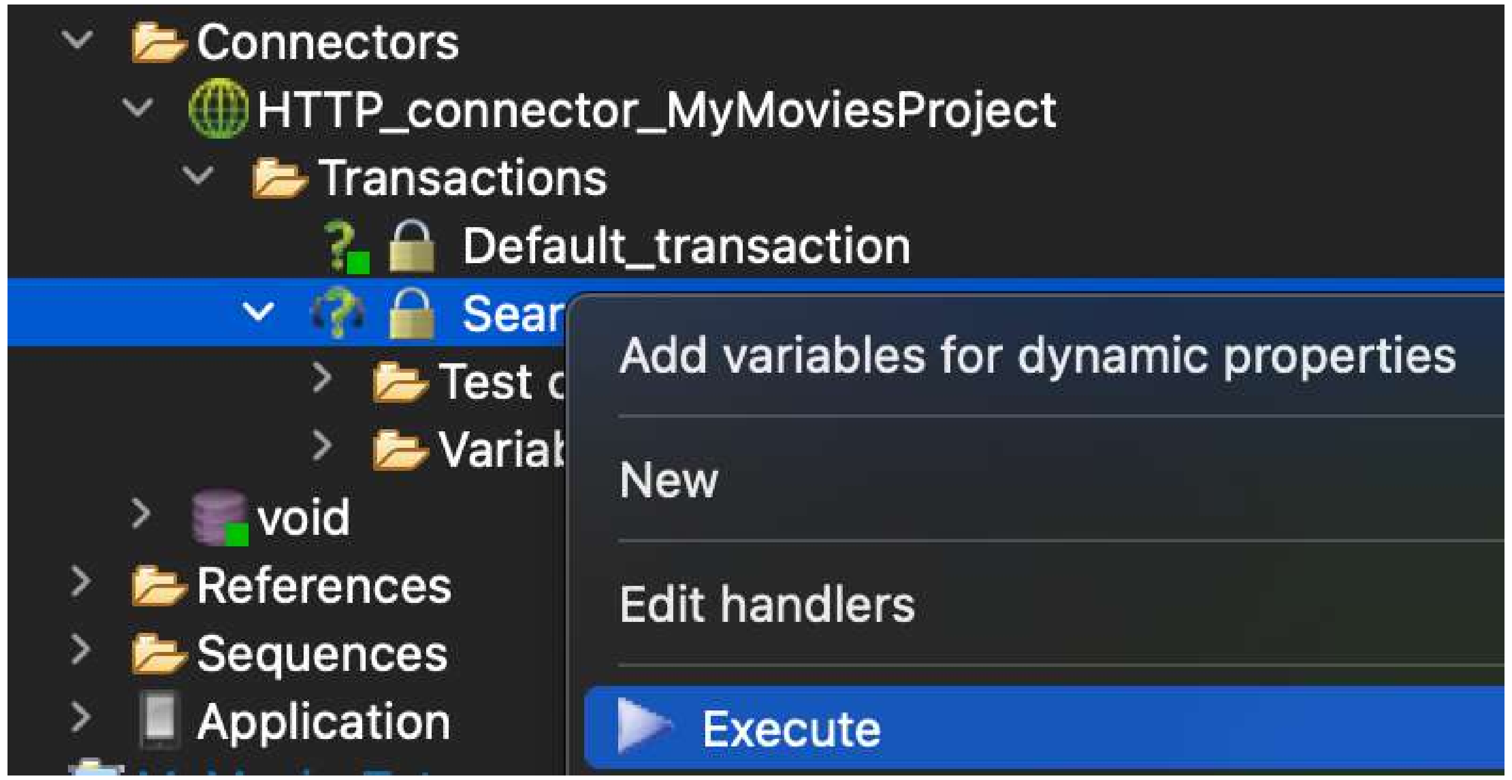

Call_MyMoviesProject_HTTP_connector_MyMoviesProject_SearchMoviesByTitle



The **schema shown in the picker** does **not contain response elements**, you need to **update the transaction schema**.

There are 2 cases : **a transaction with or without variables**.

If the transaction **doesn't need variables**, you can right-click on the source transaction and choose **Execute** to generate response data.

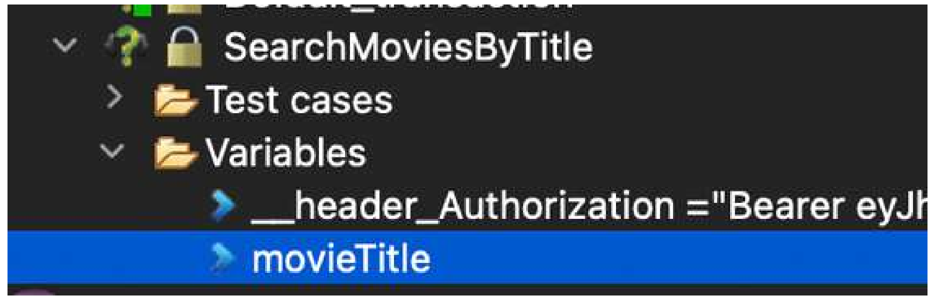


4.6 Call a transaction from a sequence

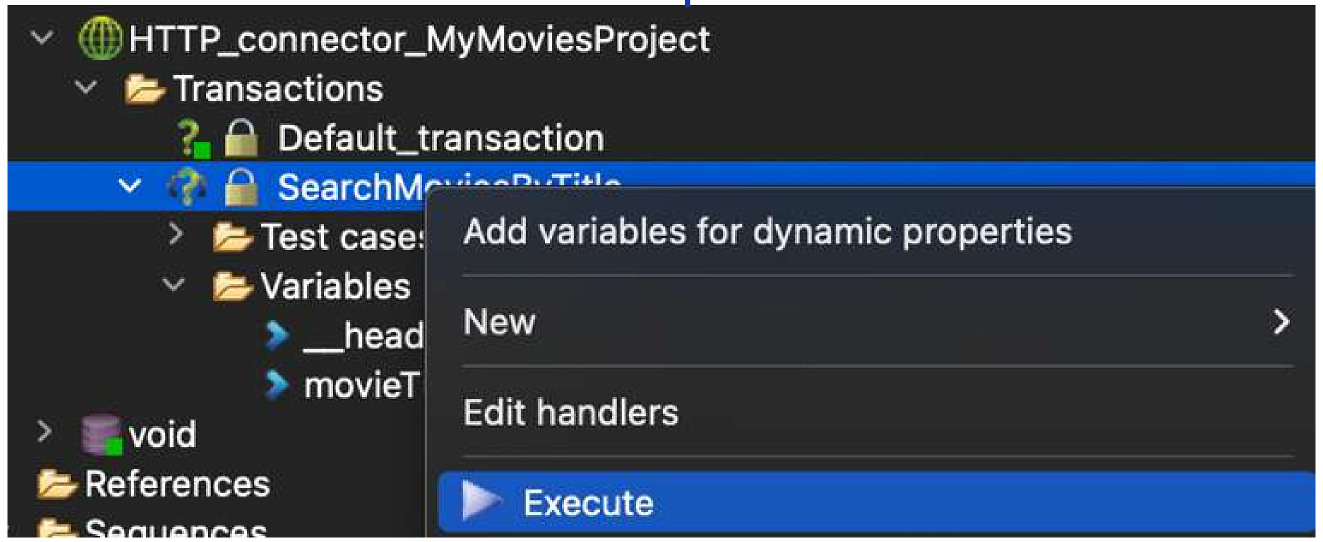
Update the transaction schema

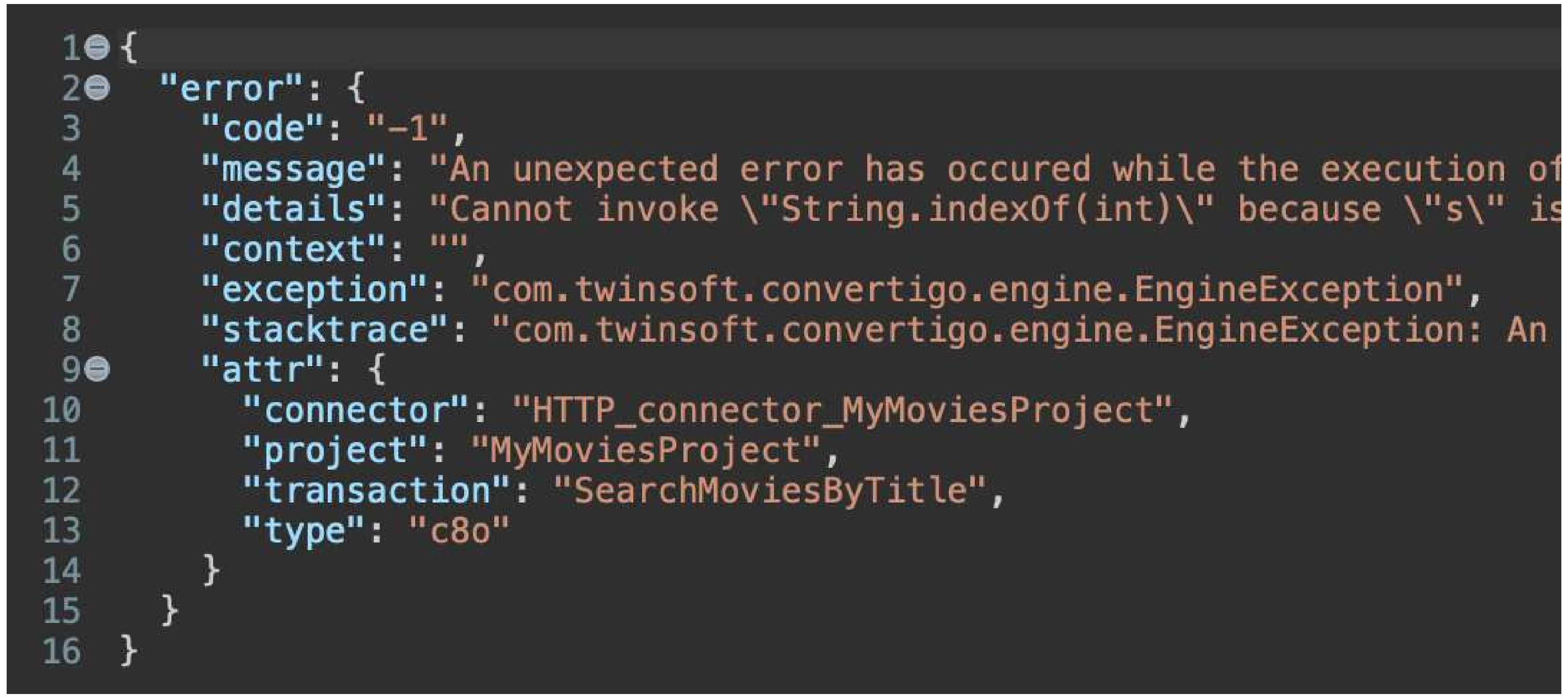
In our case, we have 2 variables :

- **_header_Authorization** which has already a value
- **movieTitle** whose value is empty.



Executing the transaction as it is will result in an error response.





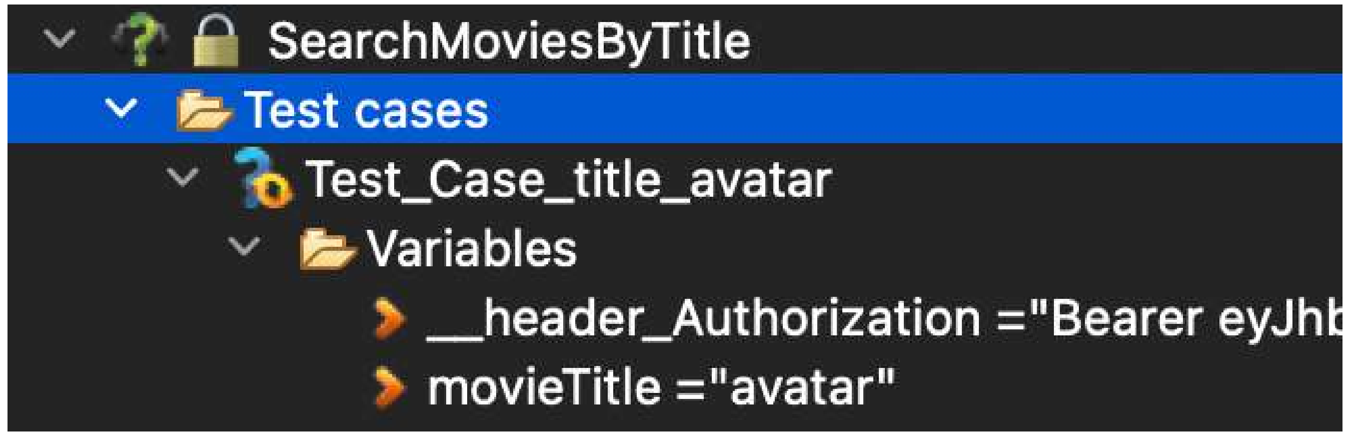
```
{
  "error": {
    "code": "-1",
    "message": "An unexpected error has occured while the execution of
    "details": "Cannot invoke \"String.indexOf(int)\" because \"s\" is
    "context": "",
    "exception": "com.twinsoft.convertigo.engine.EngineException",
    "stacktrace": "com.twinsoft.convertigo.engine.EngineException: An
    "attr": {
      "connector": "HTTP_connector_MyMoviesProject",
      "project": "MyMoviesProject",
      "transaction": "SearchMoviesByTitle",
      "type": "c8o"
    }
  }
}
```

4.6 Call a transaction from a sequence

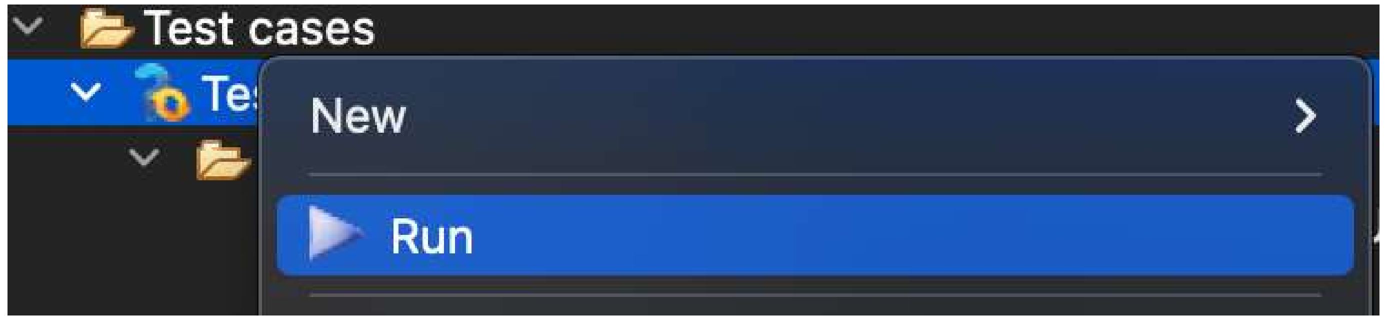
Update the transaction schema

To get a valid response, you need to use the test case you created before in the transaction SearchMoviesByTitle. In this test case, the variable **movieTitle** has already a value.

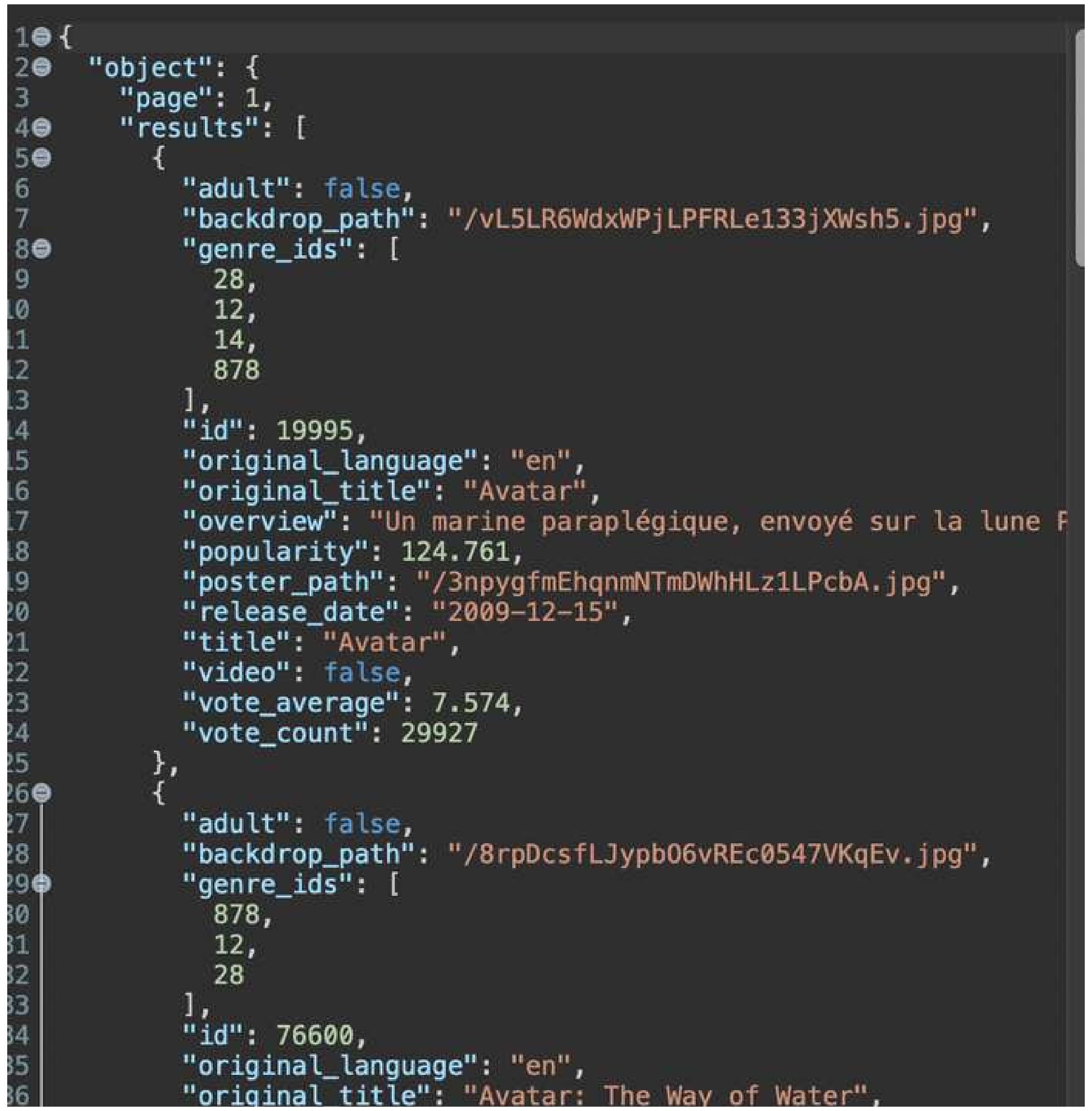
Right-click on the test case.



Choose **Run** to generate response data.



The results are displayed in the editors panel.

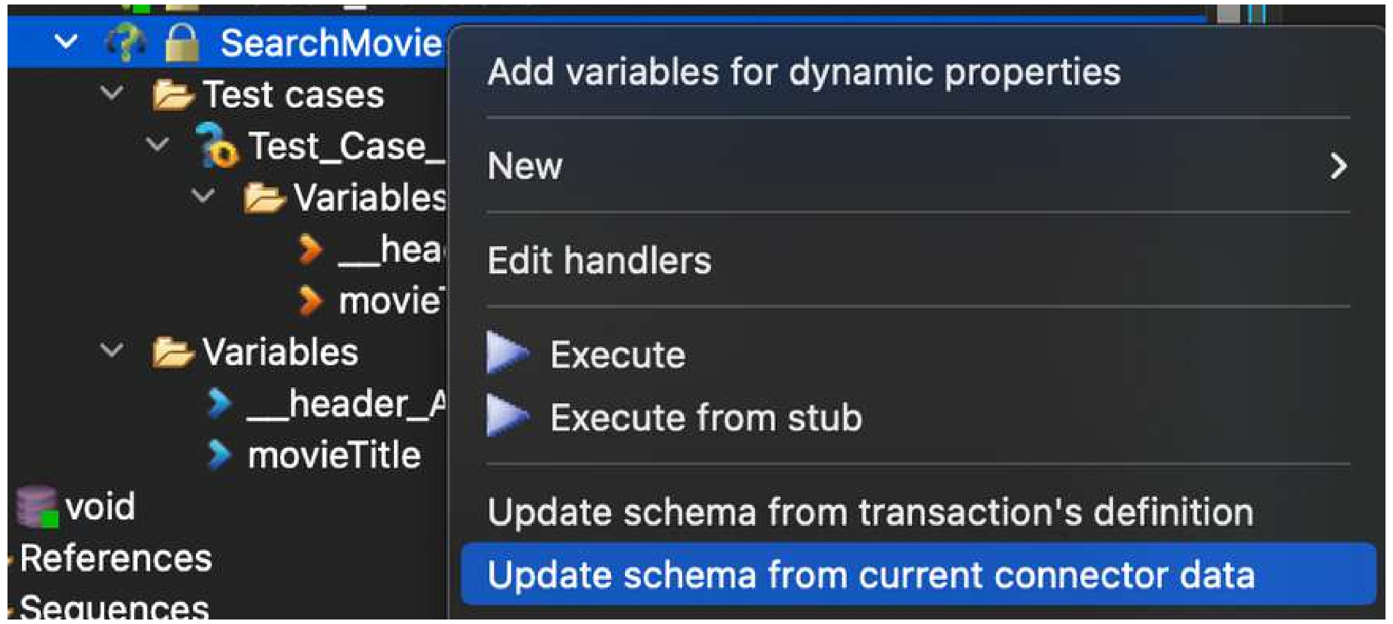


```
{
  "object": {
    "page": 1,
    "results": [
      {
        "adult": false,
        "backdrop_path": "/vL5LR6WdxWPjLPFRLe133jXWsh5.jpg",
        "genre_ids": [
          28,
          12,
          14,
          878
        ],
        "id": 19995,
        "original_language": "en",
        "original_title": "Avatar",
        "overview": "Un marine paraplégique, envoyé sur la lune
        "popularity": 124.761,
        "poster_path": "/3npygfmEhqnmNTmDWhHLz1LPcbA.jpg",
        "release_date": "2009-12-15",
        "title": "Avatar",
        "video": false,
        "vote_average": 7.574,
        "vote_count": 29927
      },
      {
        "adult": false,
        "backdrop_path": "/8rpDcsfLJypbO6vREc0547VKqEv.jpg",
        "genre_ids": [
          878,
          12,
          28
        ],
        "id": 76600,
        "original_language": "en",
        "original_title": "Avatar: The Way of Water",
```

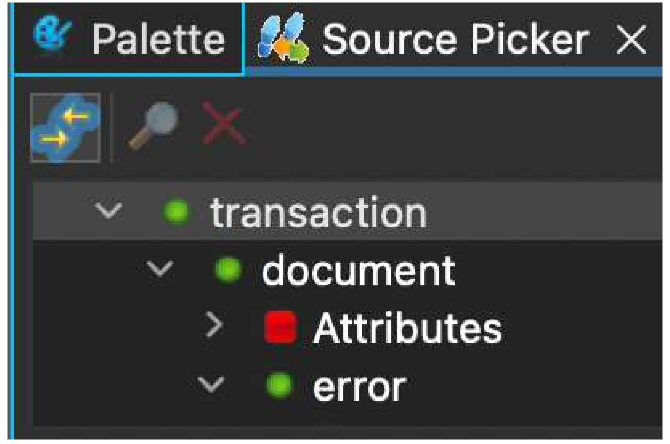
4.6 Call a transaction from a sequence

Update the transaction schema

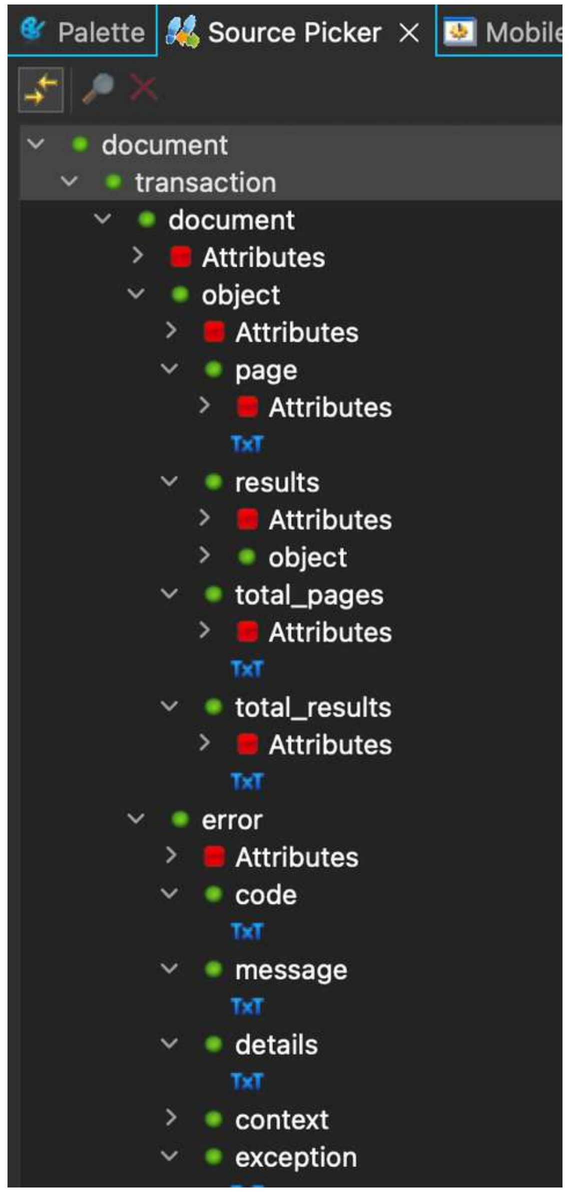
Right-click on the source transaction **SearchMoviesByTitle** and select **Update schema from current connector data**.



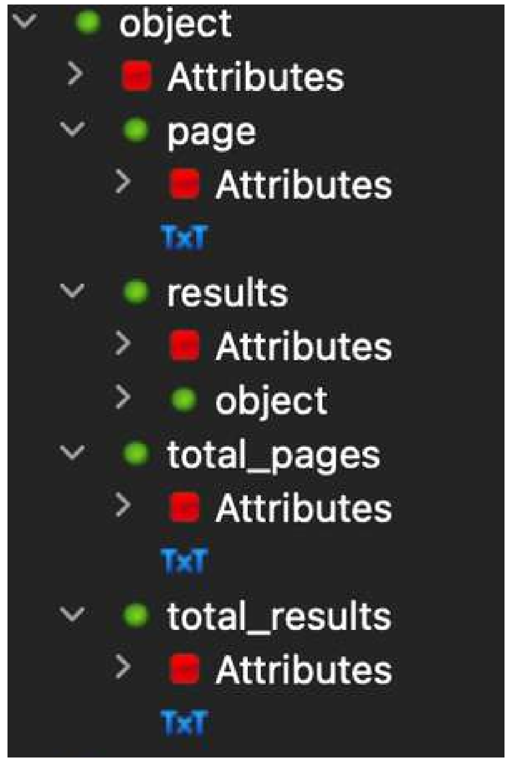
Before updating the schema



Double-click once again on the **Call Transaction Step** to display the **updated transaction schema** in the Source Picker.



After updating the schema, an **object node** appears in the source picker

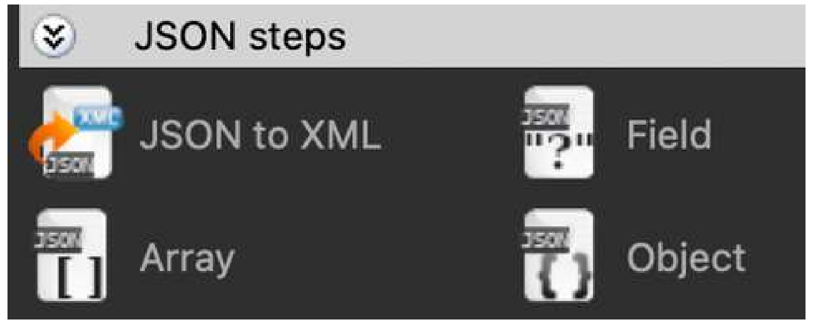


4.7 Create a custom data structure

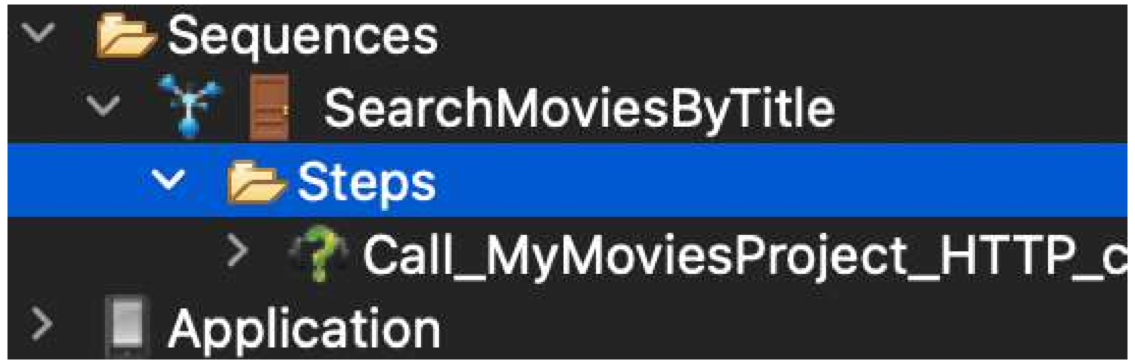
Your sequence is now calling the **transaction SearchMoviesByTitle** which is a **JSON HTTP Transaction**.

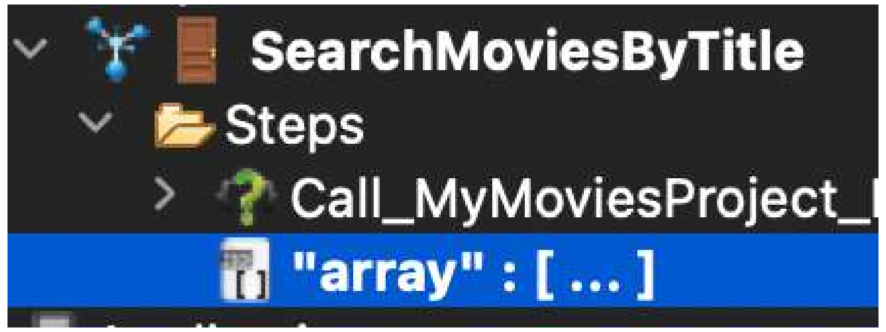
To construct our own **response data structure**

from the transaction's r**esponse data**, let's use the Array Step (JSON step).

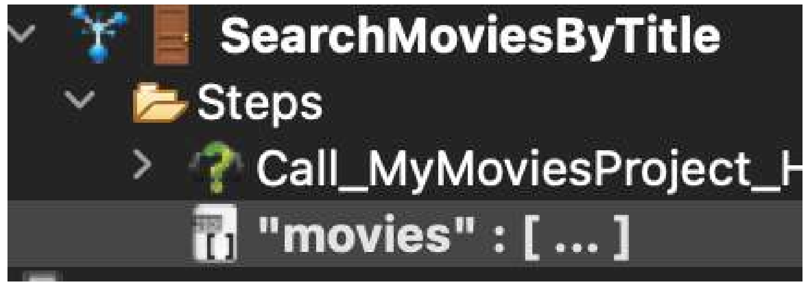


Drag the **Array step** from the palette and drop it into the **steps folder of your sequence** after the **transaction call**.

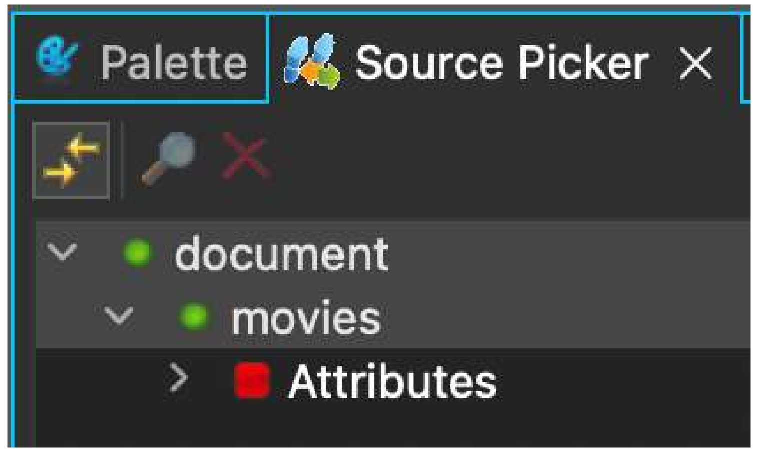




Rename it movies.

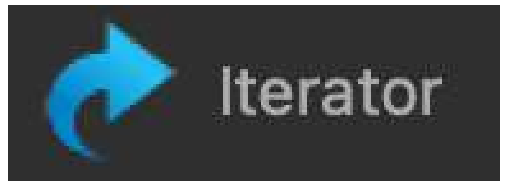


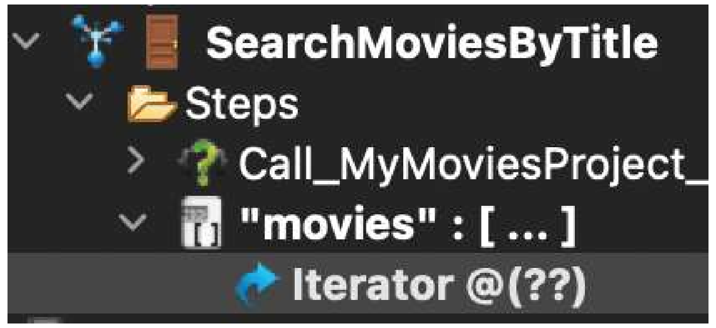
It appears as movies in the source picker.



4.7 Create a custom data structure

Drag the **Iterator** step from the palette and drop it into the step **movies** in your sequence.

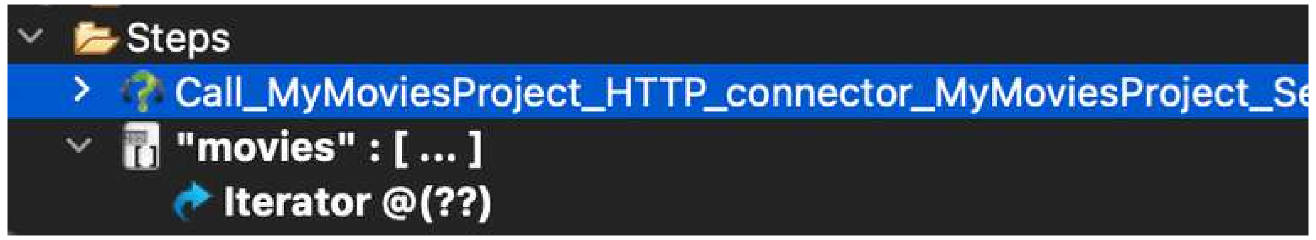

Iterator



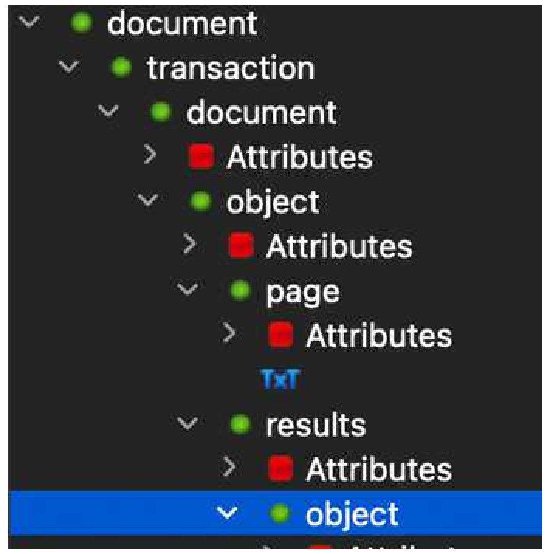
```
SearchMoviesByTitle
  Steps
    Call_MyMoviesProject_
    "movies" : [ ... ]
      Iterator @(??)
```

Then, double-click on the transaction call in your sequence

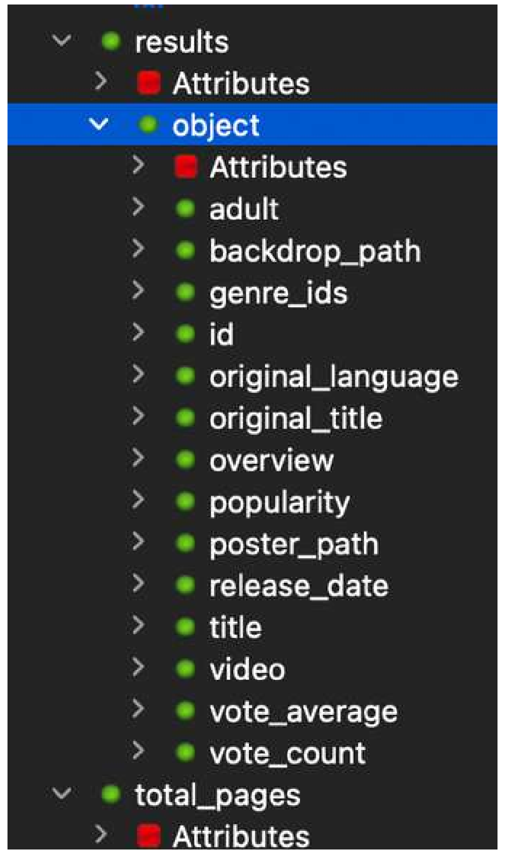
to open the source picker.



```
Steps
  Call_MyMoviesProject_HTTP_connector_MyMoviesProject_Se
  "movies" : [ ... ]
    Iterator @(??)
```



```
document
  transaction
    document
      Attributes
      object
        Attributes
        page
          Attributes
          TxT
        results
          Attributes
          object
```

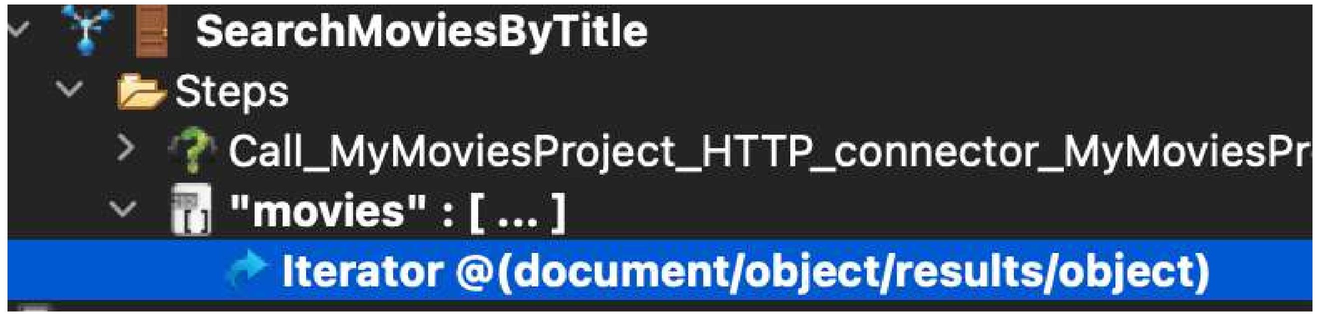


```
results
  Attributes
  object
    Attributes
    adult
    backdrop_path
    genre_ids
    id
    original_language
    original_title
    overview
    popularity
    poster_path
    release_date
    title
    video
    vote_average
    vote_count
total_pages
  Attributes
```

In the source picker, expand the **results node.**

Then drag and drop the **object node** directly into your iterator.

This **object node** provides the information you want in your iterator.

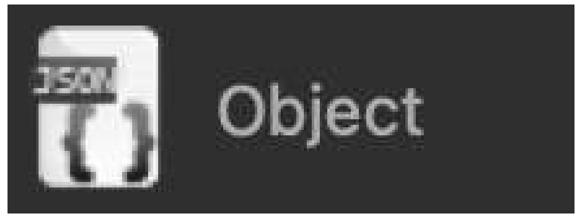


```
SearchMoviesByTitle
  Steps
    Call_MyMoviesProject_HTTP_connector_MyMoviesPr
    "movies" : [ ... ]
      Iterator @(document/object/results/object)
```

4.7 Create a custom data structure

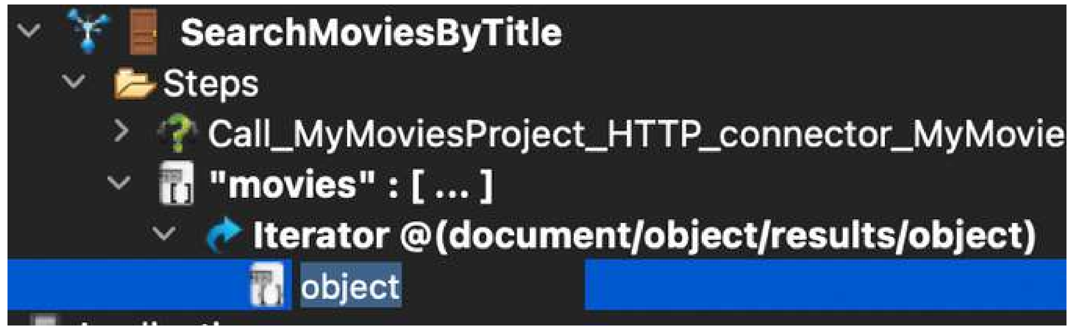
Drag the **Object step** from the palette

and drop it into the **Iterator step** in your sequence.



JSON

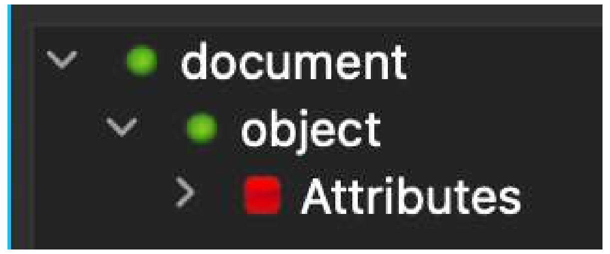
Object



```
SearchMoviesByTitle
  Steps
    Call_MyMoviesProject_HTTP_connector_MyMovie
    "movies" : [ ... ]
      Iterator @(document/object/results/object)
        object
```

In the source picker,

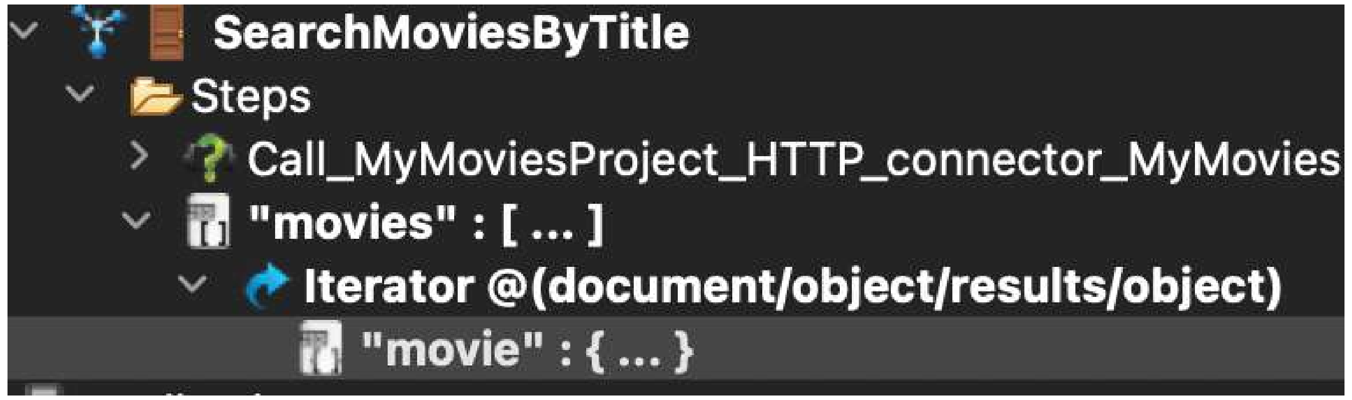
it appears as **object**.



```
document
  object
    Attributes
```

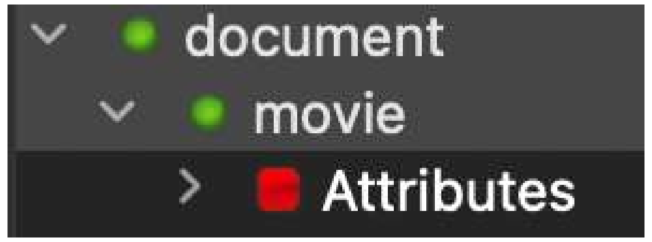
The Object step is a container for the various elements you'll add to it.

Let's rename it **movie** in the treeview.



```
SearchMoviesByTitle
  Steps
    Call_MyMoviesProject_HTTP_connector_MyMovies
    "movies" : [ ... ]
      Iterator @(document/object/results/object)
        "movie" : { ... }
```

Now, in the source picker, it appears as movie.



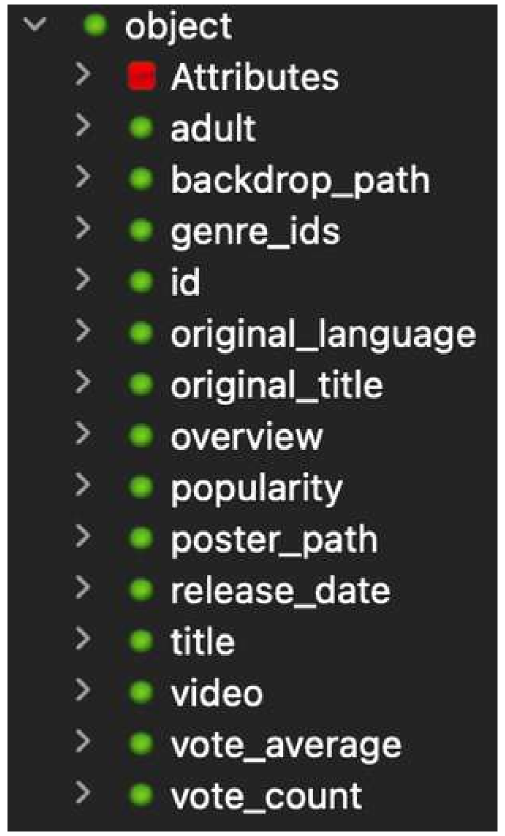
```
document
  movie
    Attributes
```

4.7 Create a custom data structure

In the **response data** from the transaction,

for each item, we receive an object movie with many fields, as shown in the source picker.

In our application, we only need a few of them and we're going to select the fields that interest us.

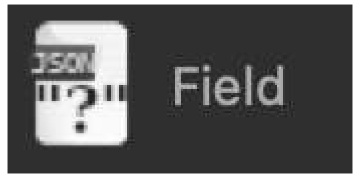


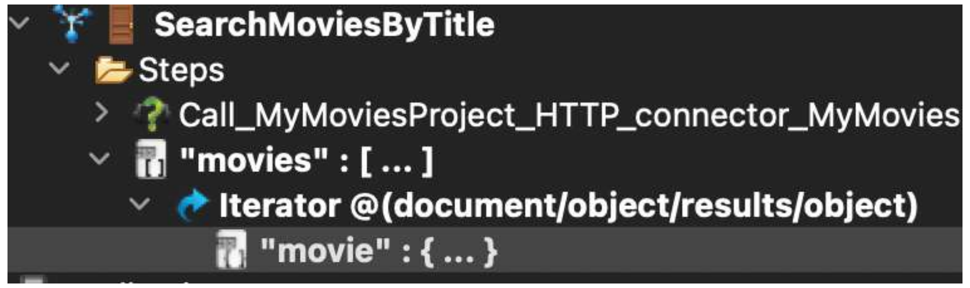
```
object
  Attributes
  adult
  backdrop_path
  genre_ids
  id
  original_language
  original_title
  overview
  popularity
  poster_path
  release_date
  title
  video
  vote_average
  vote_count
```

Let's say I want the following fields displayed in the front-end :

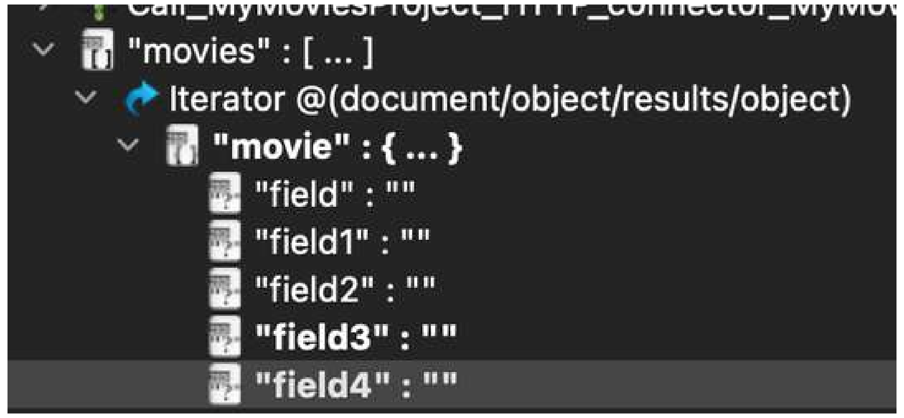
- title
- overview
- poster_path
- release_date
- original_title

Drag the Field step from the palette and drop it 5 times into the Object step movie.





```
SearchMoviesByTitle
  Steps
    Call_MyMoviesProject_HTTP_connector_MyMovies
    "movies" : [ ... ]
      Iterator @(document/object/results/object)
        "movie" : { ... }
```

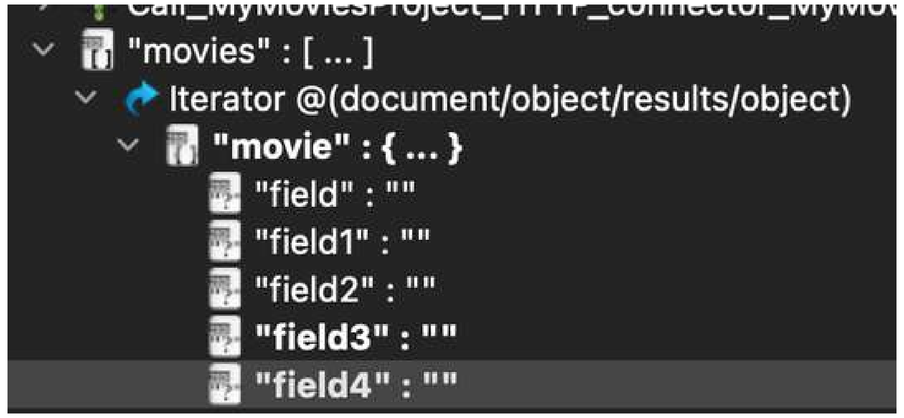


```
    Call_MyMoviesProject_HTTP_connector_MyMov
  "movies" : [ ... ]
    Iterator @(document/object/results/object)
      "movie" : { ... }
        "field" : ""
        "field1" : ""
        "field2" : ""
        "field3" : ""
        "field4" : ""
```

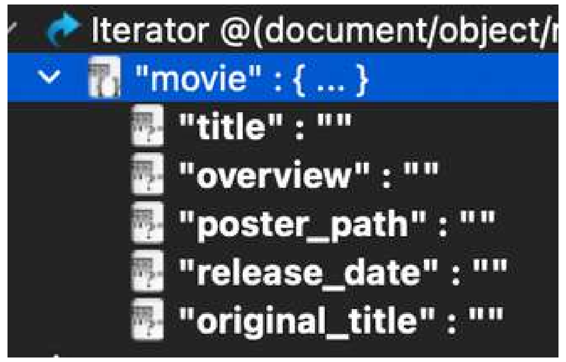
4.7 Create a custom data structure

Click twice on the movie step to display it in the source picker.

Rename the 5 fields in movie as title, overview, poster_path, release_date, original_title

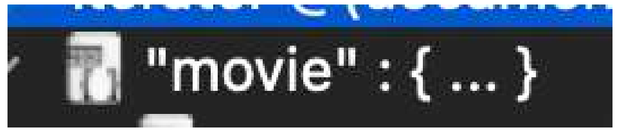


```
  Call_MyMoviesProject_HTTP_connector_MyMo
  "movies" : [ ... ]
    Iterator @(document/object/results/object)
      "movie" : { ... }
        "field" : ""
        "field1" : ""
        "field2" : ""
        "field3" : ""
        "field4" : ""
```

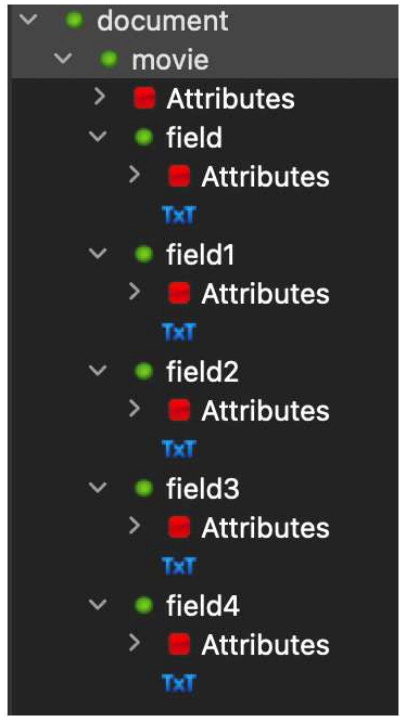


```
Iterator @(document/object/r
  "movie" : { ... }
    "title" : ""
    "overview" : ""
    "poster_path" : ""
    "release_date" : ""
    "original_title" : ""
```

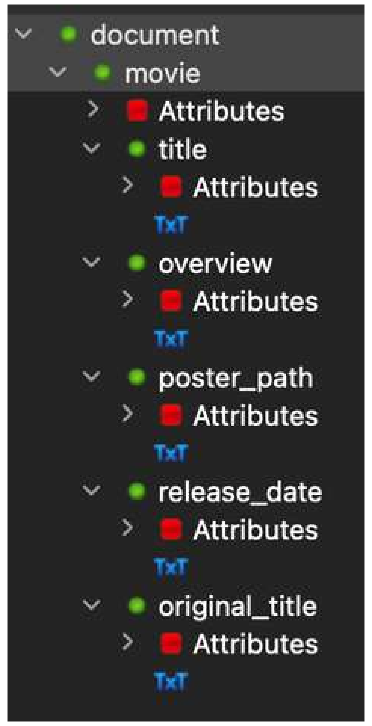
In the source picker,
you can see that the 5 fields in movie have been renamed as well.



```
Iterator @(documen
  "movie" : { ... }
```

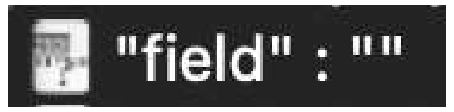


```
document
  movie
    Attributes
    field
      Attributes
      TxT
    field1
      Attributes
      TxT
    field2
      Attributes
      TxT
    field3
      Attributes
      TxT
    field4
      Attributes
      TxT
```

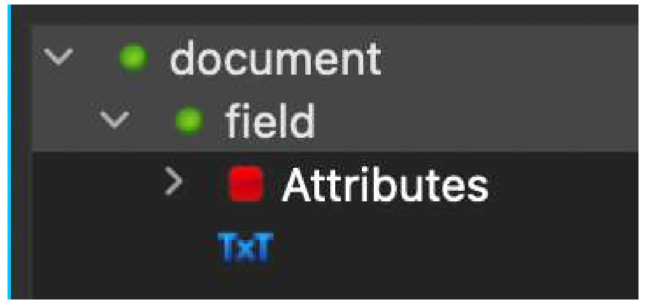


```
document
  movie
    Attributes
    title
      Attributes
      TxT
    overview
      Attributes
      TxT
    poster_path
      Attributes
      TxT
    release_date
      Attributes
      TxT
    original_title
      Attributes
      TxT
```

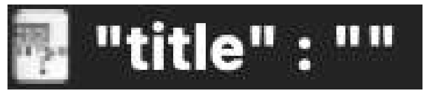
The structure of each field has been renamed



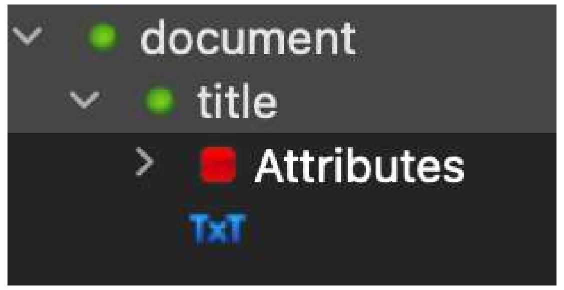
```
"field" : ""
```



```
document
  field
    Attributes
    TxT
```



```
"title" : ""
```

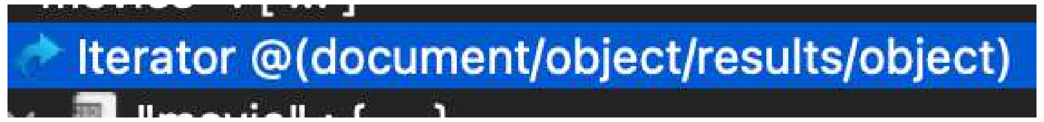


```
document
  title
    Attributes
    TxT
```

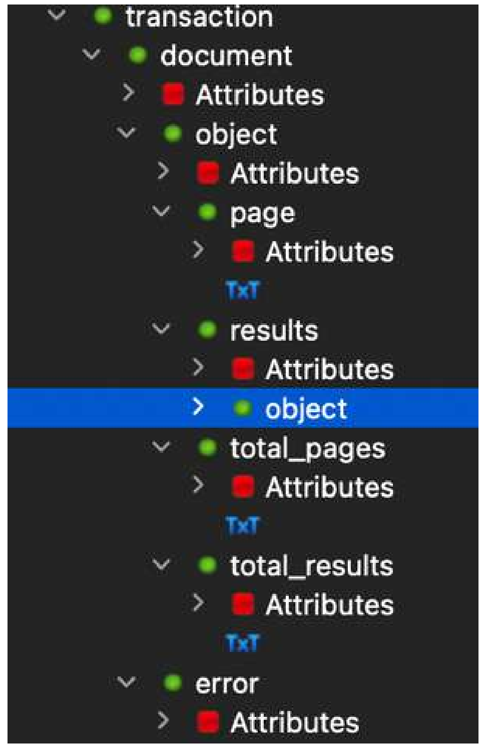
4.7 Create a custom data structure

Now we want to bind these fields to the values of the fields in the iterator.

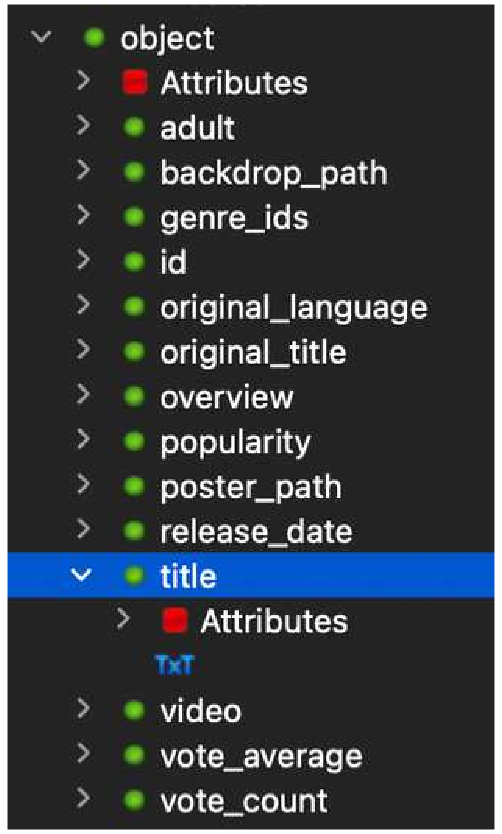
Double-click on the **Iterator step** to display its data in the Source Picker and open the object node.



```
Iterator @(document/object/results/object)
  "movie" : { ... }
```

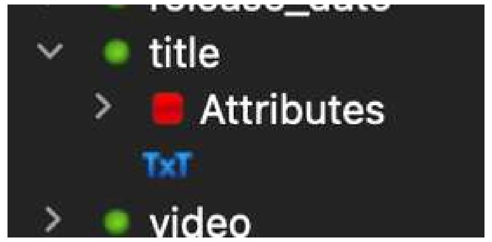


```
transaction
  document
    Attributes
    object
      Attributes
      page
        Attributes
        TxT
      results
        Attributes
        object
      total_pages
        Attributes
        TxT
      total_results
        Attributes
        TxT
    error
      Attributes
```

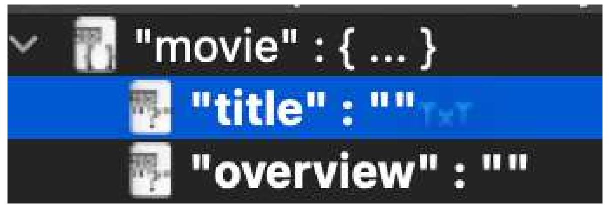


```
object
  Attributes
  adult
  backdrop_path
  genre_ids
  id
  original_language
  original_title
  overview
  popularity
  poster_path
  release_date
  title
    Attributes
    TxT
  video
  vote_average
  vote_count
```

Drag and drop the **TxT element** corresponding to the required information into the various steps of the element.

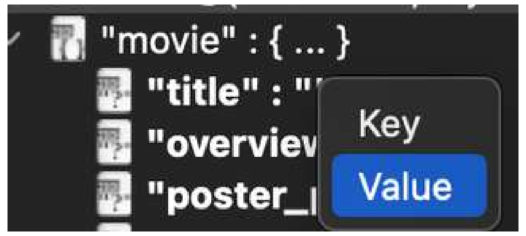


```
  title
    Attributes
    TxT
  video
```

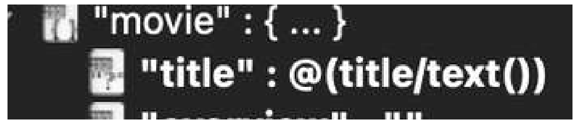


```
"movie" : { ... }
  "title" : ""
  "overview" : ""
```

Choose **Value** each time you are prompted to set the value property of the step.

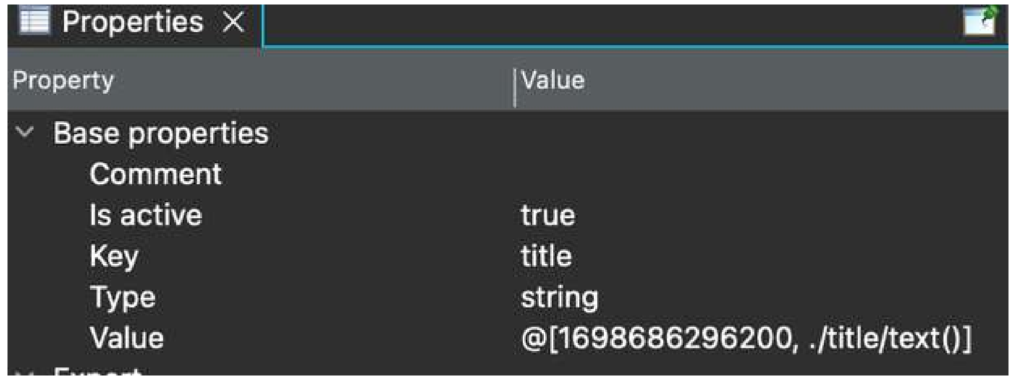


```
"movie" : { ... }
  "title" : "
  "overview
  "poster_
```



```
"movie" : { ... }
  "title" : @(title/text())
```

In properties, the value appears as binded.

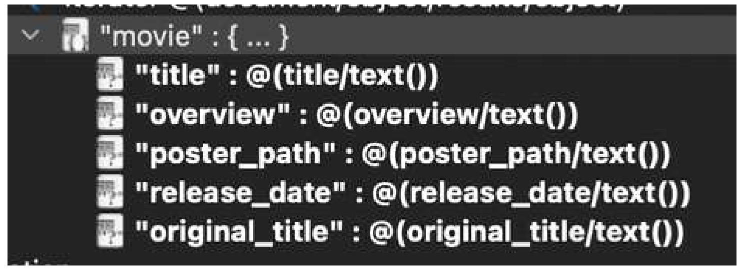


Properties

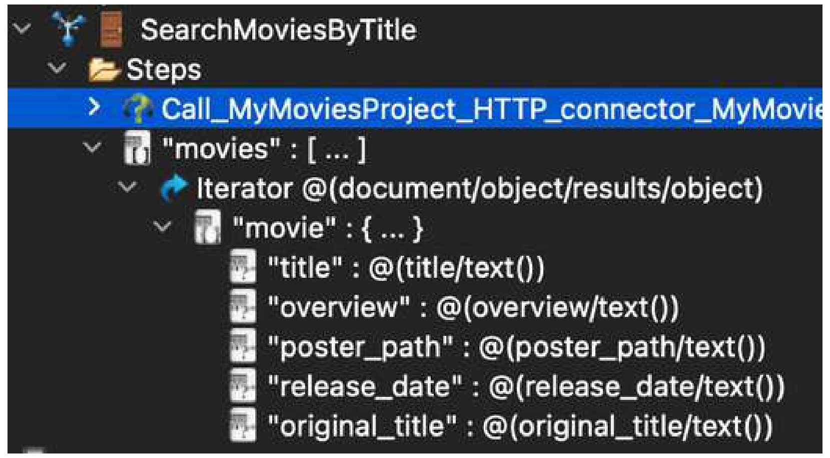
Property	Value
Base properties	
Comment	
Is active	true
Key	title
Type	string
Value	@[1698686296200, ./title/text()]

4.7 Create a custom data structure

Repeat the same operation for the 5 fields.

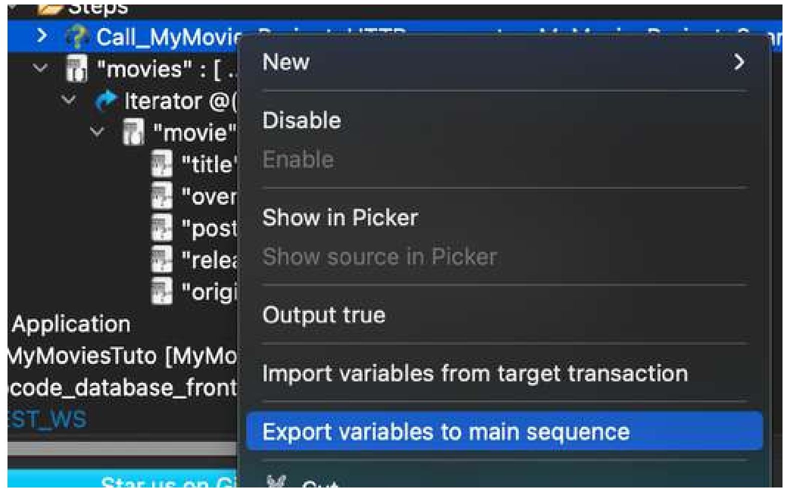


```
"movie" : { ... }
    "title" : @(title/text())
    "overview" : @(overview/text())
    "poster_path" : @(poster_path/text())
    "release_date" : @(release_date/text())
    "original_title" : @(original_title/text())
```

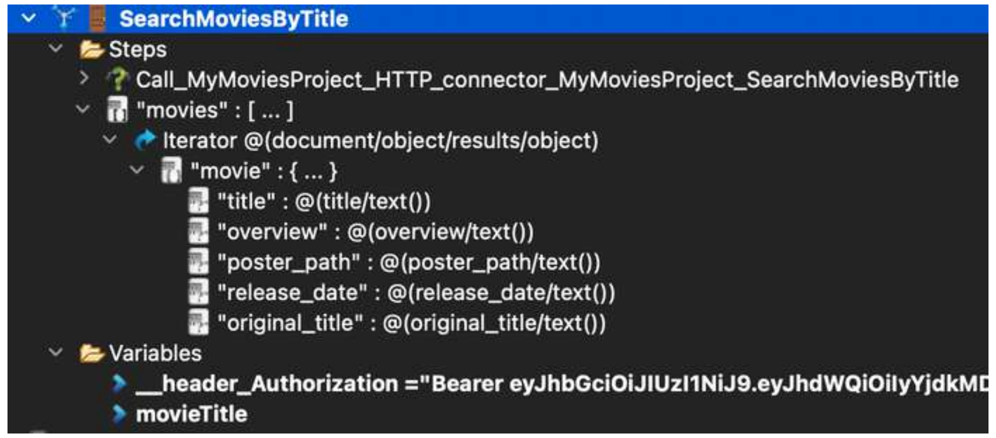


```
SearchMoviesByTitle
  Steps
    Call_MyMoviesProject_HTTP_connector_MyMovie
    "movies" : [ ... ]
      Iterator @(document/object/results/object)
        "movie" : { ... }
          "title" : @(title/text())
          "overview" : @(overview/text())
          "poster_path" : @(poster_path/text())
          "release_date" : @(release_date/text())
          "original_title" : @(original_title/text())
```

Now we want to **import the variables of the transaction into the sequence**,

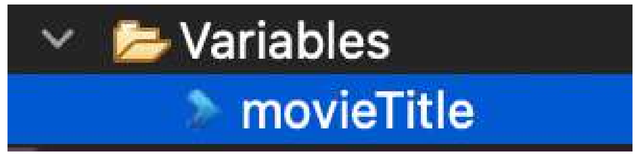

New
Disable
Enable
Show in Picker
Show source in Picker
Output true
Import variables from target transaction
Export variables to main sequence

Right-click on the **transaction call**, and select **Export variables to main sequences**.



```
SearchMoviesByTitle
  Steps
    Call_MyMoviesProject_HTTP_connector_MyMoviesProject_SearchMoviesByTitle
    "movies" : [ ... ]
      Iterator @(document/object/results/object)
        "movie" : { ... }
          "title" : @(title/text())
          "overview" : @(overview/text())
          "poster_path" : @(poster_path/text())
          "release_date" : @(release_date/text())
          "original_title" : @(original_title/text())
  Variables
    __header_Authorization ="Bearer eyJhbGciOiJIUzI1NiJ9.eyJhdWQiOiIyYjdkMD
    movieTitle
```

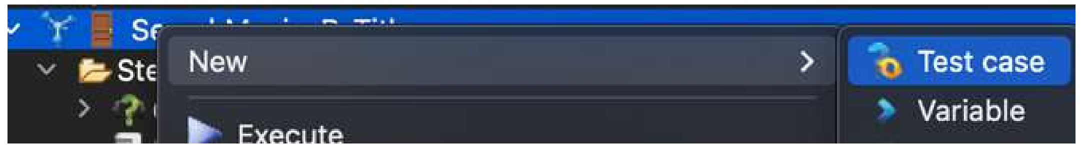
A folder **Variables** has been added to the sequence.

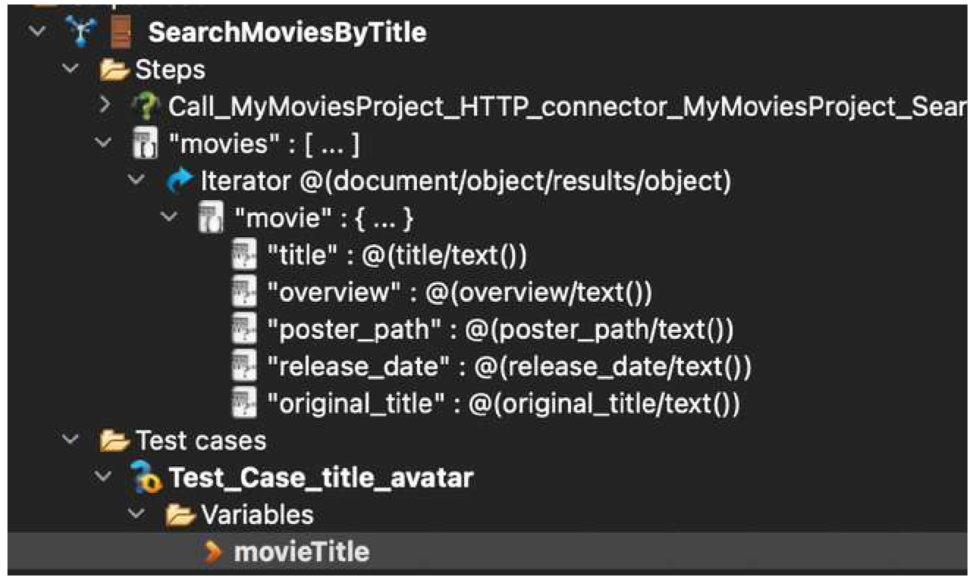


```
Variables
  movieTitle
```

4.8 Test the sequence

Now, let’s create a test case for the sequence
(as shown in the previous slides for the transaction SearchMoviesByTitle).

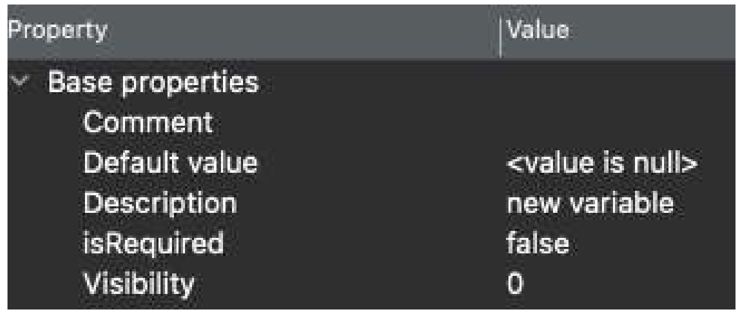




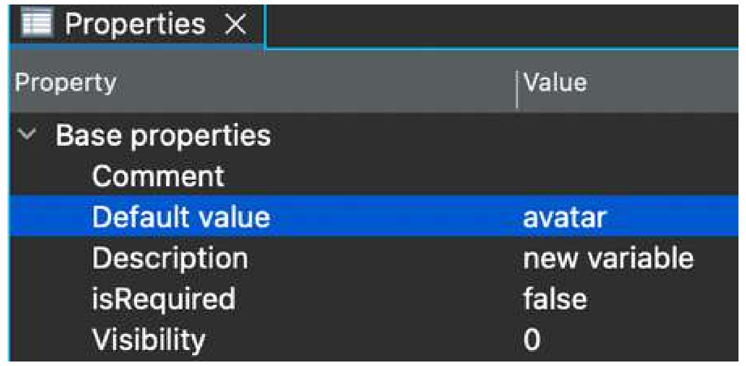
```
SearchMoviesByTitle
  Steps
    Call_MyMoviesProject_HTTP_connector_MyMoviesProject_Sear
    "movies" : [ ... ]
      Iterator @(document/object/results/object)
        "movie" : { ... }
          "title" : @(title/text())
          "overview" : @(overview/text())
          "poster_path" : @(poster_path/text())
          "release_date" : @(release_date/text())
          "original_title" : @(original_title/text())
  Test cases
    Test_Case_title_avatar
      Variables
        movieTitle
```

Click on the variable movieTitle in the test case.

In properties, change the Default value of movieTitle to “avatar“.



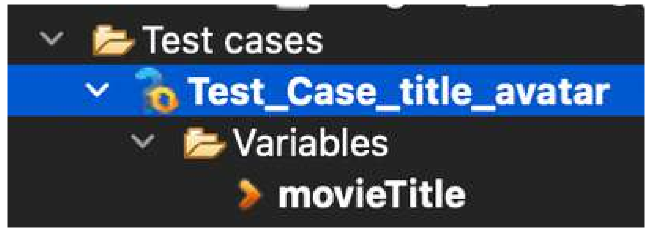
Property	Value
Base properties	
Comment	
Default value	<value is null>
Description	new variable
isRequired	false
Visibility	0



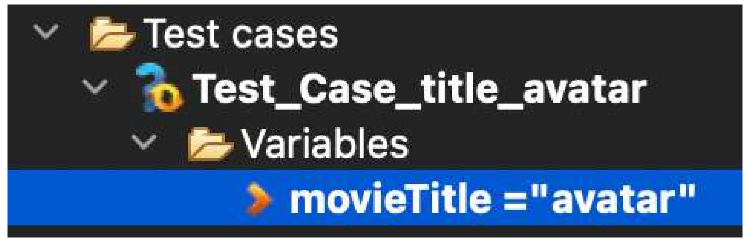
Properties

Property	Value
Base properties	
Comment	
Default value	avatar
Description	new variable
isRequired	false
Visibility	0

The value appears in the treeview.



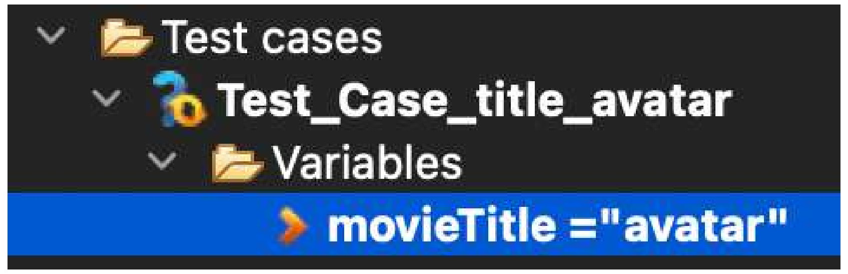
```
Test cases
  Test_Case_title_avatar
    Variables
      movieTitle
```

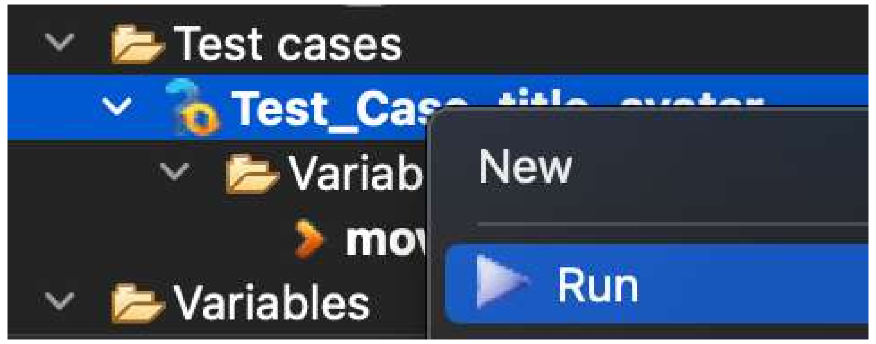


```
Test cases
  Test_Case_title_avatar
    Variables
      movieTitle ="avatar"
```

4.8 Test the sequence

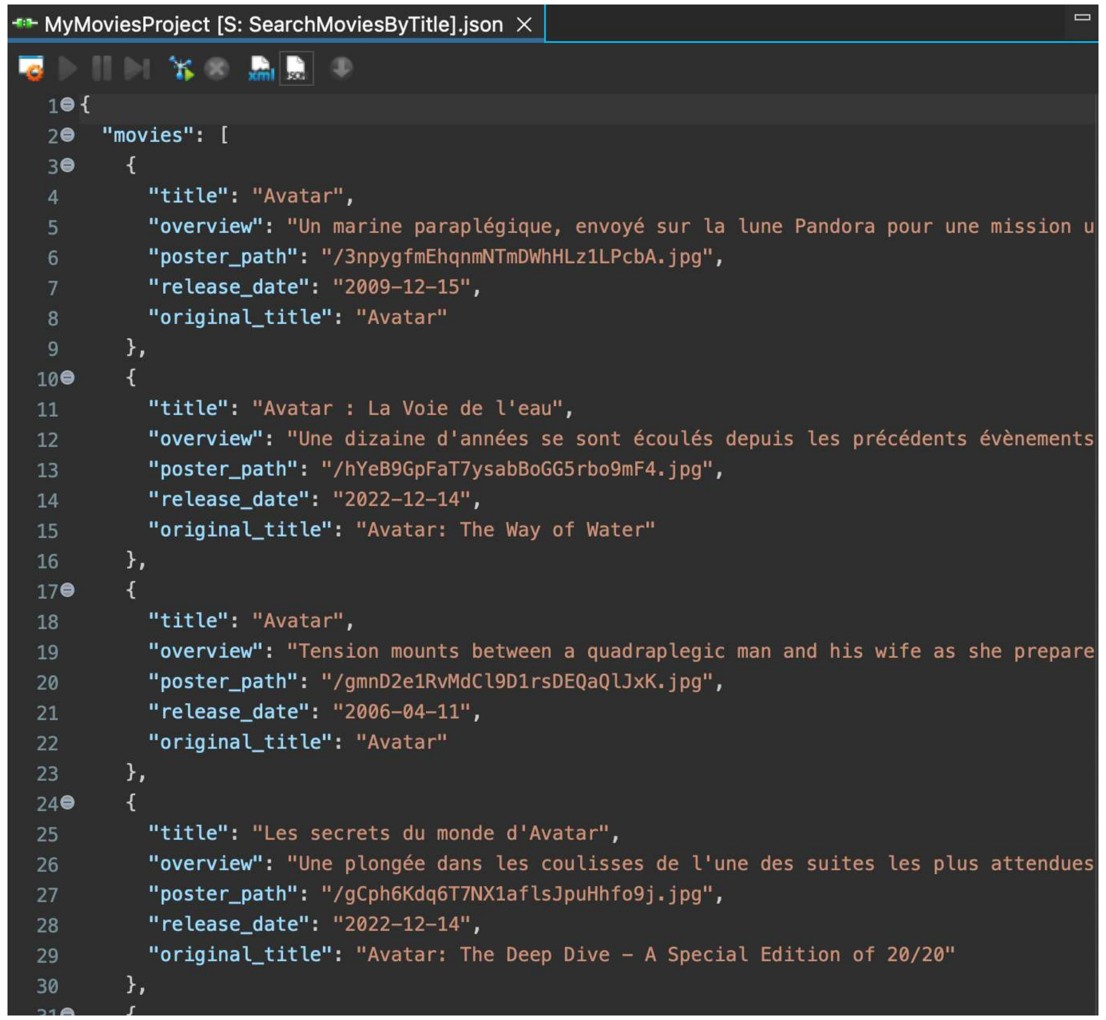
Right-click on the test case, choose **Run** to execute it and generate response data.





The results are displayed in the editors panel.

The response data generated by the sequence will display only the information you requested.



```
{
  "movies": [
    {
      "title": "Avatar",
      "overview": "Un marine paraplégique, envoyé sur la lune Pandora pour une mission u
      "poster_path": "/3npygfmEhqnmNTmDWhHLz1LPcbA.jpg",
      "release_date": "2009-12-15",
      "original_title": "Avatar"
    },
    {
      "title": "Avatar : La Voie de l'eau",
      "overview": "Une dizaine d'années se sont écoulés depuis les précédents évènements
      "poster_path": "/hYeB9GpFaT7ysabBoGG5rbo9mF4.jpg",
      "release_date": "2022-12-14",
      "original_title": "Avatar: The Way of Water"
    },
    {
      "title": "Avatar",
      "overview": "Tension mounts between a quadraplegic man and his wife as she prepare
      "poster_path": "/gmnD2e1RvMdCl9D1rsDEQaQlJxK.jpg",
      "release_date": "2006-04-11",
      "original_title": "Avatar"
    },
    {
      "title": "Les secrets du monde d'Avatar",
      "overview": "Une plongée dans les coulisses de l'une des suites les plus attendues
      "poster_path": "/gCph6Kdq6T7NX1aflsJpuHhfo9j.jpg",
      "release_date": "2022-12-14",
      "original_title": "Avatar: The Deep Dive - A Special Edition of 20/20"
    },
    {
```

```
"movies": [
  {
    "title": "Avatar",
    "overview": "Un marine paraple
    "poster_path": "/3npygfmEhqnm
    "release_date": "2009-12-15",
    "original_title": "Avatar"
  },
```

5 – JavaScript Scope

How to handle JavaScript in the studio.

6.1 What is the JavaScript Scope ?

6.2 Interactions with JS Scope

6.3 Back-end Objects bound to JS Scope

6.4 Step Sequence JS

6.5 Step Input variables

6.6 Modify a sequence with the JS Scope

5.1 What is the JavaScript Scope ?

By default, **every execution** of a transaction or a sequence has a **JavaScript environment**.
This is called the **JavaScript Scope**.

You can use JS to **manipulate data** in the sequence.
For example, perform calculations and data transformations...

Transaction or sequence input variables

All variables declared as **input vars** (input variables) of the sequence

- are inserted into the global scope of the JS environment.
- are automatically JavaScript variables
- become global variables of the sequence.

5.2 Interactions with JS Scope

In order to **manipulate data in JavaScript**,

Convertigo uses **backend objects** as gateways between the **structured context** and the **JS scope**.

These objects

- manage **interactions** between XML data sources and JavaScript.
- are used as **steps in sequences**.

These objects or steps can either

- **transform XML data** from the source defined in the Source property **into JavaScript variables** in the current executed sequence JS scope. These JS variables can be manipulated in JS.
- **transform JavaScript scope variables** into **XML data sources**.
- use **JavaScript expressions** as **data sources**.

5.3 Back-end Objects bound to JS Scope

Steps transforming XML data sources into JavaScript variables

jSimpleSource

jSimpleSource - JS step

This step **transforms a single node** from the source defined in the Source property into a **JS variable** (String)

jSource

jSource - JS step

This step **transforms a list of XML nodes** into a **JS variable** (Java NodeList object)

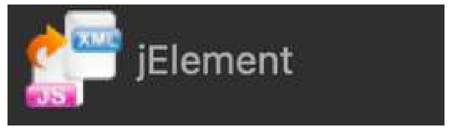
JsonSource

JsonSource - JS step

This step **extracts a JSON typed XML structure** from the source defined in the Source property,
parses it as JSON,
and sets it as a **JS variable** (JS Object or JS Array).

5.3 Back-end Objects bound to JS Scope

Steps transforming JS variables into XML



jElement - XML Step

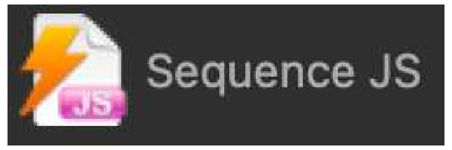
This step **adds an XML element node** based on a **JS expression** to parent XML element in the **sequence XML output**.

JSON to XML - JSON step

This step **adds an XML attribute node** based on a **JS expression** to parent XML element in the **sequence XML output**.

5.3 Back-end Objects bound to JS Scope

Steps used to manipulate JavaScript


Sequence JS

Sequence JS – JS step

This step is used to **write JavaScript code** which is **executed in the sequence scope** (initialize variables, calculations…)

jException

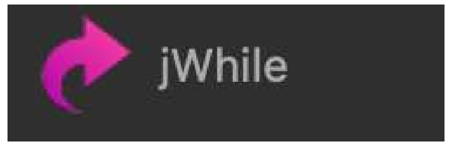
jException – JS step

This step **raises a Convertigo Engine exception**. It **breaks the sequence execution flow**, ending the sequence just after this step.

jIf

jIf – Flow control step

This step is **based on a JavaScript condition** and contains other steps executed only **if the condition is fulfilled**.


jWhile

jWhile – Flow control step

This step **executes a group of child steps** as the **condition expression** set in the Condition property remains true.

5.4 Sequence JS Step

The JS Scope is useful to **modify Sequences**.

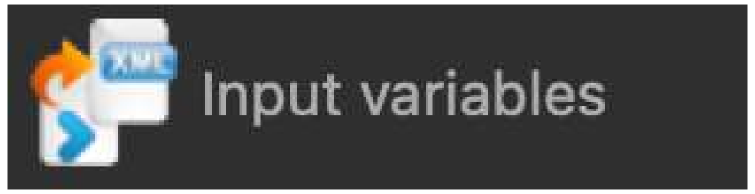
When you need to **write code directly in JavaScript**, the **Sequence JS step** is very helpful.
This JavaScript code will be **executed in the sequence scope**.

Sequence JS

With the Sequence JS step, you can :

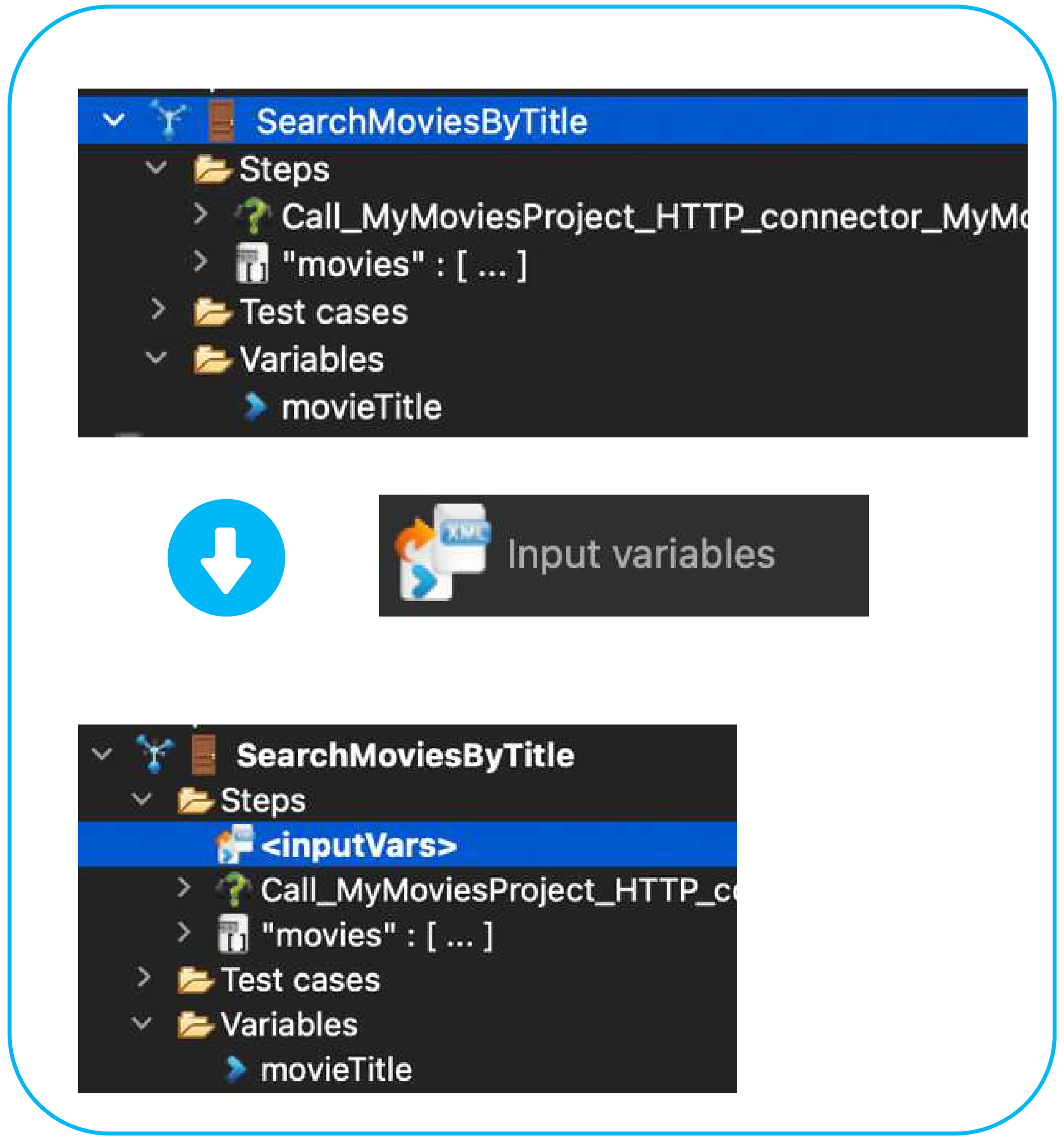
- initialize variables,
- perform complex calculations,
- access the context object to get useful properties (contextID, httpSession, isCacheEnabled, lockPooledContext, etc.)
- use some context methods to manipulate the result XML DOM, encode and decode data, abort sequence...

5.5 Input variables Step



The **step Input variables** is an XML element containing dynamically the **input variables of parent Sequence**.

Placed at the **beginning of a Sequence**, this step allows **steps ordered after** to **use the Sequence input variables as source**.

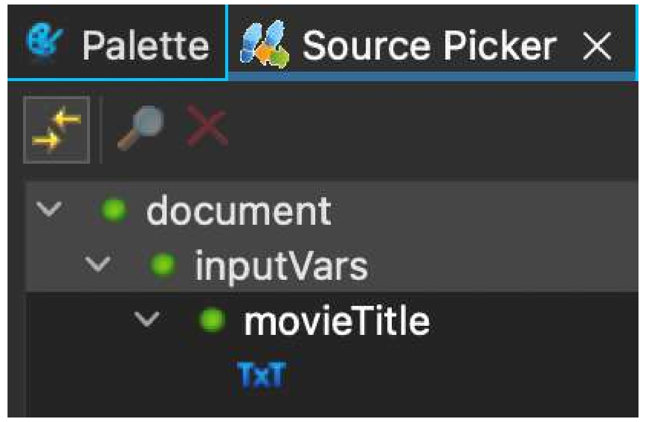


```
SearchMoviesByTitle
  Steps
    Call_MyMoviesProject_HTTP_connector_MyMo
    "movies" : [ ... ]
  Test cases
  Variables
    movieTitle
```

```
SearchMoviesByTitle
  Steps
    <inputVars>
    Call_MyMoviesProject_HTTP_c
    "movies" : [ ... ]
  Test cases
  Variables
    movieTitle
```

When you add it as the first step of the sequence,

it **appears as a source** in the **source picker**.

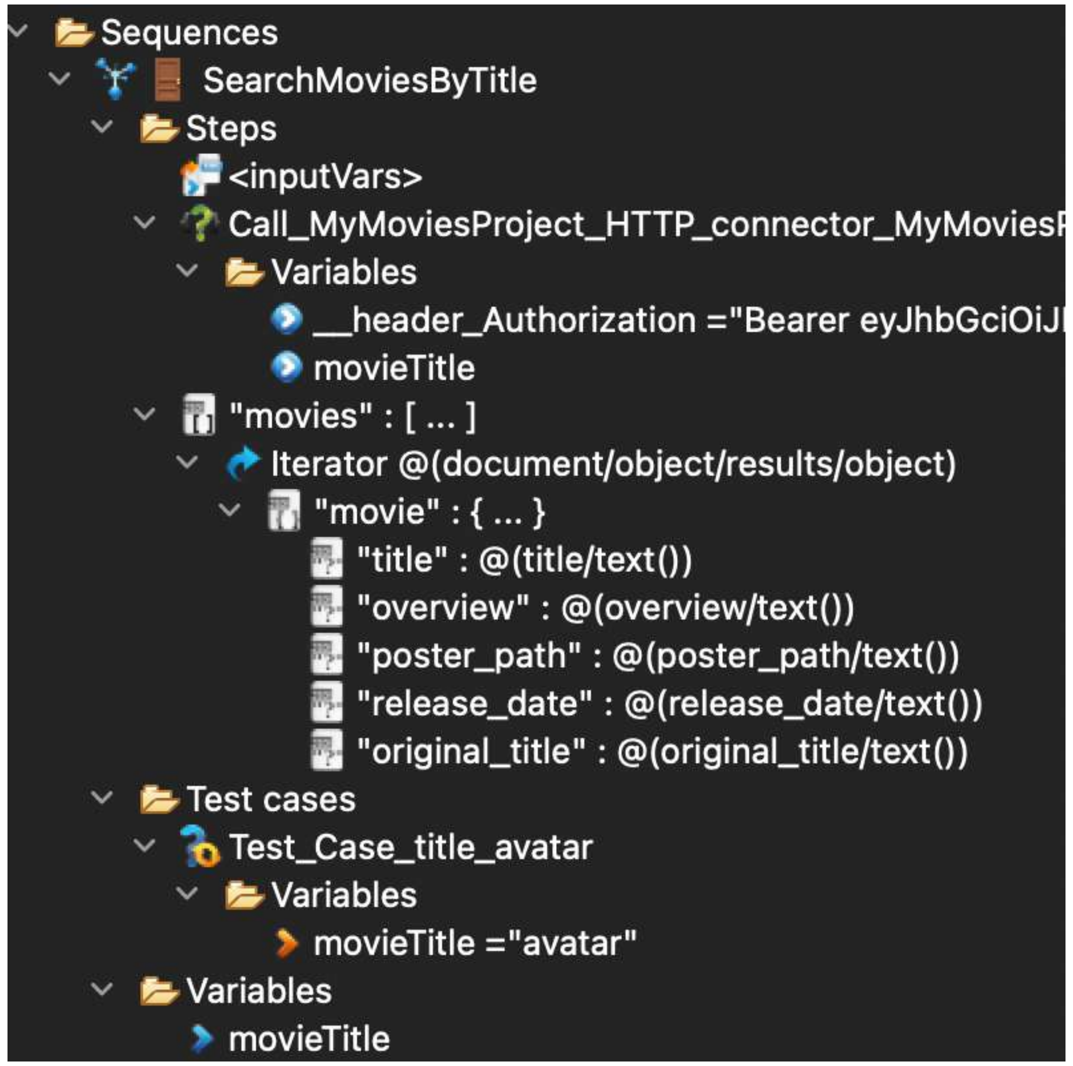


```
document
  inputVars
    movieTitle
      TxT
```

5.6 Modify a sequence with the JS Scope

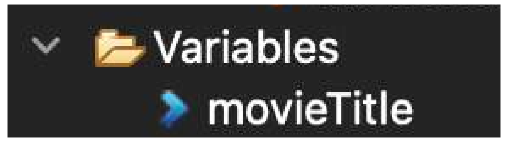
Exercice 1 : Change a variable name with Sequence JS

Here is our sequence SearchMoviesByTitle



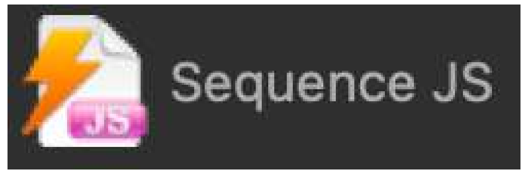
```
Sequences
  SearchMoviesByTitle
    Steps
        <inputVars>
      Call_MyMoviesProject_HTTP_connector_MyMoviesF
        Variables
            __header_Authorization ="Bearer eyJhbGciOiJI
            movieTitle
      "movies" : [ ... ]
        Iterator @(document/object/results/object)
          "movie" : { ... }
              "title" : @(title/text())
              "overview" : @(overview/text())
              "poster_path" : @(poster_path/text())
              "release_date" : @(release_date/text())
              "original_title" : @(original_title/text())
    Test cases
      Test_Case_title_avatar
        Variables
            movieTitle ="avatar"
    Variables
        movieTitle
```

Let's say we want the name of the **input variable** "**movieTitle**" to appear as **title** in our sequence.



```
Variables
  movieTitle
```

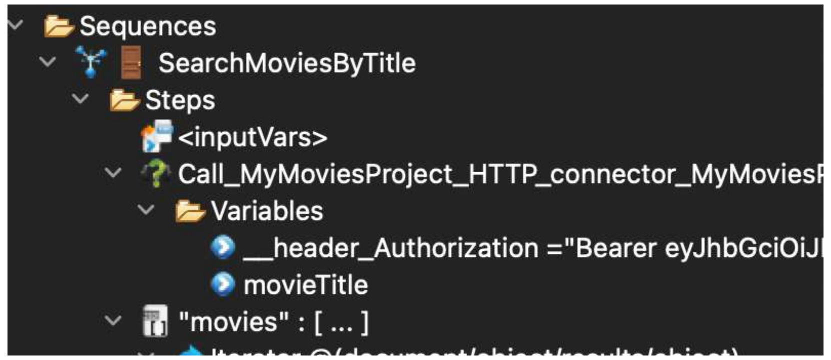
To change the name of the input variable "movieTitle", we are going to use JavaScript in a **Sequence JS**.



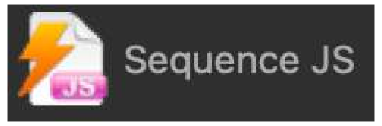
5.6 Modify a sequence with the JS Scope

Exercice 1 : Change a variable name with Sequence JS

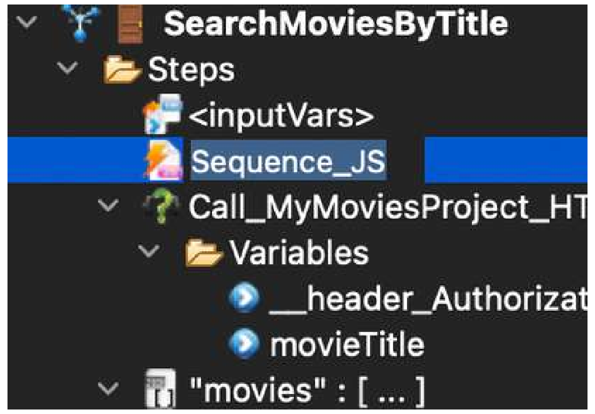
Drag the **Sequence JS step** from the palette in the steps folder after the step InputVars.



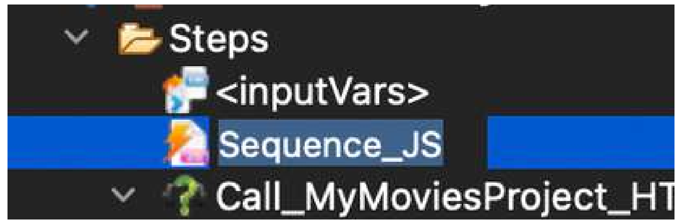
```
Sequences
  SearchMoviesByTitle
    Steps
      <inputVars>
      Call_MyMoviesProject_HTTP_connector_MyMoviesP
        Variables
          __header_Authorization ="Bearer eyJhbGciOiJI
          movieTitle
      "movies" : [ ... ]
```



Sequence JS

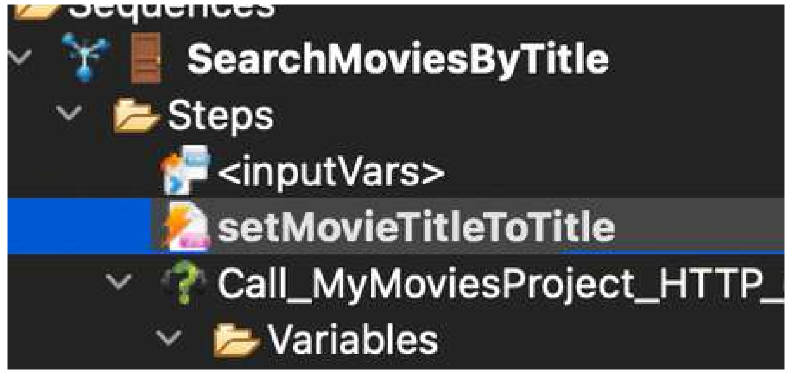


```
SearchMoviesByTitle
  Steps
    <inputVars>
    Sequence_JS
    Call_MyMoviesProject_HT
      Variables
        __header_Authorizat
        movieTitle
    "movies" : [ ... ]
```



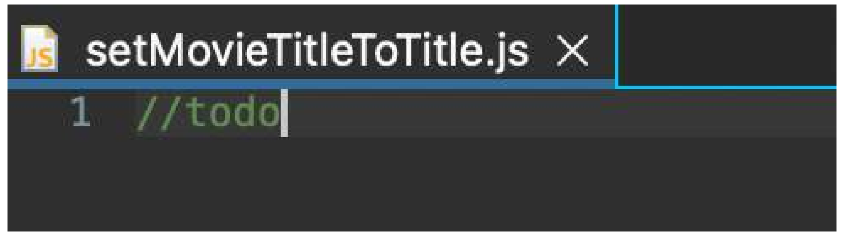
```
Steps
  <inputVars>
  Sequence_JS
  Call_MyMoviesProject_HT
```

Rename it setMovieTitleToTitle.



```
Sequences
  SearchMoviesByTitle
    Steps
      <inputVars>
      setMovieTitleToTitle
      Call_MyMoviesProject_HTTP_
        Variables
```

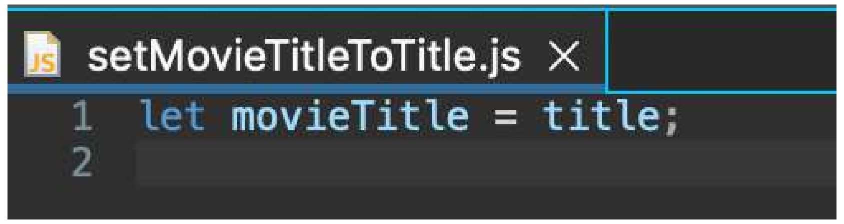
Click twice on **setMovieTitleToTitle** to open the **file setMovieTitleToTitle.js** in the editor panel



setMovieTitleToTitle.js

```
//todo
```

Change the variable name in the file with JavaScript.

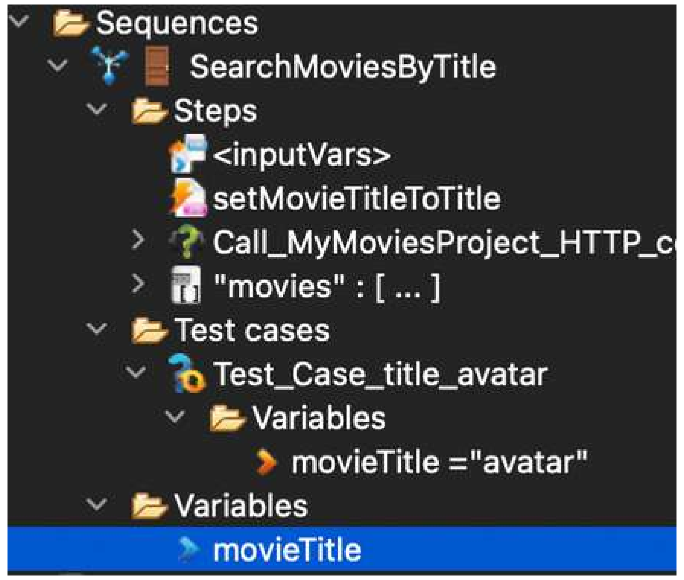


setMovieTitleToTitle.js

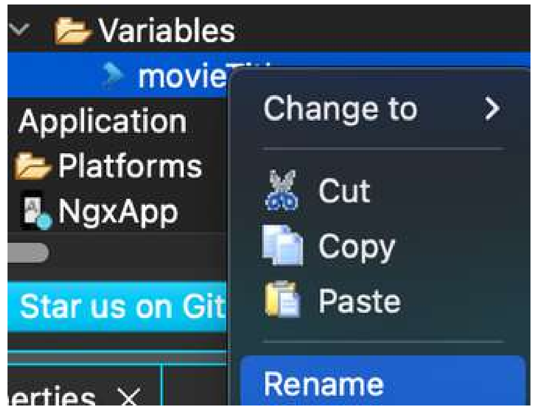
```
let movieTitle = title;
```

5.6 Modify a sequence with the JS Scope

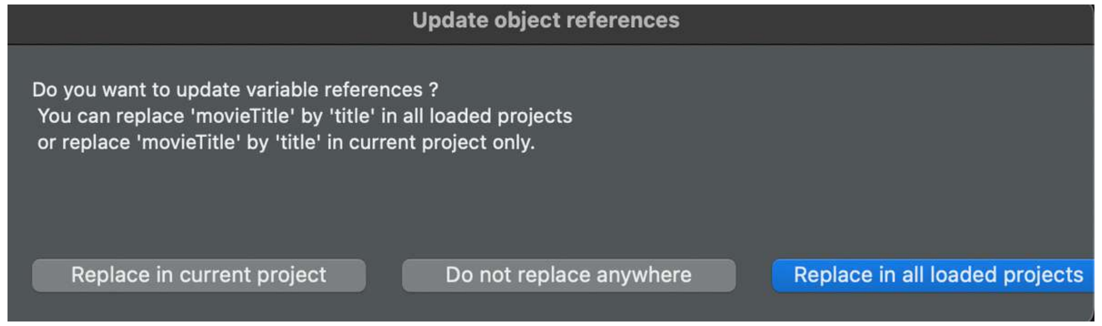
Exercice 1 : Change a variable name with Sequence JS



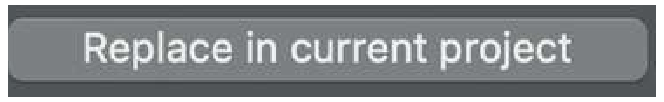
Rename the **movieTitle variable** to **title** in the **Variables folder** of the sequence.



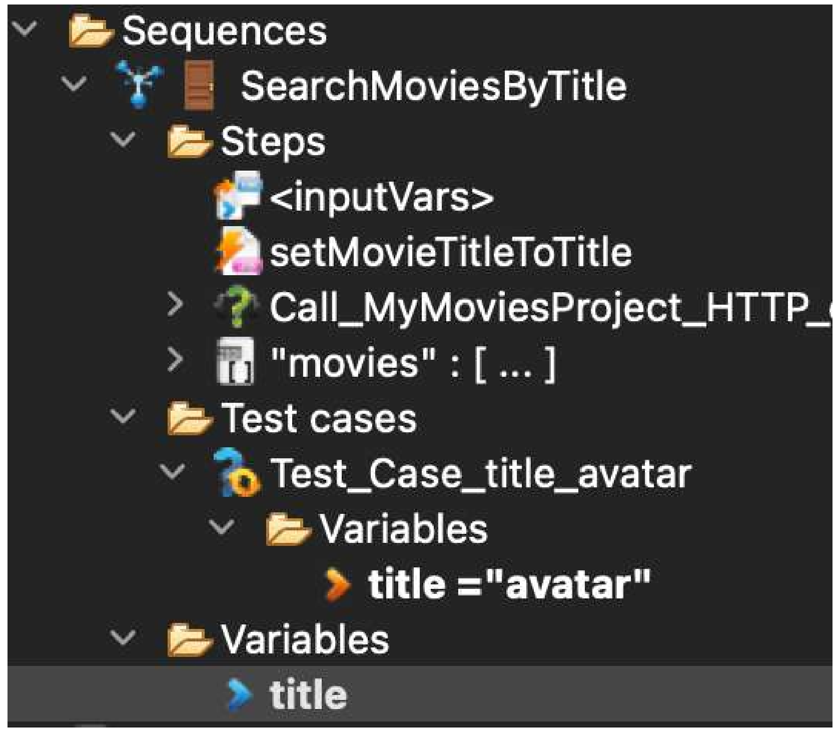
In the **Update object references window**, select **Replace in current project**.

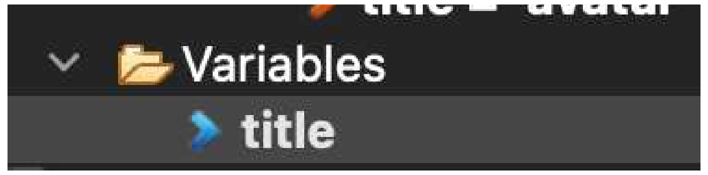


Do you want to update variable references ?
You can replace 'movieTitle' by 'title' in all loaded projects
or replace 'movieTitle' by 'title' in current project only.



The variable **appears as title** in the sequence (Test cases, Variables folder)





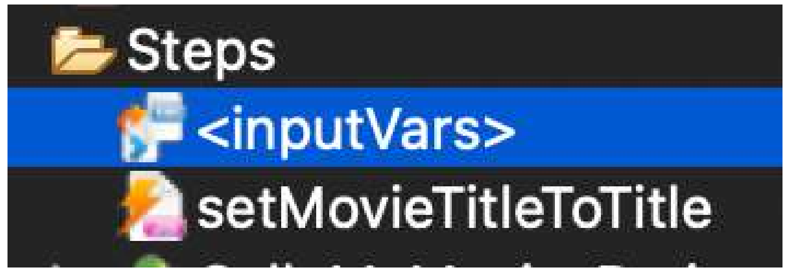
5.6 Modify a sequence with the JS Scope

Exercice 1 : Change a variable name with Sequence JS

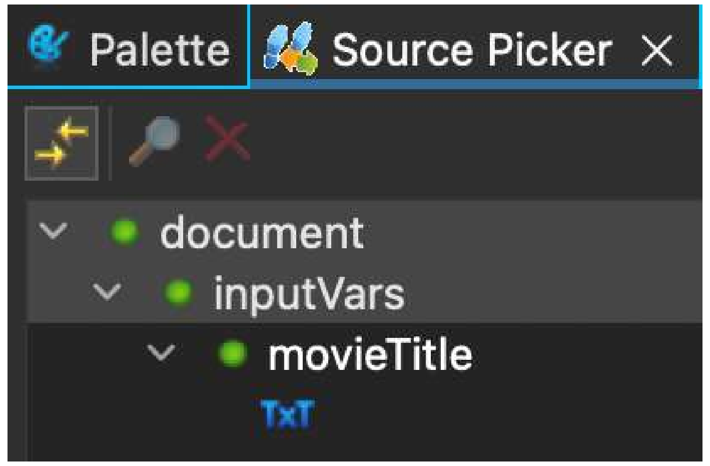
When focused on **inputVars step**,

the source picker shows the entry variable of the sequence as **title**

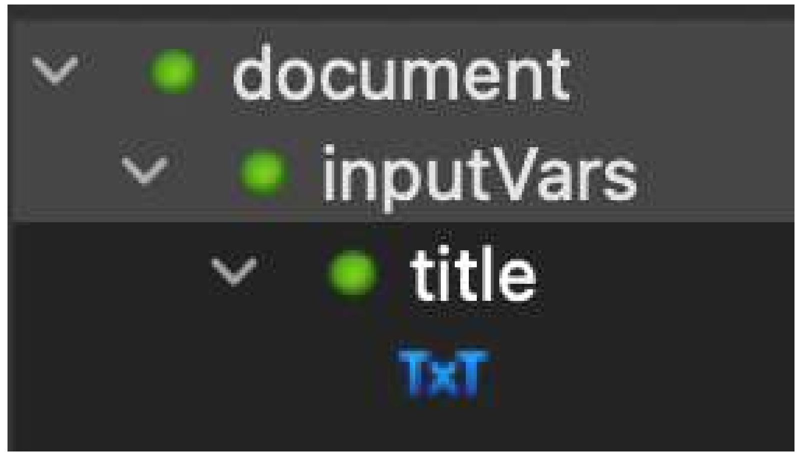
(not as movieTitle anymore).



Before



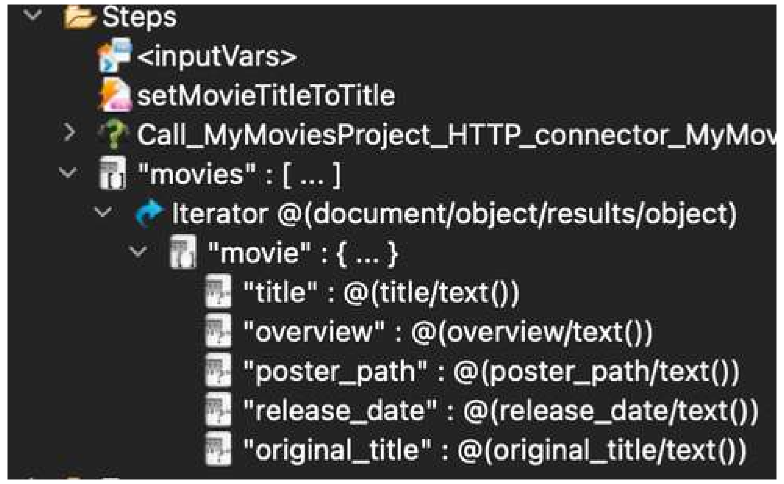
Now



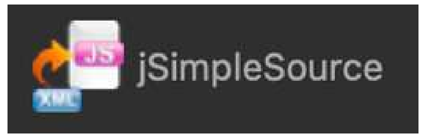
5.6 Modify a sequence with the JS Scope

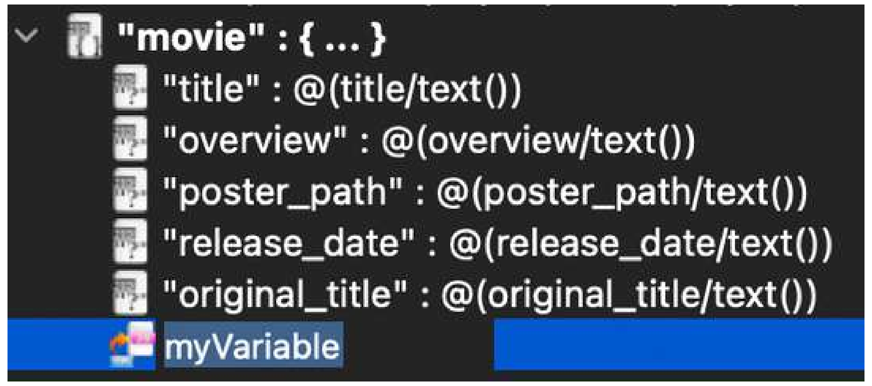
Exercice 2 : Set the title to uppercase with Sequence JS

Let’s say we want a field with the title in uppercase in our JSON data.



```
Steps
  <inputVars>
  setMovieTitleToTitle
> Call_MyMoviesProject_HTTP_connector_MyMov
  "movies" : [ ... ]
    Iterator @(document/object/results/object)
      "movie" : { ... }
        "title" : @(title/text())
        "overview" : @(overview/text())
        "poster_path" : @(poster_path/text())
        "release_date" : @(release_date/text())
        "original_title" : @(original_title/text())
```





```
"movie" : { ... }
  "title" : @(title/text())
  "overview" : @(overview/text())
  "poster_path" : @(poster_path/text())
  "release_date" : @(release_date/text())
  "original_title" : @(original_title/text())
  myVariable
```

In the movie object,

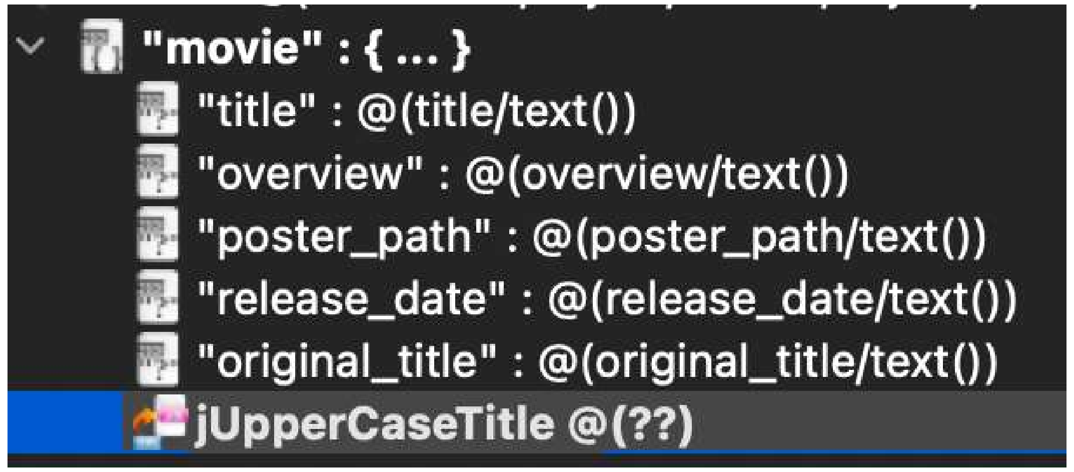
after the field steps,

we **add a jSimpleSource step**..

We name it **jUpperCaseTitle**

(for JS variables, good practice is

to add a “j” at the beginning)



```
"movie" : { ... }
  "title" : @(title/text())
  "overview" : @(overview/text())
  "poster_path" : @(poster_path/text())
  "release_date" : @(release_date/text())
  "original_title" : @(original_title/text())
  jUpperCaseTitle @(??)
```

5.6 Modify a sequence with the JS Scope

Exercice 2 : Set the title to uppercase with Sequence JS

The jSimpleSource step is used

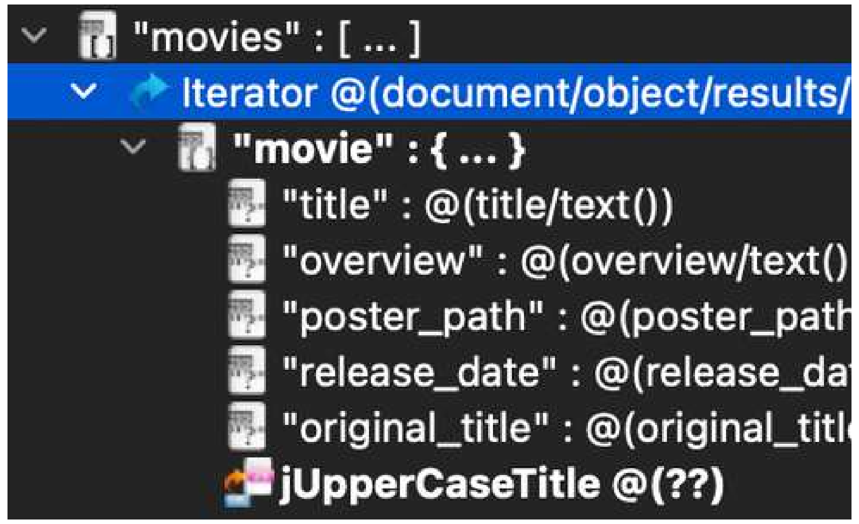
to **transform a single node** from a source **into a JS variable.**

Now we want to **bind it to the value of the field title in the iterator**.

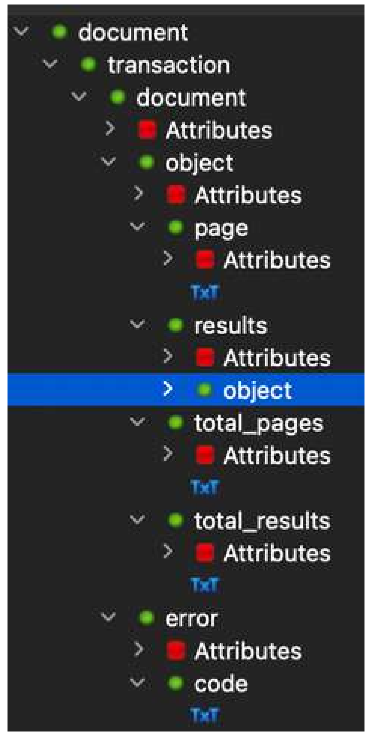
Double-click on the **Iterator step**

to display its data in the Source Picker

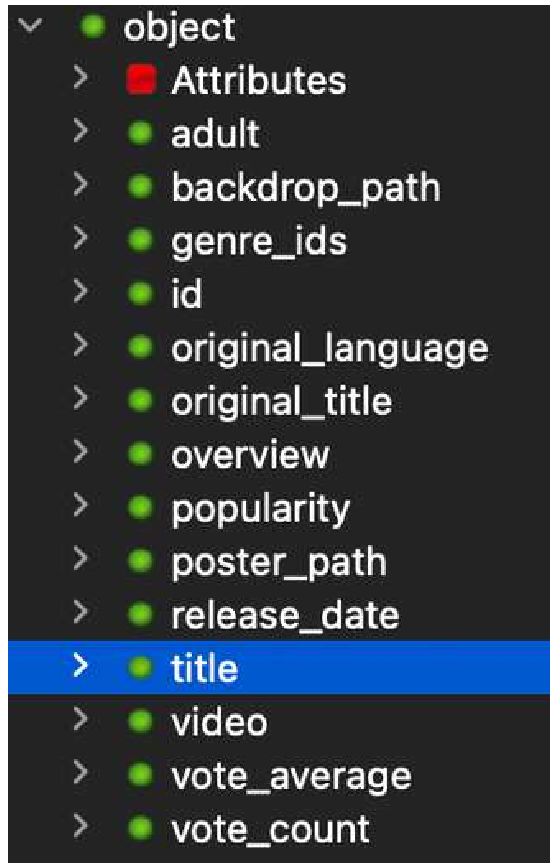
and open the object node.



```
"movies" : [ ... ]
  Iterator @(document/object/results/
    "movie" : { ... }
      "title" : @(title/text())
      "overview" : @(overview/text()
      "poster_path" : @(poster_path
      "release_date" : @(release_da
      "original_title" : @(original_titl
      jUpperCaseTitle @(??)
```



```
document
  transaction
    document
      Attributes
      object
        Attributes
        page
          Attributes
          TxT
        results
          Attributes
          object
        total_pages
          Attributes
          TxT
        total_results
          Attributes
          TxT
      error
        Attributes
        code
          TxT
```

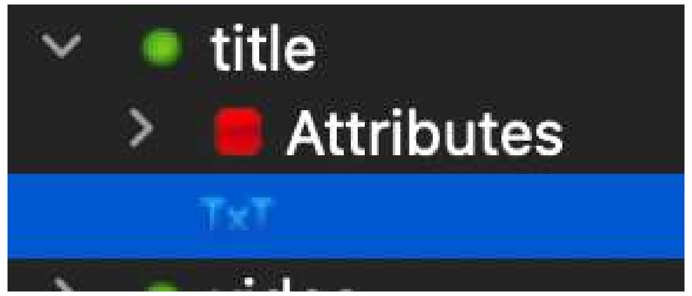


```
object
  Attributes
  adult
  backdrop_path
  genre_ids
  id
  original_language
  original_title
  overview
  popularity
  poster_path
  release_date
  title
  video
  vote_average
  vote_count
```

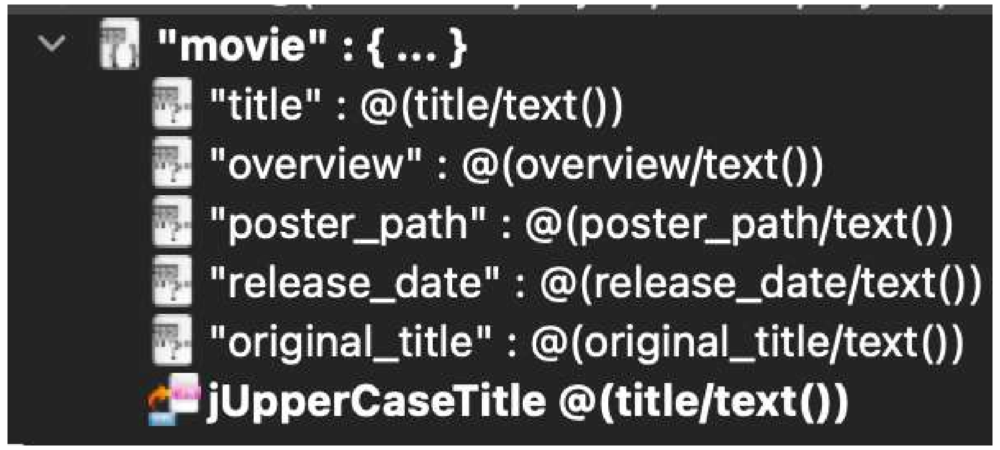
Drag and drop

the **TxT element of the title**

into the jSimpleSource step.



```
title
  Attributes
  TxT
```

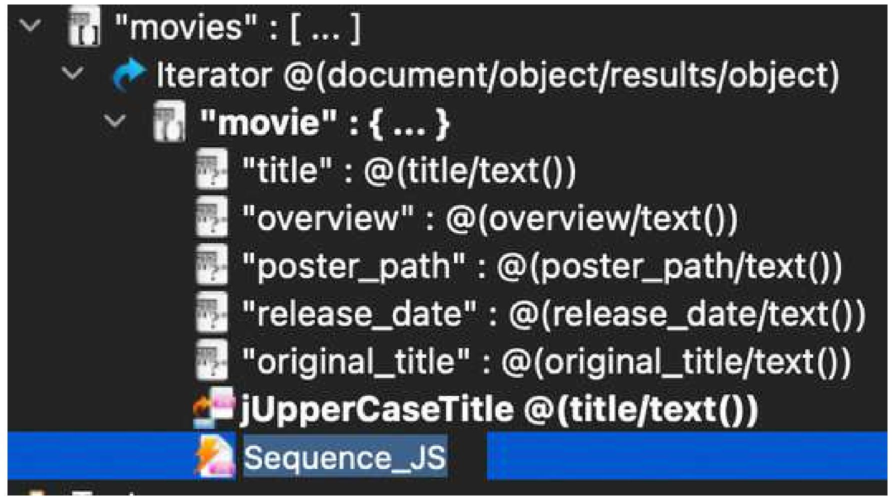


```
"movie" : { ... }
  "title" : @(title/text())
  "overview" : @(overview/text())
  "poster_path" : @(poster_path/text())
  "release_date" : @(release_date/text())
  "original_title" : @(original_title/text())
  jUpperCaseTitle @(title/text())
```

5.6 Modify a sequence with the JS Scope

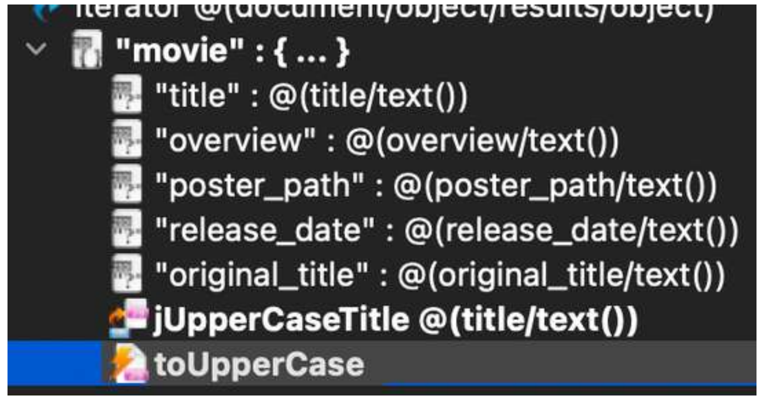
Exercice 2 : Set the title to uppercase with Sequence JS

Add a Sequence JS step
in the sequence,
after the step jUpperCaseTitle.

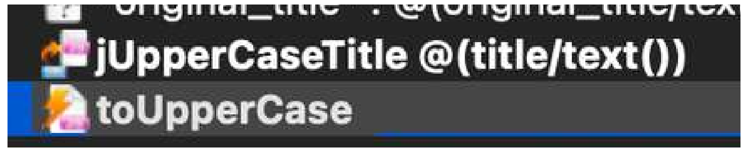


```
"movies" : [ ... ]
  Iterator @(document/object/results/object)
    "movie" : { ... }
      "title" : @(title/text())
      "overview" : @(overview/text())
      "poster_path" : @(poster_path/text())
      "release_date" : @(release_date/text())
      "original_title" : @(original_title/text())
      jUpperCaseTitle @(title/text())
      Sequence_JS
```

Rename it toUpperCase.

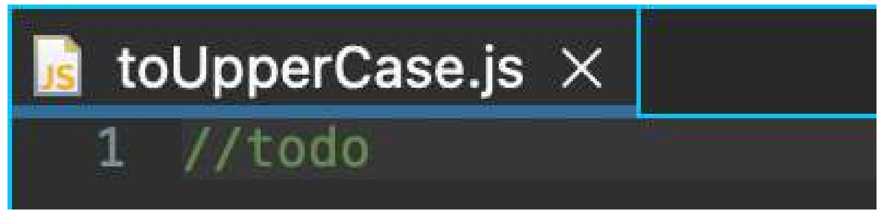


```
Iterator @(document/object/results/object)
  "movie" : { ... }
    "title" : @(title/text())
    "overview" : @(overview/text())
    "poster_path" : @(poster_path/text())
    "release_date" : @(release_date/text())
    "original_title" : @(original_title/text())
    jUpperCaseTitle @(title/text())
    toUpperCase
```



```
"original_title" : @(original_title/tex
jUpperCaseTitle @(title/text())
toUpperCase
```

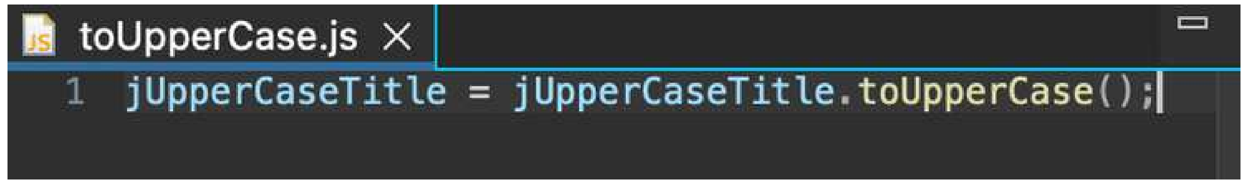
Click twice on the step
to open toUpperCase.js



toUpperCase.js ×

```
//todo
```

Edit the toUpperCase.js
with JS code.



toUpperCase.js ×

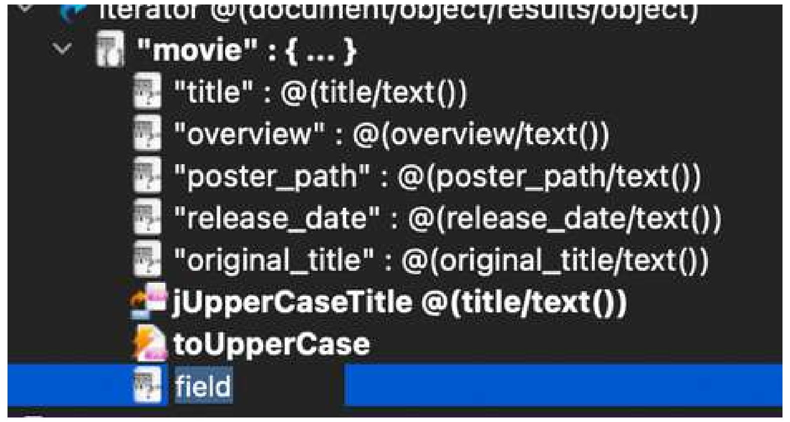
```
jUpperCaseTitle = jUpperCaseTitle.toUpperCase();
```

5.6 Modify a sequence with the JS Scope

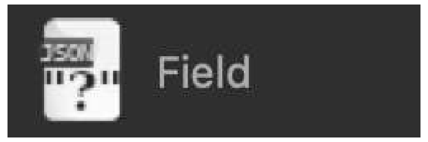
convertigo

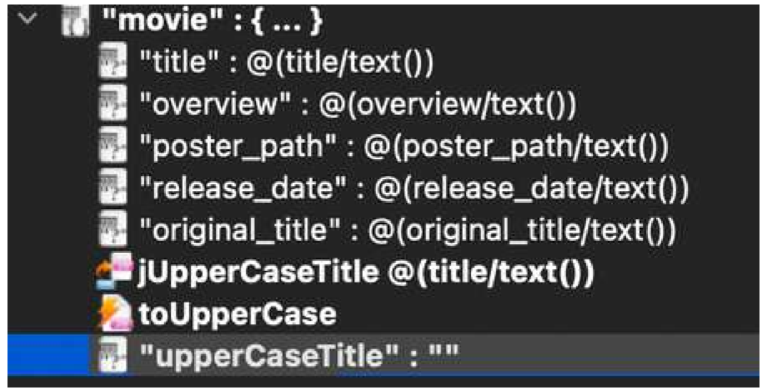
Exercice 2 : Set the title to uppercase with Sequence JS

Add a field step after toUpperCase and name it **upperCaseTitle**.



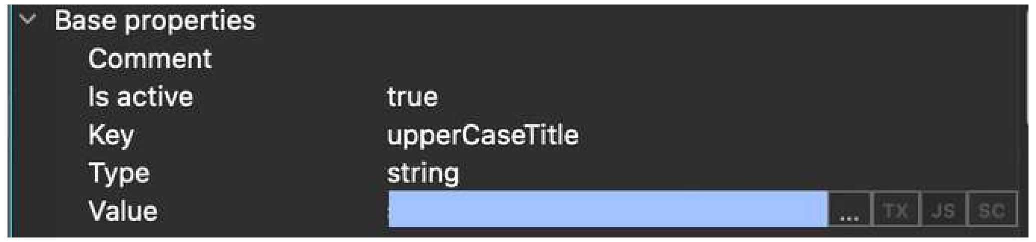
```
iterator @(document/object/results/object)
  "movie" : { ... }
    "title" : @(title/text())
    "overview" : @(overview/text())
    "poster_path" : @(poster_path/text())
    "release_date" : @(release_date/text())
    "original_title" : @(original_title/text())
    jUpperCaseTitle @(title/text())
    toUpperCase
    field
```



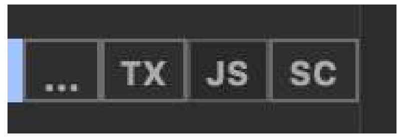


```
  "movie" : { ... }
    "title" : @(title/text())
    "overview" : @(overview/text())
    "poster_path" : @(poster_path/text())
    "release_date" : @(release_date/text())
    "original_title" : @(original_title/text())
    jUpperCaseTitle @(title/text())
    toUpperCase
    "upperCaseTitle" : ""
```

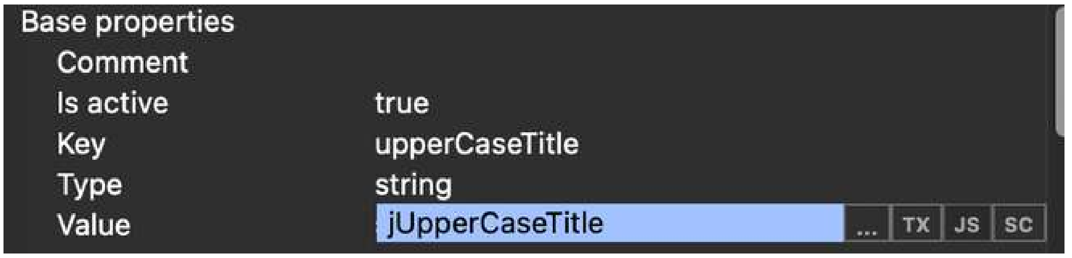
In the **Value property of upperCaseTitle**, select the **JS Scope** by clicking on JS.



Base properties	
Comment	
Is active	true
Key	upperCaseTitle
Type	string
Value	

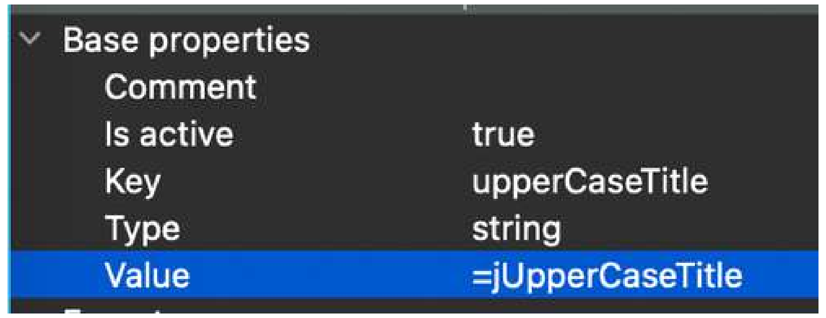


Enter **jUpperCaseTitle** to select the JS variable as value.

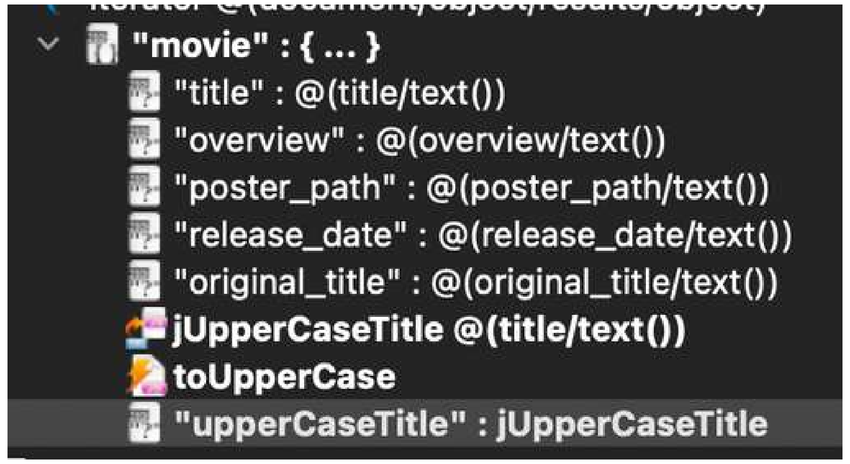


Base properties	
Comment	
Is active	true
Key	upperCaseTitle
Type	string
Value	jUpperCaseTitle

The value of upperCaseTitle is now sourced on the value of the JS variable jUpperCaseTitle.



Base properties	
Comment	
Is active	true
Key	upperCaseTitle
Type	string
Value	=jUpperCaseTitle

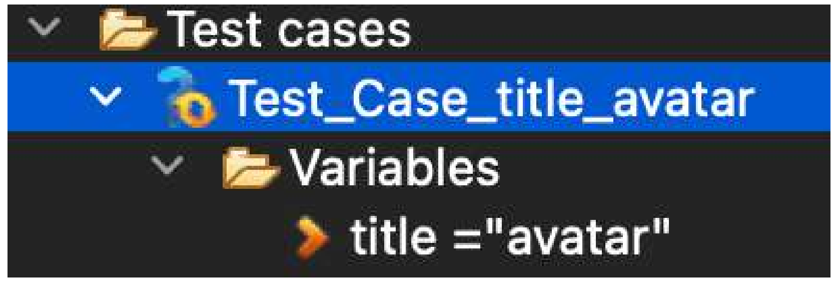


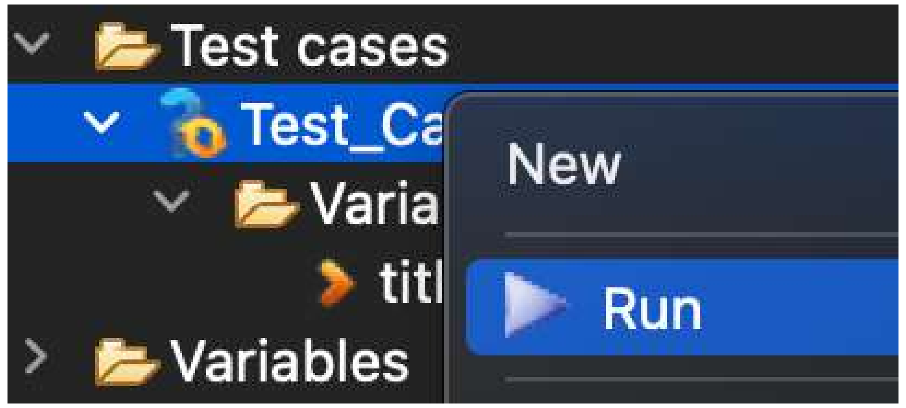
```
iterator @(document/object/results/object)
  "movie" : { ... }
    "title" : @(title/text())
    "overview" : @(overview/text())
    "poster_path" : @(poster_path/text())
    "release_date" : @(release_date/text())
    "original_title" : @(original_title/text())
    jUpperCaseTitle @(title/text())
    toUpperCase
    "upperCaseTitle" : jUpperCaseTitle
```

5.6 Modify a sequence with the JS Scope

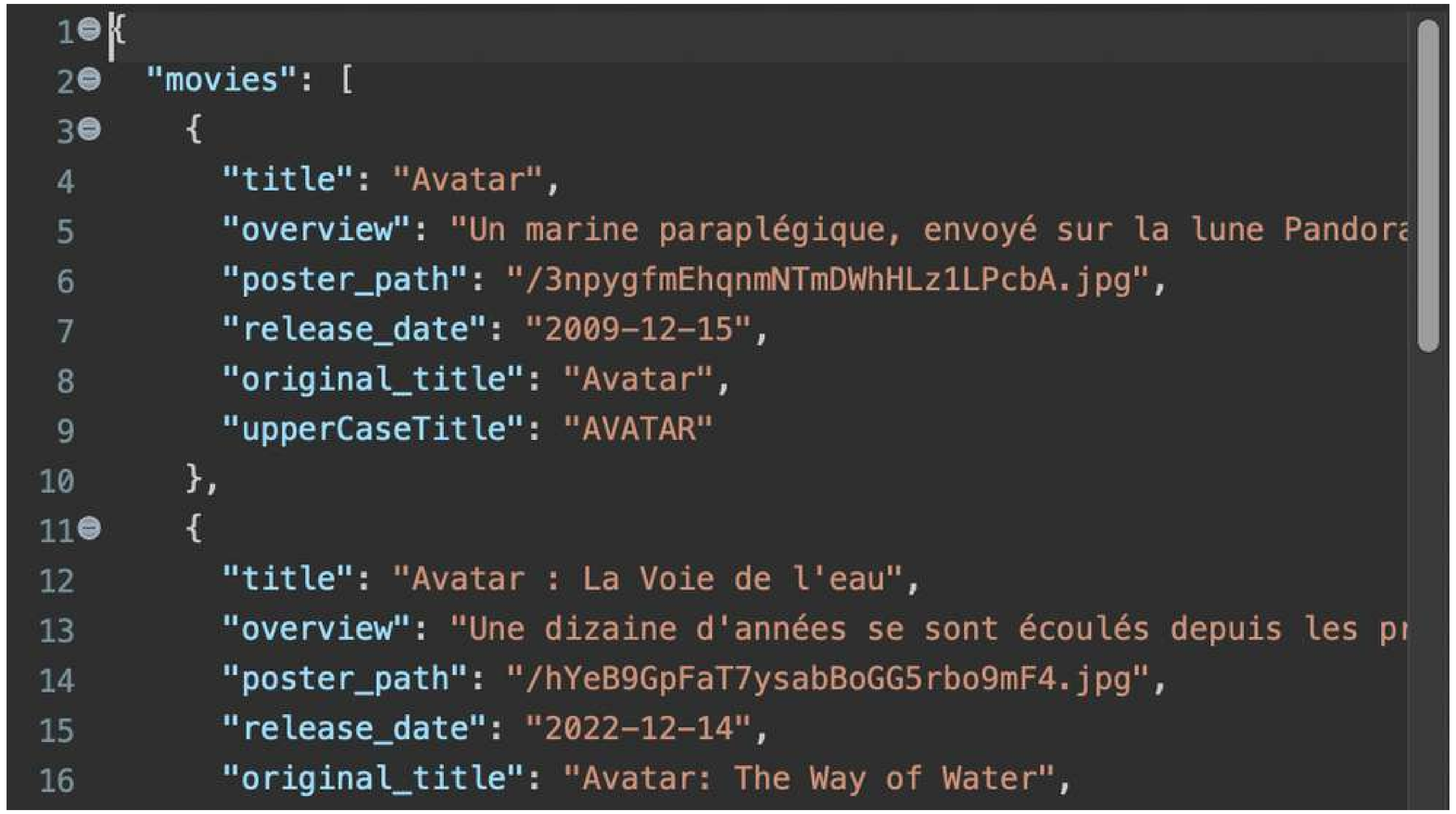
Exercice 2 : Set the title to uppercase with Sequence JS

Let’s run the Test Case





The key upperCaseTitle and the value in upperCase appears in the response data

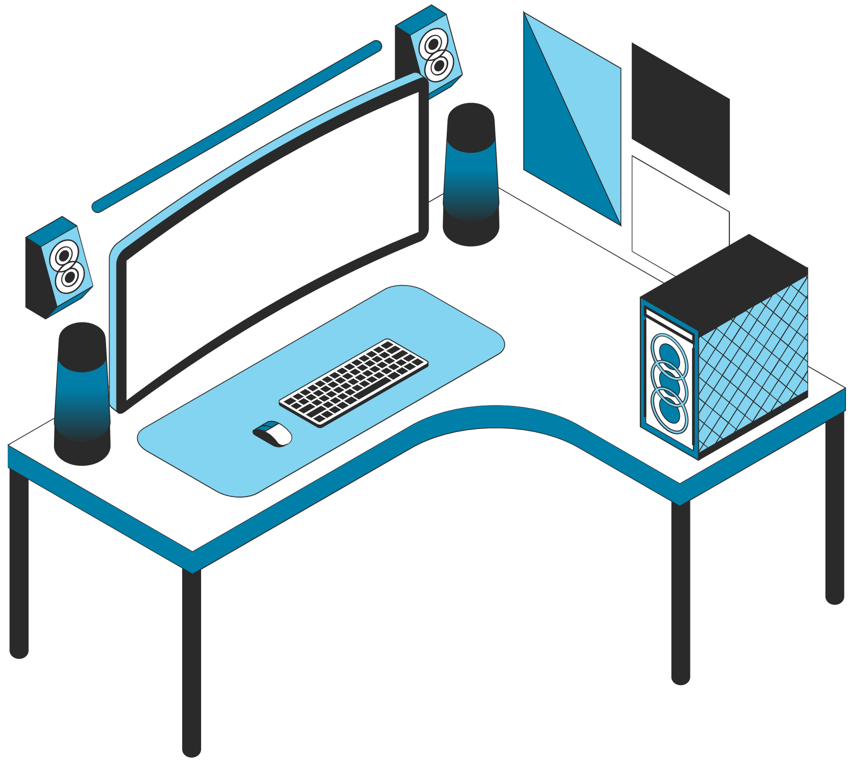


```
{
  "movies": [
    {
      "title": "Avatar",
      "overview": "Un marine paraplégique, envoyé sur la lune Pandora
      "poster_path": "/3npygfmEhqnmNTmDWhHLz1LPcbA.jpg",
      "release_date": "2009-12-15",
      "original_title": "Avatar",
      "upperCaseTitle": "AVATAR"
    },
    {
      "title": "Avatar : La Voie de l'eau",
      "overview": "Une dizaine d'années se sont écoulés depuis les pr
      "poster_path": "/hYeB9GpFaT7ysabBoGG5rbo9mF4.jpg",
      "release_date": "2022-12-14",
      "original_title": "Avatar: The Way of Water",
```

convertigo

6 – Error Management

How to handle errors in the studio.



6.1 Basics on Error Management

6.2 Error Management steps

6.3 Error node & error tag

6.4 Using the IfExist step

6.5 Using the Error Structure step

6.6 Using the Return step

6.1 Basics on Error Management

During its execution, a step can fail, and an error happens.

There are two types of errors:

- Functional errors
- System errors

In Convertigo, to handle errors:

- We don't start with "If everything is OK, then...Or else..."
  => otherwise, there would be too much depth in the tree structure.
- We start with "If there is a problem, then...Or else...,"
  => It means we begin error handling before dealing with successful execution.

After each transaction call in a sequence, we test for errors.

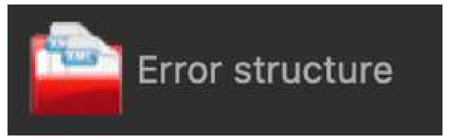
6.2 Error Management steps

Convertigo provides steps to handle errors in sequences.

IfExist

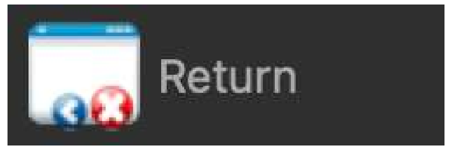
IfExist – Flow control step

This step is used to define an **IF condition looking for node(s) on a source**.
It contains **other steps executed only if** the **source** defined through the Source property **exists**.


Error structure

Error structure – XML step

This step is used to **generate an output XML structure** corresponding to an **applicative error**.
It **doesn't break** the **sequence execution flow**.

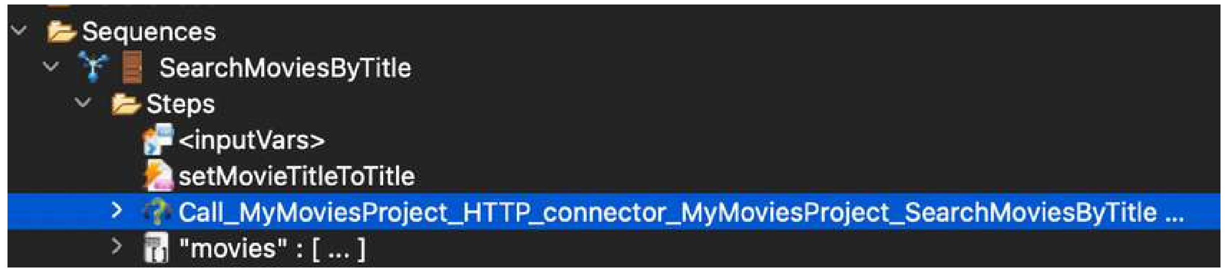

Return

Return – Flow control step

This step is used to **exit the current sequence** in which it is positioned.

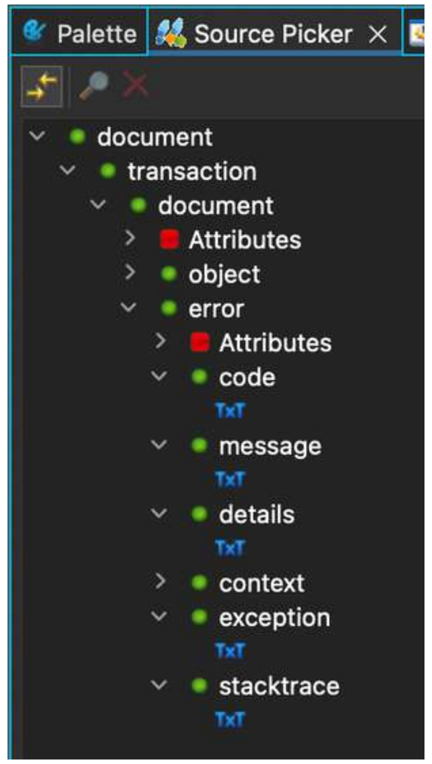
6.3 Error node & error tag

Let's have a look on the XML structure
of the transaction call
in the sequence SearchMoviesByTitle.

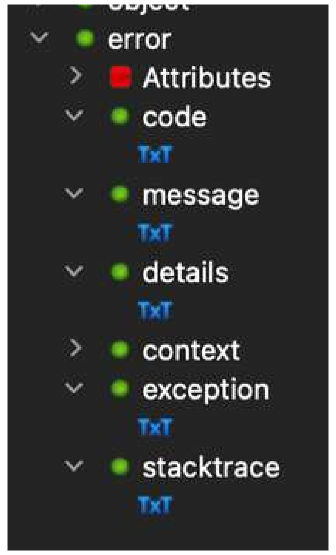


```
Sequences
  SearchMoviesByTitle
    Steps
      <inputVars>
      setMovieTitleToTitle
    > Call_MyMoviesProject_HTTP_connector_MyMoviesProject_SearchMoviesByTitle ...
    > "movies" : [ ... ]
```

Double click on the transaction
to display its structure
in the source picker

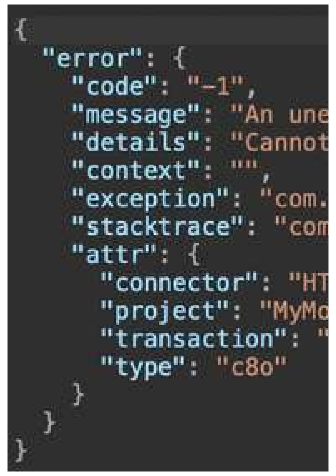


```
document
  transaction
    document
      > Attributes
      > object
      error
        > Attributes
        code
          TxT
        message
          TxT
        details
          TxT
        > context
        exception
          TxT
        stacktrace
          TxT
```



```
object
error
  > Attributes
  code
    TxT
  message
    TxT
  details
    TxT
  > context
  exception
    TxT
  stacktrace
    TxT
```

In the **XML structure of a source**,
there is always an **error node**,
so that **errors can be picked or sourced**
if they are present.



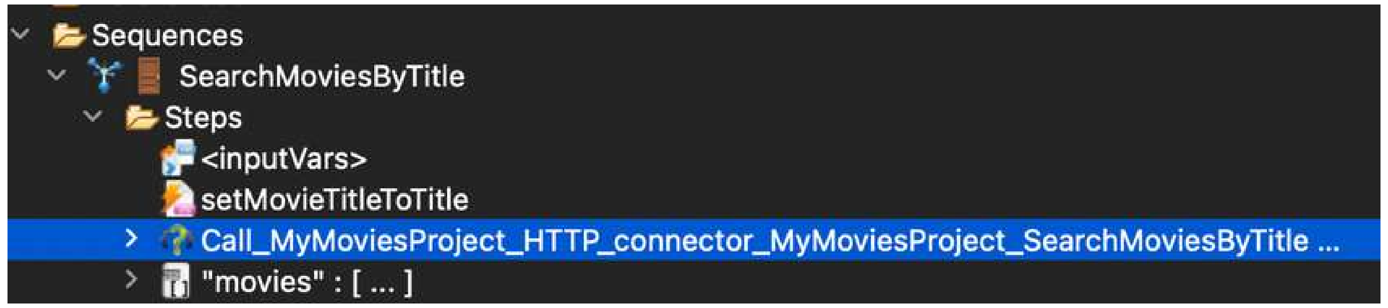
```
{
  "error": {
    "code": "-1",
    "message": "An une
    "details": "Cannot
    "context": "",
    "exception": "com.
    "stacktrace": "com
    "attr": {
      "connector": "HT
      "project": "MyMo
      "transaction": "
      "type": "c8o"
    }
  }
}
```

When a system error
or a functional error happens,
it **generates an error tag**
which "fills" the error node.

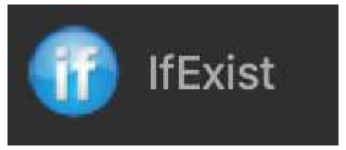
The **error tag structure** is **standardized**.
The error will always be in the same place
and have the same format in the sequence.

6.4 Using the IfExist step

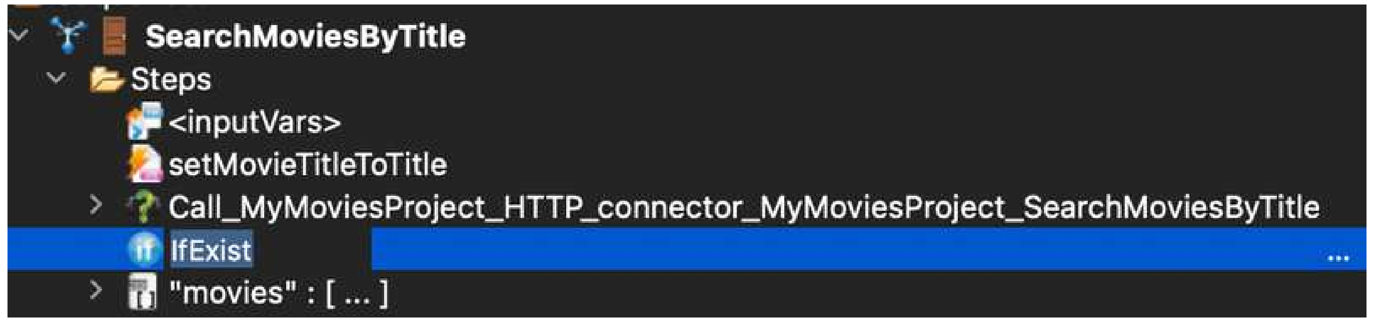
In the **sequence SearchMoviesByTitle**,
just **after the transaction call**
and **before the array movies**,
let's **add an IfExist step**.



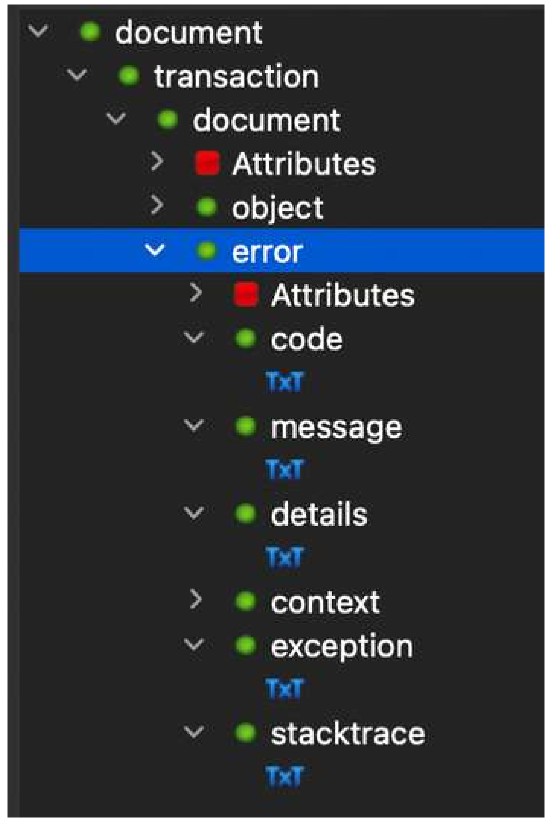
```
Sequences
  SearchMoviesByTitle
    Steps
      <inputVars>
      setMovieTitleToTitle
    > Call_MyMoviesProject_HTTP_connector_MyMoviesProject_SearchMoviesByTitle ...
    > "movies" : [ ... ]
```



IfExist

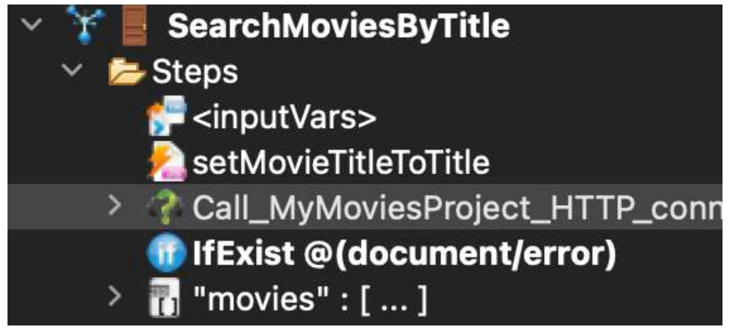


```
SearchMoviesByTitle
  Steps
    <inputVars>
    setMovieTitleToTitle
  > Call_MyMoviesProject_HTTP_connector_MyMoviesProject_SearchMoviesByTitle
    IfExist                                                              ...
  > "movies" : [ ... ]
```

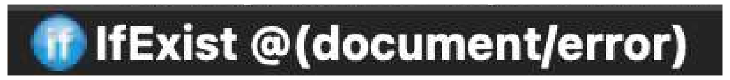


```
document
  transaction
    document
      > Attributes
      > object
        error
          > Attributes
            code
              TxT
            message
              TxT
            details
              TxT
          > context
            exception
              TxT
            stacktrace
              TxT
```

Then **drag the node error**
of the XML structure
of the transaction call
in the step IfExist.



```
SearchMoviesByTitle
  Steps
    <inputVars>
    setMovieTitleToTitle
  > Call_MyMoviesProject_HTTP_conn
    IfExist @(document/error)
  > "movies" : [ ... ]
```

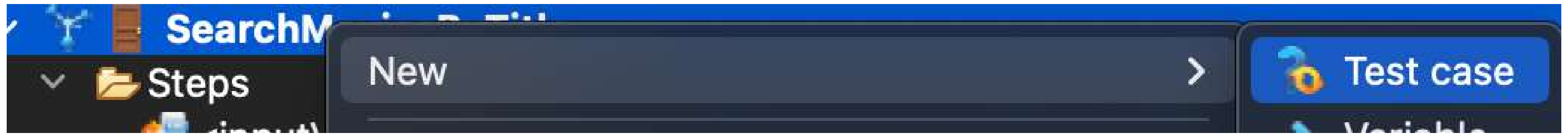


IfExist @(document/error)

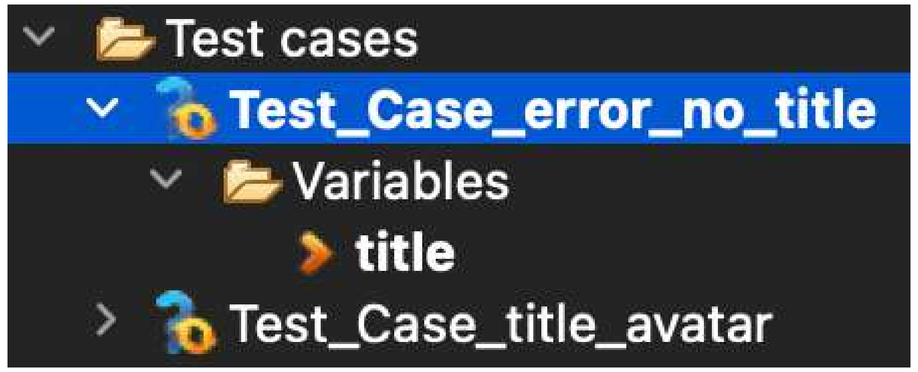
When there is any kind of error,
an **error tag is generated within the error node**.
The IfExists step **checks the XPath of the transaction call**
to see if there is an **error tag in the XPath**.

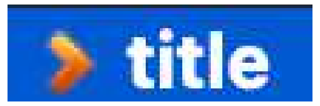
6.4 Using the IfExist step

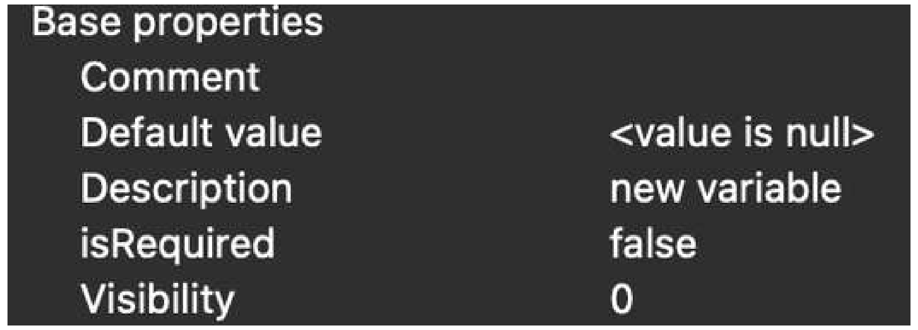
Let’s test it by adding a new test case with an error.



Our test case is created.







Base properties	
Comment	
Default value	<value is null>
Description	new variable
isRequired	false
Visibility	0

In the properties,

the Default value of variable title is null.

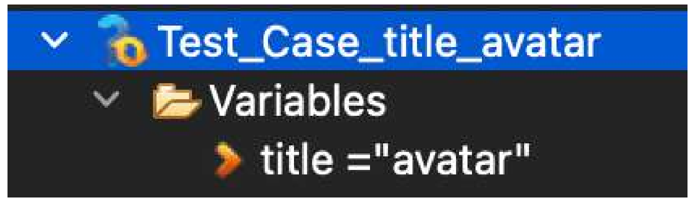
The transaction needs a title to search a movie.

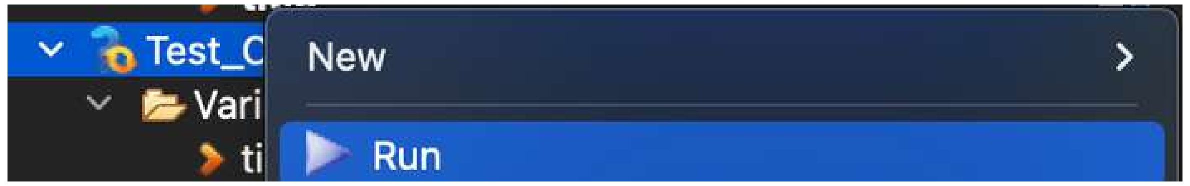
If we forget to add a value to the variable title, an error will be generated.

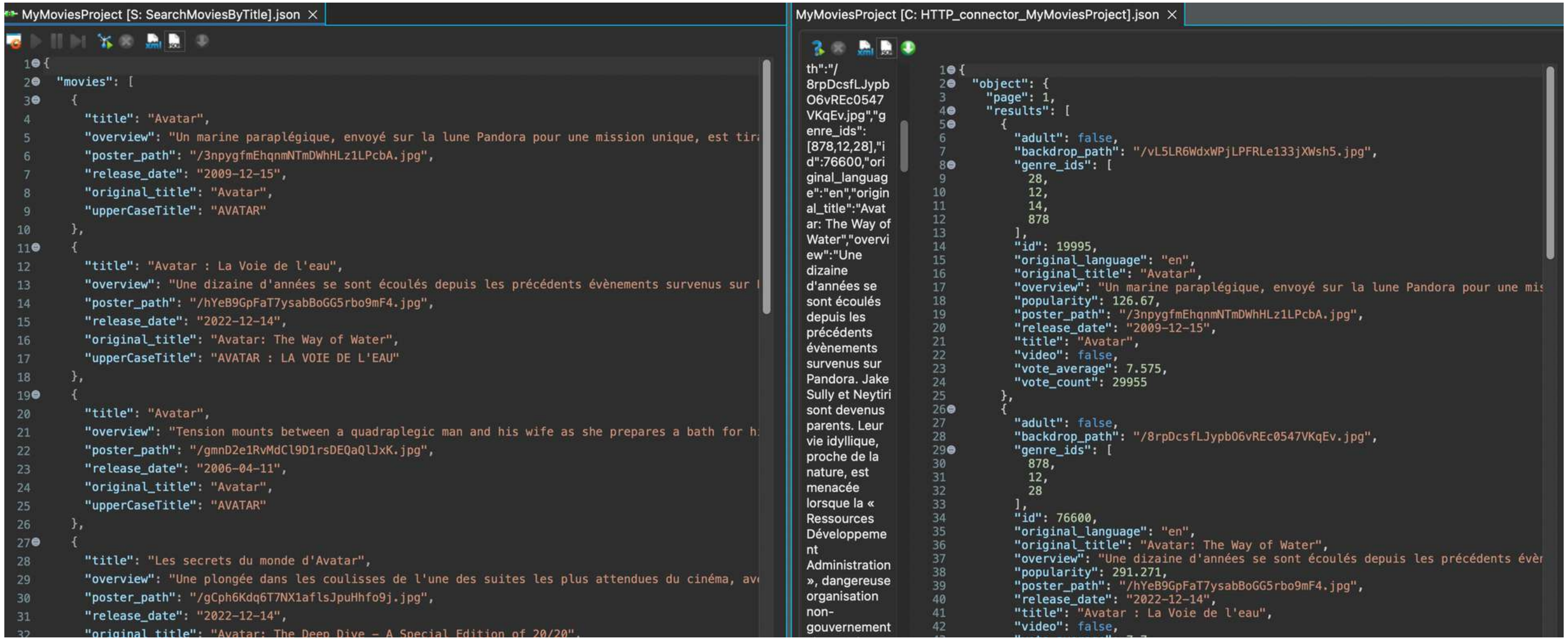
6.4 Using the IfExist step

A a reminder, this is the result of the sequence execution

with a test case where the title has a value and everything is OK.






MyMoviesProject [S: SearchMoviesByTitle].json

```
{
  "movies": [
    {
      "title": "Avatar",
      "overview": "Un marine paraplégique, envoyé sur la lune Pandora pour une mission unique, est tir
      "poster_path": "/3npygfmEhqnmNTmDWhHLz1LPcbA.jpg",
      "release_date": "2009-12-15",
      "original_title": "Avatar",
      "upperCaseTitle": "AVATAR"
    },
    {
      "title": "Avatar : La Voie de l'eau",
      "overview": "Une dizaine d'années se sont écoulés depuis les précédents évènements survenus sur
      "poster_path": "/hYeB9GpFaT7ysabBoGG5rbo9mF4.jpg",
      "release_date": "2022-12-14",
      "original_title": "Avatar: The Way of Water",
      "upperCaseTitle": "AVATAR : LA VOIE DE L'EAU"
    },
    {
      "title": "Avatar",
      "overview": "Tension mounts between a quadraplegic man and his wife as she prepares a bath for h
      "poster_path": "/gmnD2e1RvMdCl9D1rsDEQaQlJxK.jpg",
      "release_date": "2006-04-11",
      "original_title": "Avatar",
      "upperCaseTitle": "AVATAR"
    },
    {
      "title": "Les secrets du monde d'Avatar",
      "overview": "Une plongée dans les coulisses de l'une des suites les plus attendues du cinéma, av
      "poster_path": "/gCph6Kdq6T7NX1aflsJpuHhfo9j.jpg",
      "release_date": "2022-12-14",
      "original_title": "Avatar: The Deep Dive – A Special Edition of 20/20",
```

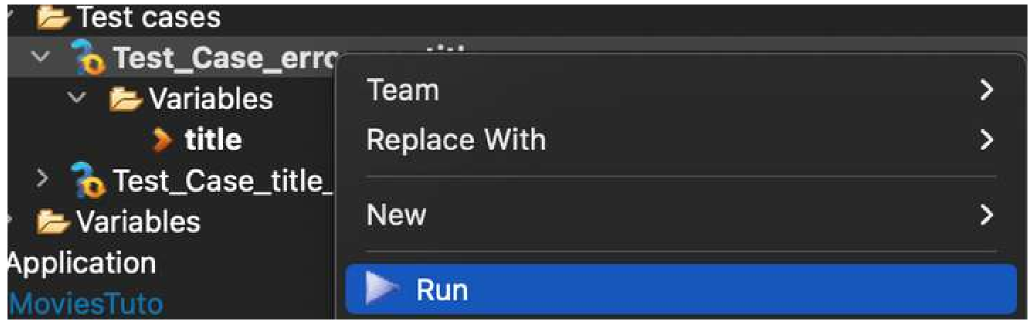
MyMoviesProject [C: HTTP_connector_MyMoviesProject].json

```
th":"/
8rpDcsfLJypb
O6vREc0547
VKqEv.jpg","g
enre_ids":
[878,12,28],"i
d":76600,"ori
ginal_languag
e":"en","origin
al_title":"Avat
ar: The Way of
Water","overvi
ew":"Une
dizaine
d'années se
sont écoulés
depuis les
précédents
évènements
survenus sur
Pandora. Jake
Sully et Neytiri
sont devenus
parents. Leur
vie idyllique,
proche de la
nature, est
menacée
lorsque la «
Ressources
Développeme
nt
Administration
», dangereuse
organisation
non-
gouvernement
```

```
{
  "object": {
    "page": 1,
    "results": [
      {
        "adult": false,
        "backdrop_path": "/vL5LR6WdxWPjLPFRLe133jXWsh5.jpg",
        "genre_ids": [
          28,
          12,
          14,
          878
        ],
        "id": 19995,
        "original_language": "en",
        "original_title": "Avatar",
        "overview": "Un marine paraplégique, envoyé sur la lune Pandora pour une mis
        "popularity": 126.67,
        "poster_path": "/3npygfmEhqnmNTmDWhHLz1LPcbA.jpg",
        "release_date": "2009-12-15",
        "title": "Avatar",
        "video": false,
        "vote_average": 7.575,
        "vote_count": 29955
      },
      {
        "adult": false,
        "backdrop_path": "/8rpDcsfLJypbO6vREc0547VKqEv.jpg",
        "genre_ids": [
          878,
          12,
          28
        ],
        "id": 76600,
        "original_language": "en",
        "original_title": "Avatar: The Way of Water",
        "overview": "Une dizaine d'années se sont écoulés depuis les précédents évèr
        "popularity": 291.271,
        "poster_path": "/hYeB9GpFaT7ysabBoGG5rbo9mF4.jpg",
        "release_date": "2022-12-14",
        "title": "Avatar : La Voie de l'eau",
        "video": false,
```

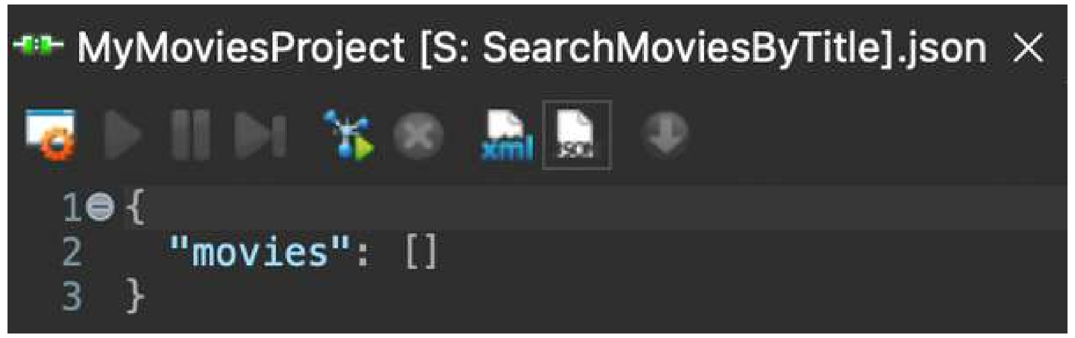
6.4 Using the IfExist step

Let’s run the error test case



Without a title,

the sequence returns an empty array

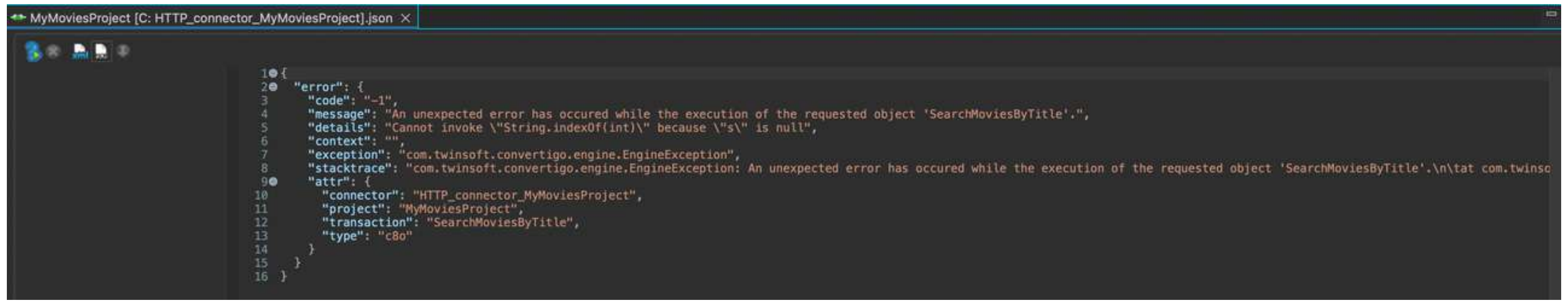


MyMoviesProject [S: SearchMoviesByTitle].json

```
{
  "movies": []
}
```

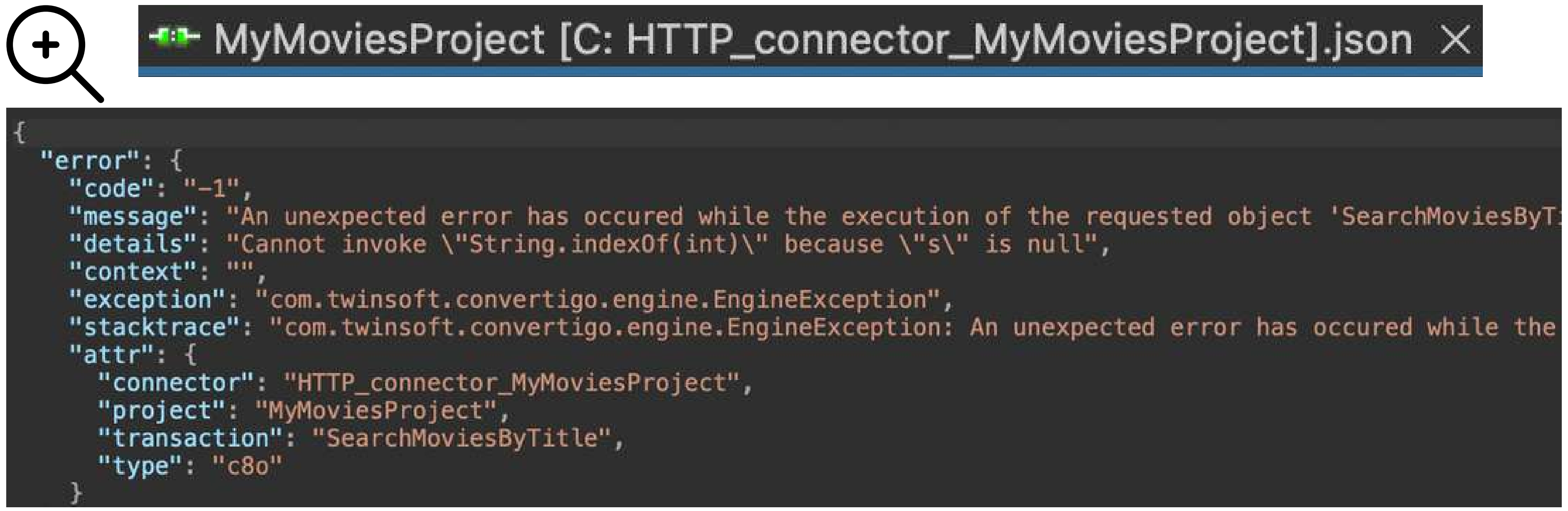
The **transaction returns an error**,

and an **error tag is generated.**



MyMoviesProject [C: HTTP_connector_MyMoviesProject].json

```
{
  "error": {
    "code": "-1",
    "message": "An unexpected error has occured while the execution of the requested object 'SearchMoviesByTitle'.",
    "details": "Cannot invoke \"String.indexOf(int)\" because \"s\" is null",
    "context": "",
    "exception": "com.twinsoft.convertigo.engine.EngineException",
    "stacktrace": "com.twinsoft.convertigo.engine.EngineException: An unexpected error has occured while the execution of the requested object 'SearchMoviesByTitle'.\n\tat com.twinso
    "attr": {
      "connector": "HTTP_connector_MyMoviesProject",
      "project": "MyMoviesProject",
      "transaction": "SearchMoviesByTitle",
      "type": "c8o"
    }
  }
}
```



MyMoviesProject [C: HTTP_connector_MyMoviesProject].json

```
{
  "error": {
    "code": "-1",
    "message": "An unexpected error has occured while the execution of the requested object 'SearchMoviesByT
    "details": "Cannot invoke \"String.indexOf(int)\" because \"s\" is null",
    "context": "",
    "exception": "com.twinsoft.convertigo.engine.EngineException",
    "stacktrace": "com.twinsoft.convertigo.engine.EngineException: An unexpected error has occured while the
    "attr": {
      "connector": "HTTP_connector_MyMoviesProject",
      "project": "MyMoviesProject",
      "transaction": "SearchMoviesByTitle",
      "type": "c8o"
    }
```

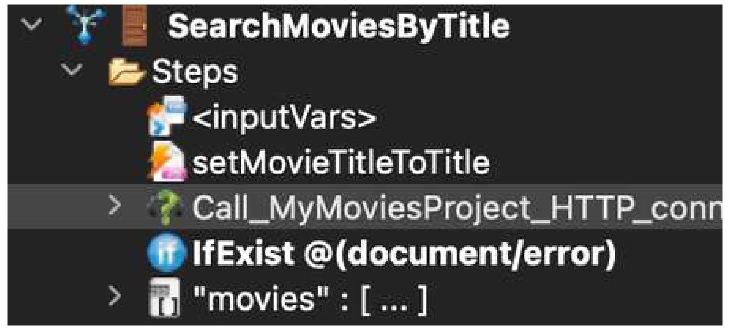
The IfExists step detects the error.

We are going to use it in combination to other steps to handle this error.

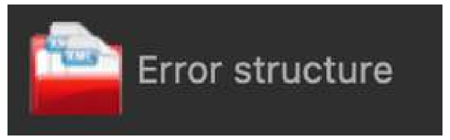
6.5 Using the Error Structure step

To report the error to the client, we use the **Error Structure step**,

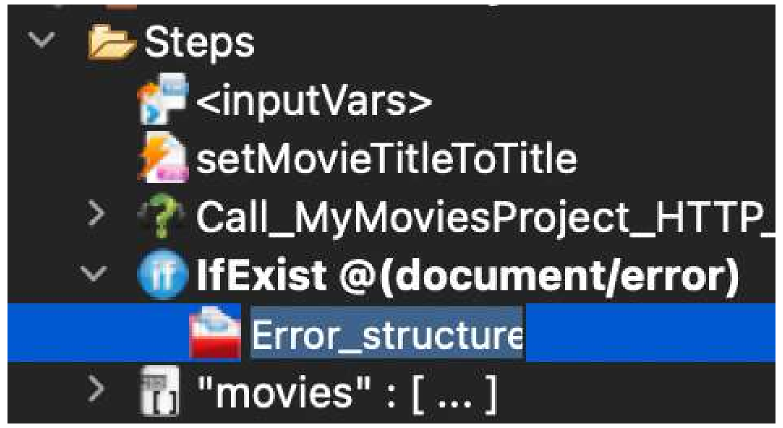
Let’s add an **Error structure step** in the **IfExist step**.



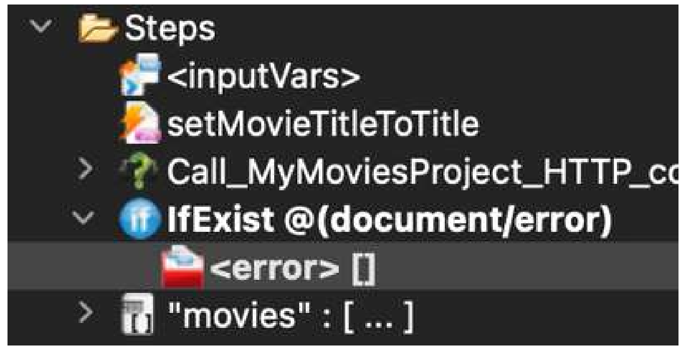
```
SearchMoviesByTitle
  Steps
    <inputVars>
    setMovieTitleToTitle
  > Call_MyMoviesProject_HTTP_conn
    IfExist @(document/error)
  > "movies" : [ ... ]
```



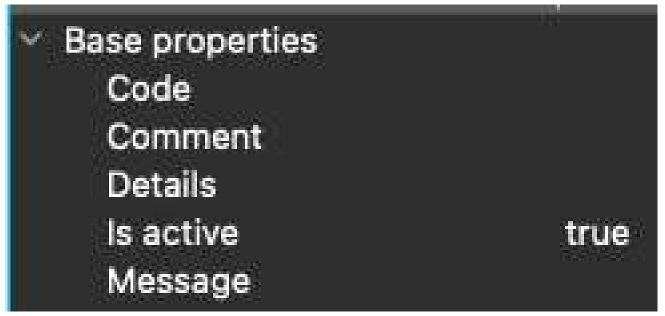
Error structure



```
Steps
  <inputVars>
  setMovieTitleToTitle
> Call_MyMoviesProject_HTTP_
  IfExist @(document/error)
    Error_structure
> "movies" : [ ... ]
```



```
Steps
  <inputVars>
  setMovieTitleToTitle
> Call_MyMoviesProject_HTTP_cc
  IfExist @(document/error)
    <error> []
> "movies" : [ ... ]
```



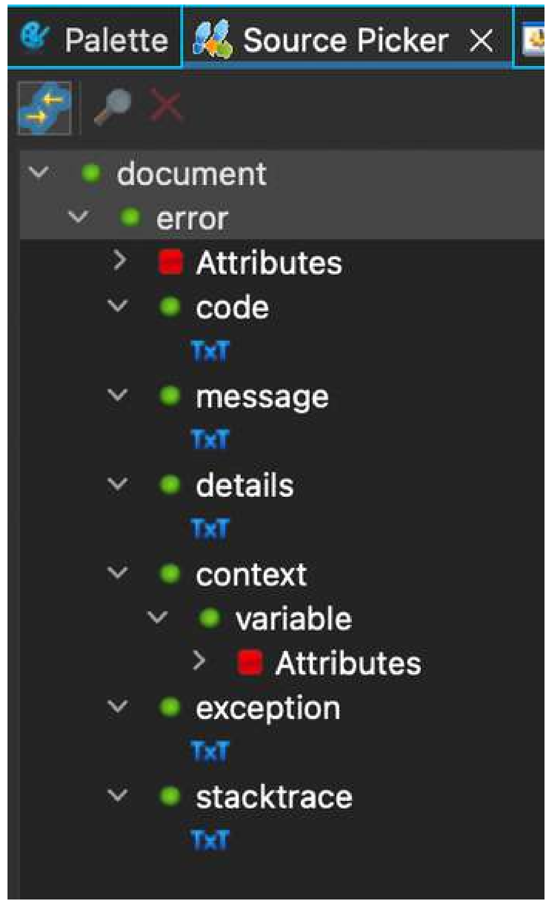
Base properties	
Code	
Comment	
Details	
Is active	true
Message	

This step has the properties

- **Code** (error status code)
- **Message**
- **Details**

These properties can be **sourced from the original error message** returned by the API in the transaction.

In the source picker, the **XML Structure** of the **Error structure step** has the same properties.

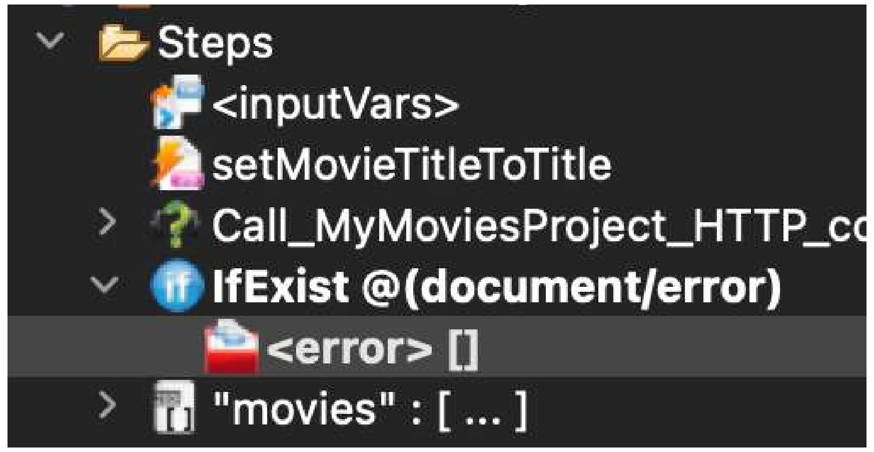


Palette | Source Picker

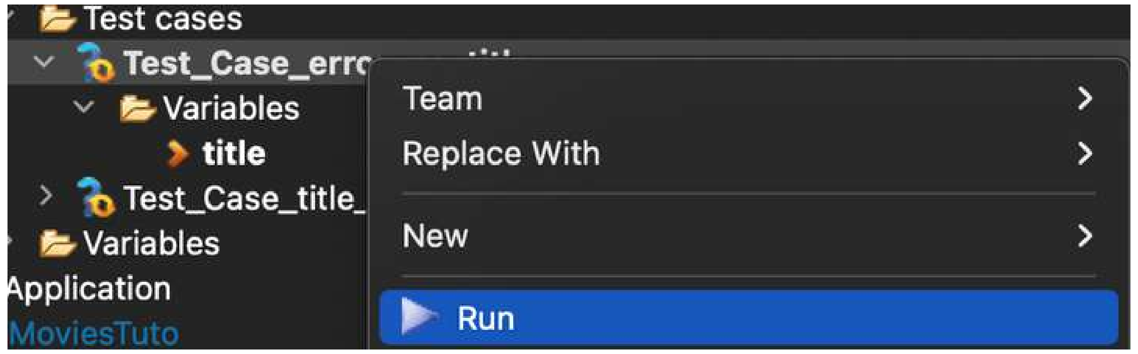
```
document
  error
    > Attributes
      code
        TxT
      message
        TxT
      details
        TxT
      context
        variable
          > Attributes
      exception
        TxT
      stacktrace
        TxT
```

6.5 Using the Error Structure step

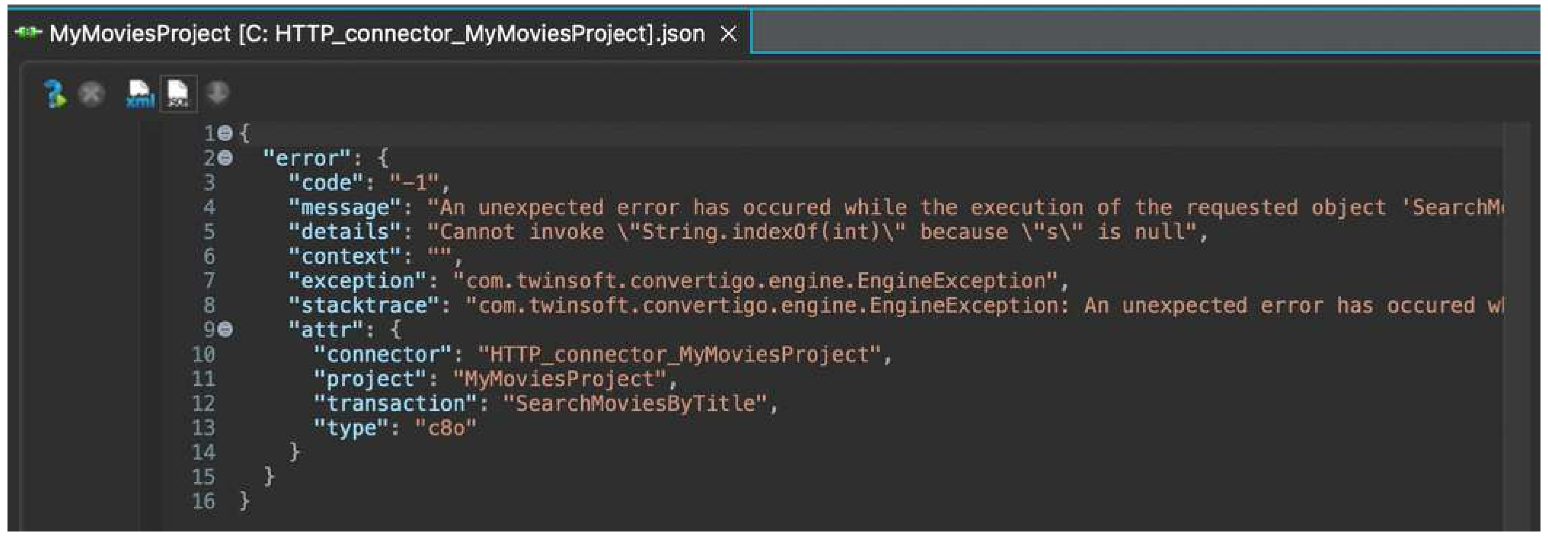
Let’s see what difference it makes to have this Error structure step in the sequence.



Let’s run the error test case again.



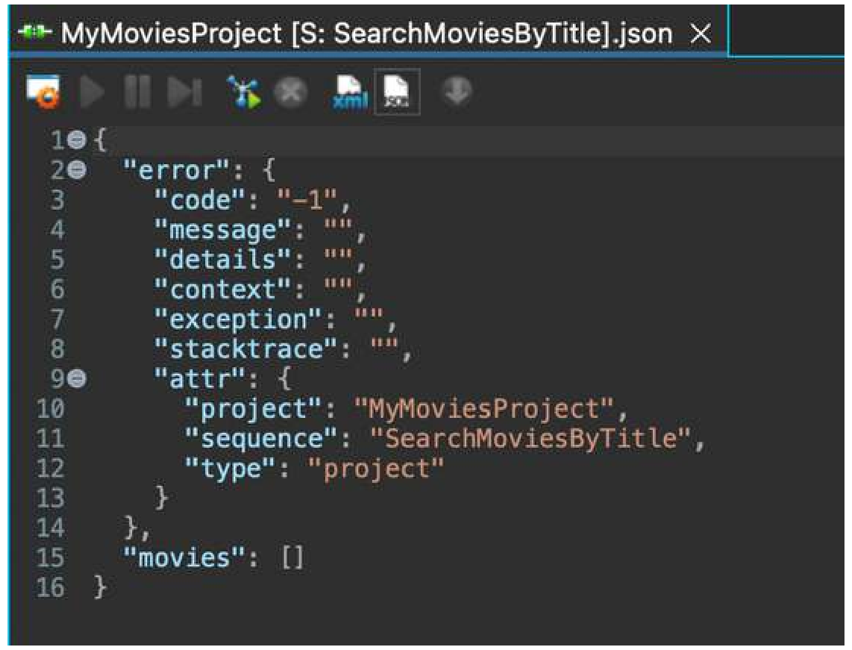
The original error message returned by the transaction is the same



MyMoviesProject [C: HTTP_connector_MyMoviesProject].json

```
{
  "error": {
    "code": "-1",
    "message": "An unexpected error has occured while the execution of the requested object 'SearchM
    "details": "Cannot invoke \"String.indexOf(int)\" because \"s\" is null",
    "context": "",
    "exception": "com.twinsoft.convertigo.engine.EngineException",
    "stacktrace": "com.twinsoft.convertigo.engine.EngineException: An unexpected error has occured wh
    "attr": {
      "connector": "HTTP_connector_MyMoviesProject",
      "project": "MyMoviesProject",
      "transaction": "SearchMoviesByTitle",
      "type": "c8o"
    }
  }
}
```

The sequence returns **an empty array and an error**.

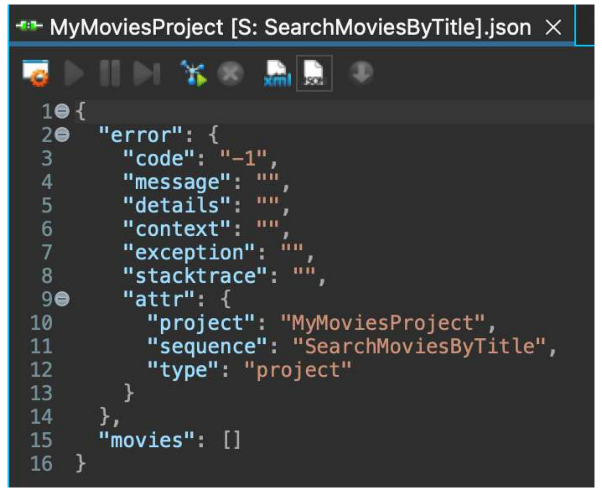


MyMoviesProject [S: SearchMoviesByTitle].json

```
{
  "error": {
    "code": "-1",
    "message": "",
    "details": "",
    "context": "",
    "exception": "",
    "stacktrace": "",
    "attr": {
      "project": "MyMoviesProject",
      "sequence": "SearchMoviesByTitle",
      "type": "project"
    }
  },
  "movies": []
}
```

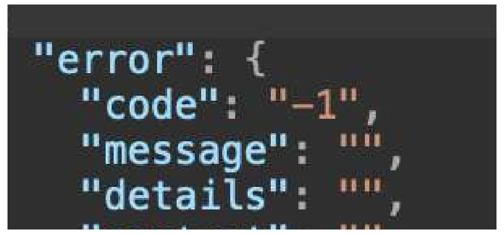
6.5 Using the Error Structure step

By changing the properties of the Error structure step, we can customize the error returned by the sequence.



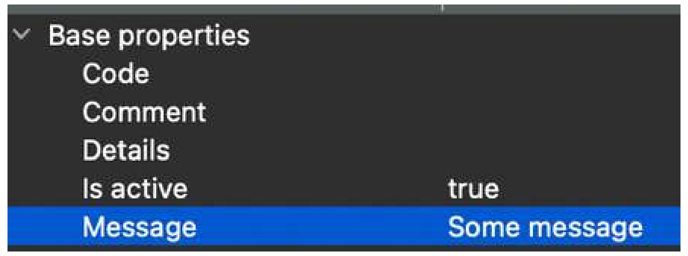
MyMoviesProject [S: SearchMoviesByTitle].json

```
{
  "error": {
    "code": "-1",
    "message": "",
    "details": "",
    "context": "",
    "exception": "",
    "stacktrace": "",
    "attr": {
      "project": "MyMoviesProject",
      "sequence": "SearchMoviesByTitle",
      "type": "project"
    }
  },
  "movies": []
}
```



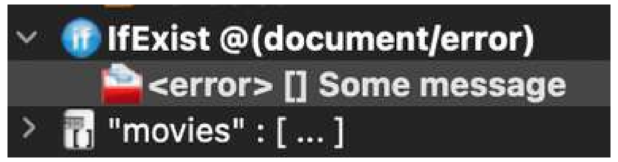
```
"error": {
  "code": "-1",
  "message": "",
  "details": "",
```

Let’s customize our error message.



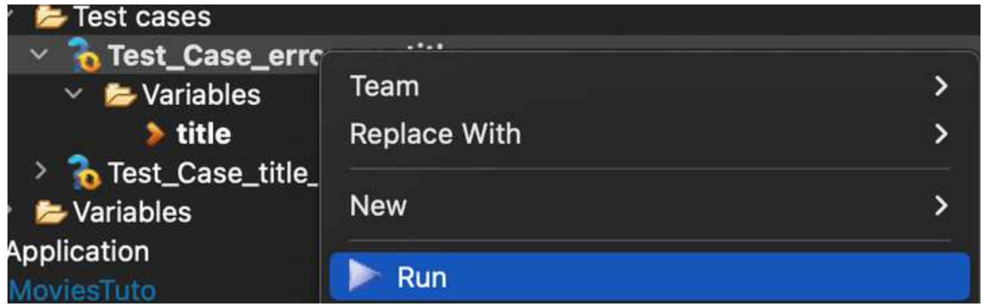
Base properties
Code
Comment
Details
Is active: true
Message: Some message

We can enter a error message directly in the message properties.

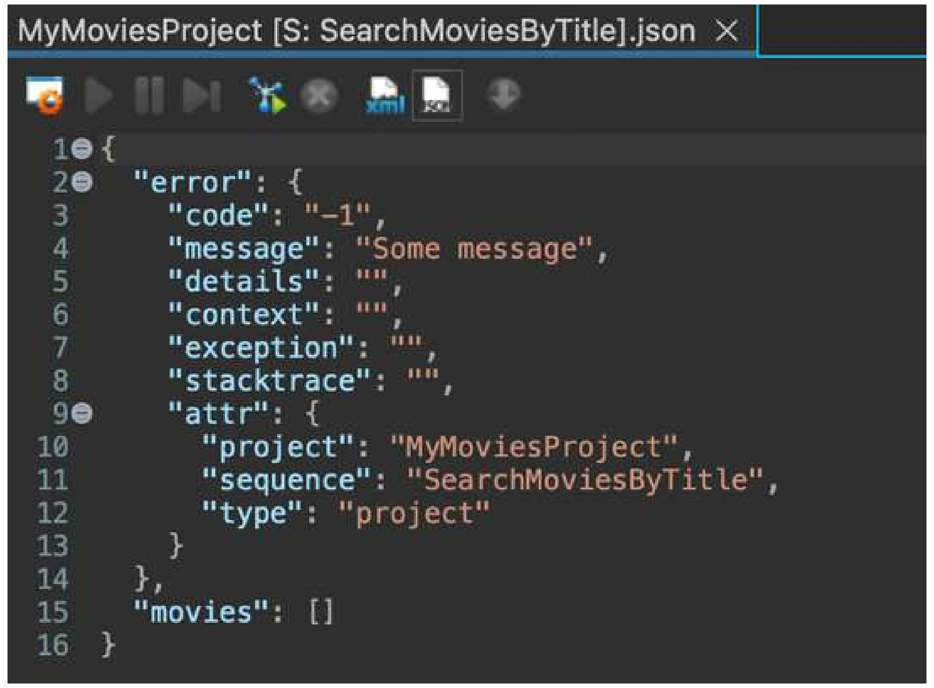


IfExist @(document/error)
<error> [] Some message
"movies" : [...]

The error message appears in the error structure step of the sequence.



When running the test case, the message will appear in the return of the execution of the sequence



MyMoviesProject [S: SearchMoviesByTitle].json

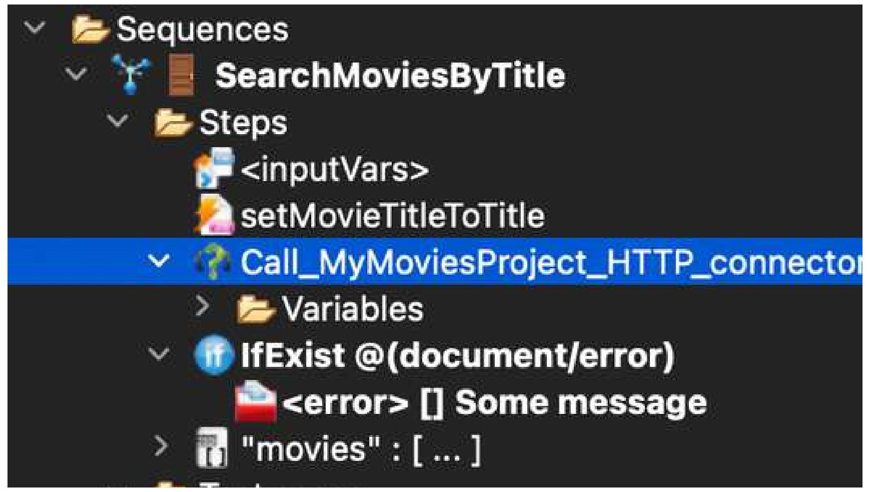
```
{
  "error": {
    "code": "-1",
    "message": "Some message",
    "details": "",
    "context": "",
    "exception": "",
    "stacktrace": "",
    "attr": {
      "project": "MyMoviesProject",
      "sequence": "SearchMoviesByTitle",
      "type": "project"
    }
  },
  "movies": []
}
```

6.5 Using the Error Structure step

Now, let’s **customize the error** returned by the sequence **dynamically**.

by using the original error message returned by the API,

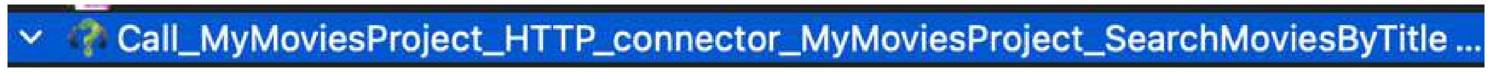
and the properties of the Error Structure step.



```
v Sequences
  v SearchMoviesByTitle
    v Steps
        <inputVars>
        setMovieTitleToTitle
      v Call_MyMoviesProject_HTTP_connector
        > Variables
      v IfExist @(document/error)
          <error> [] Some message
      > "movies" : [ ... ]
```

Click twice on the transaction call

to display its XML structure in the source picker.



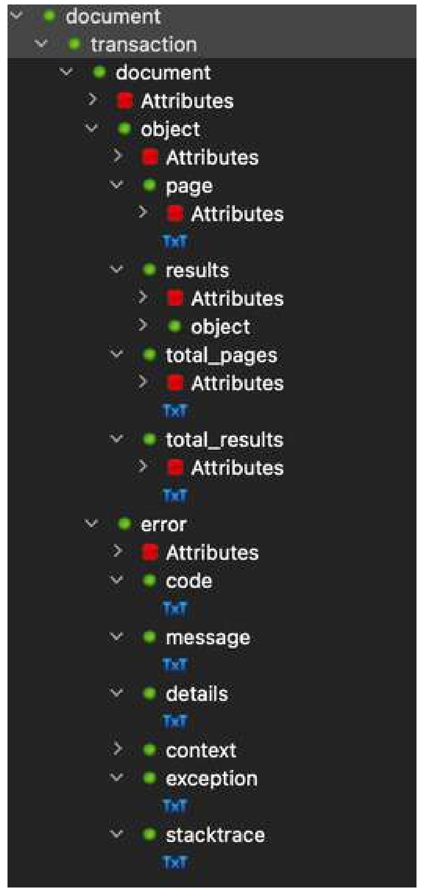
```
v Call_MyMoviesProject_HTTP_connector_MyMoviesProject_SearchMoviesByTitle ...
```

In the source picker,

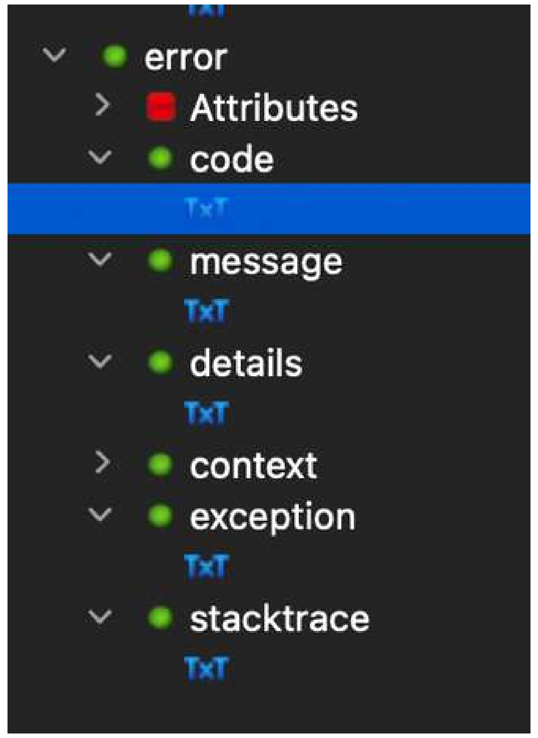
we can see the **nodes code, message and details**.

We are going to use these nodes

to **source the properties** of the Error Structure step.

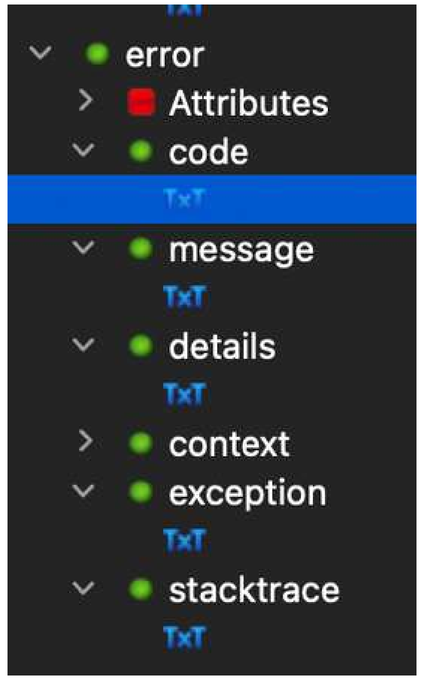


```
v document
  v transaction
    v document
      > Attributes
      v object
        > Attributes
        v page
          > Attributes
            TxT
        v results
          > Attributes
          > object
        v total_pages
          > Attributes
            TxT
        v total_results
          > Attributes
            TxT
      v error
        > Attributes
        v code
            TxT
        v message
            TxT
        v details
            TxT
        > context
        v exception
            TxT
        v stacktrace
            TxT
```



```
v error
  > Attributes
  v code
      TxT
  v message
      TxT
  v details
      TxT
  > context
  v exception
      TxT
  v stacktrace
      TxT
```

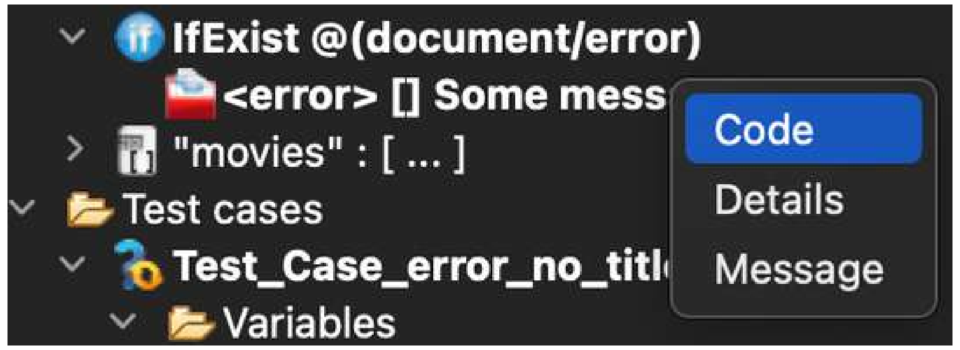
6.5 Using the Error Structure step



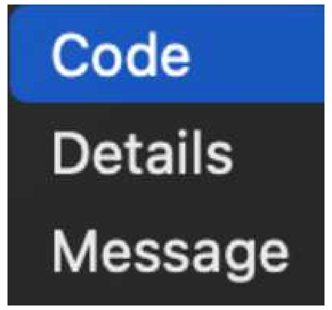
```
TxT
⌄ ● error
  › ● Attributes
  ⌄ ● code
      TxT
  ⌄ ● message
      TxT
  ⌄ ● details
      TxT
  › ● context
  ⌄ ● exception
      TxT
  ⌄ ● stacktrace
      TxT
```

Drag the TxT from the code node

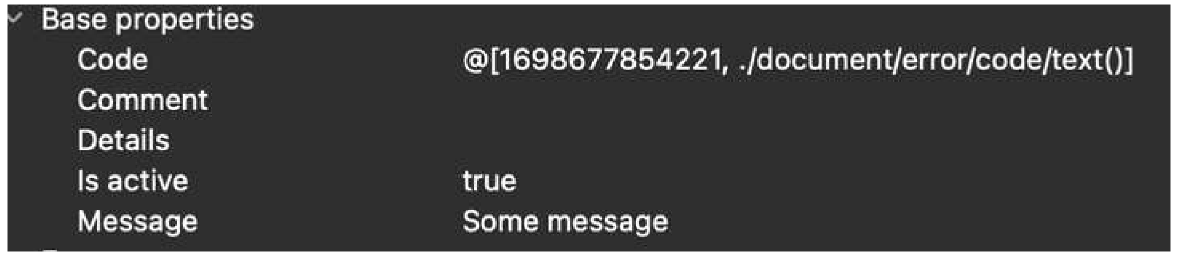
into the error structure step in the sequence.



```
⌄ IfExist @(document/error)
    <error> [] Some mess
  › "movies" : [ ... ]
⌄ Test cases
  ⌄ Test_Case_error_no_titl
    ⌄ Variables
```

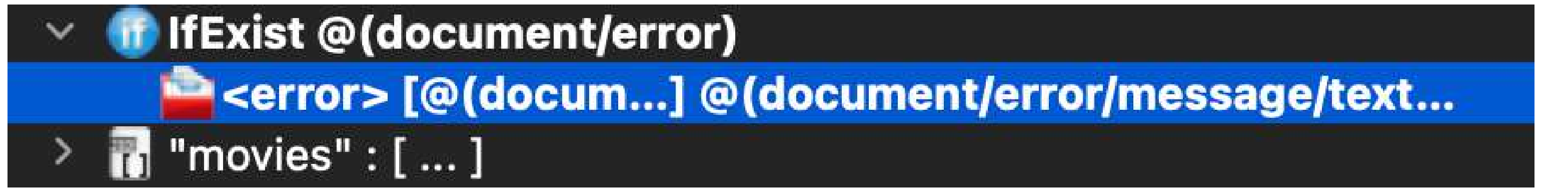


The source appears in the properties

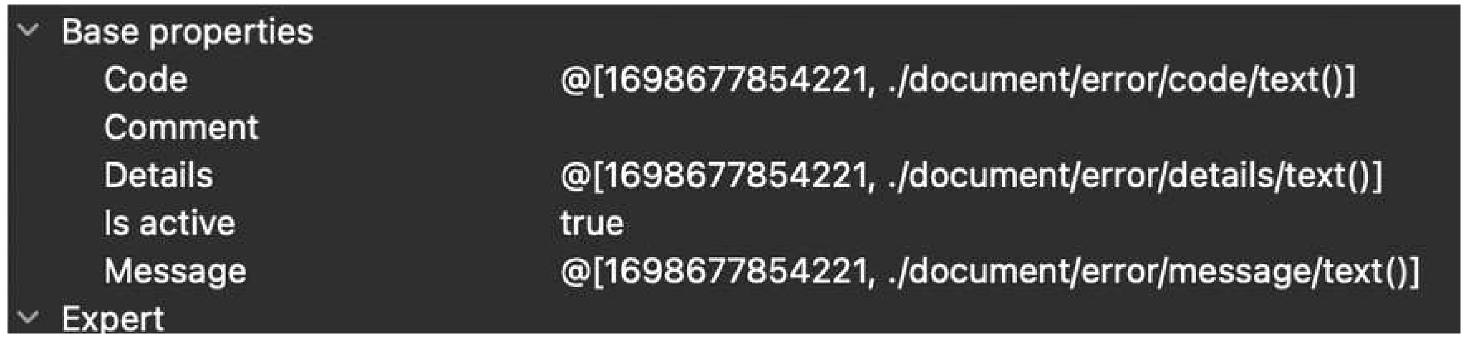


Base properties	
Code	@[1698677854221, ./document/error/code/text()]
Comment	
Details	
Is active	true
Message	Some message

Do the same thing with the TxT from the message and the detail nodes



```
⌄ IfExist @(document/error)
    <error> [@(docum...] @(document/error/message/text...
  › "movies" : [ ... ]
```

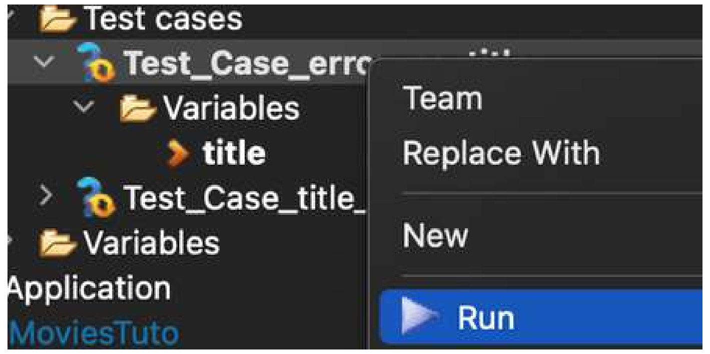


Base properties	
Code	@[1698677854221, ./document/error/code/text()]
Comment	
Details	@[1698677854221, ./document/error/details/text()]
Is active	true
Message	@[1698677854221, ./document/error/message/text()]
Expert	

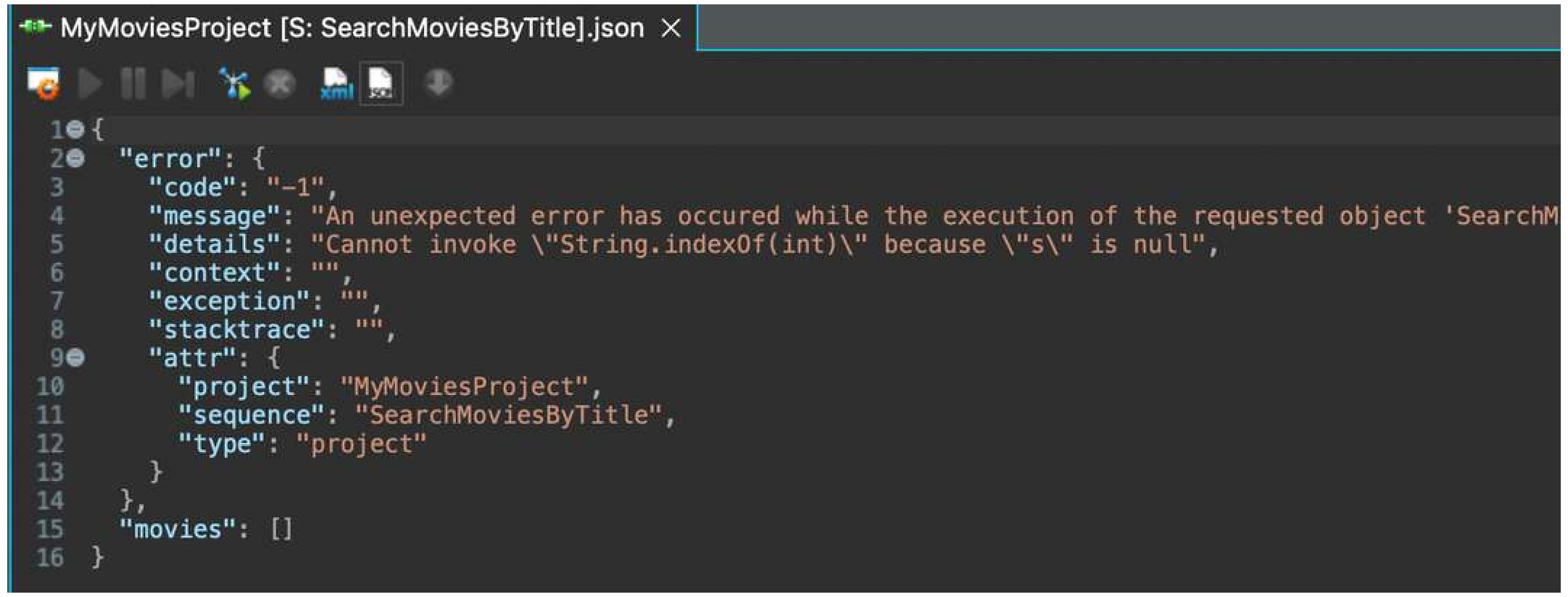
6.5 Using the Error Structure step

When we run the test case, the error returned by the sequence

has the same code, message and details as the original error message returned by the API.



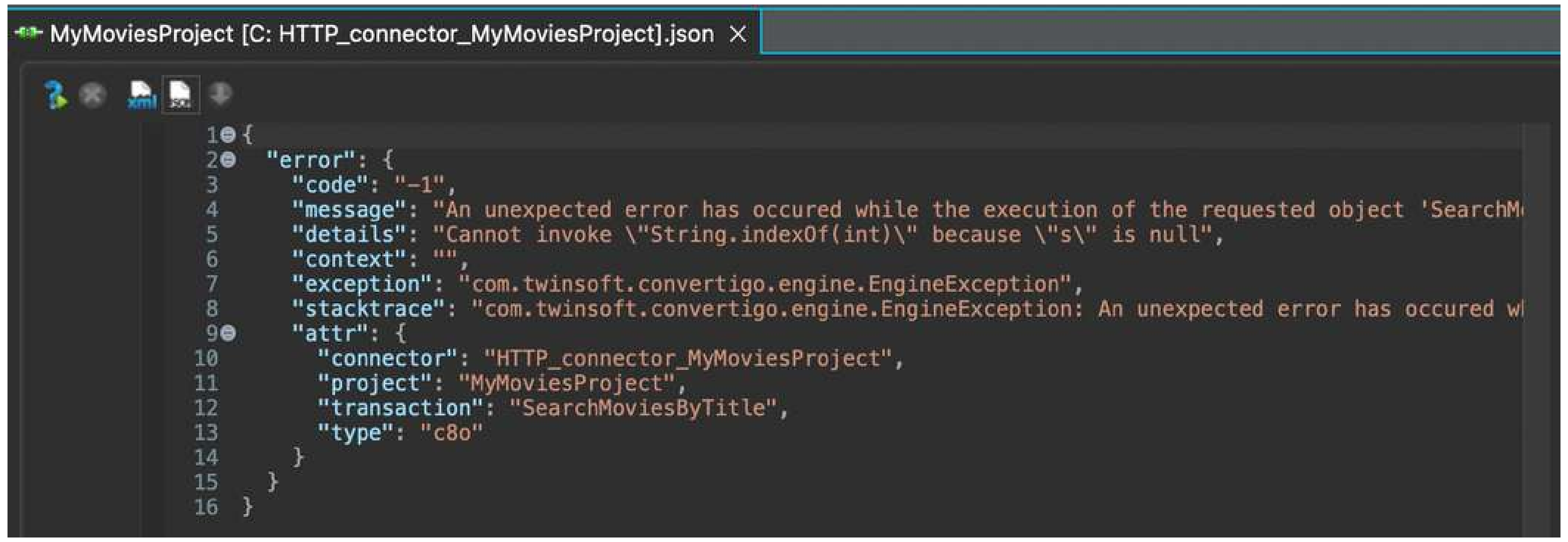
Error returned by the sequence



MyMoviesProject [S: SearchMoviesByTitle].json

```
{
  "error": {
    "code": "-1",
    "message": "An unexpected error has occured while the execution of the requested object 'SearchM
    "details": "Cannot invoke \"String.indexOf(int)\" because \"s\" is null",
    "context": "",
    "exception": "",
    "stacktrace": "",
    "attr": {
      "project": "MyMoviesProject",
      "sequence": "SearchMoviesByTitle",
      "type": "project"
    }
  },
  "movies": []
}
```

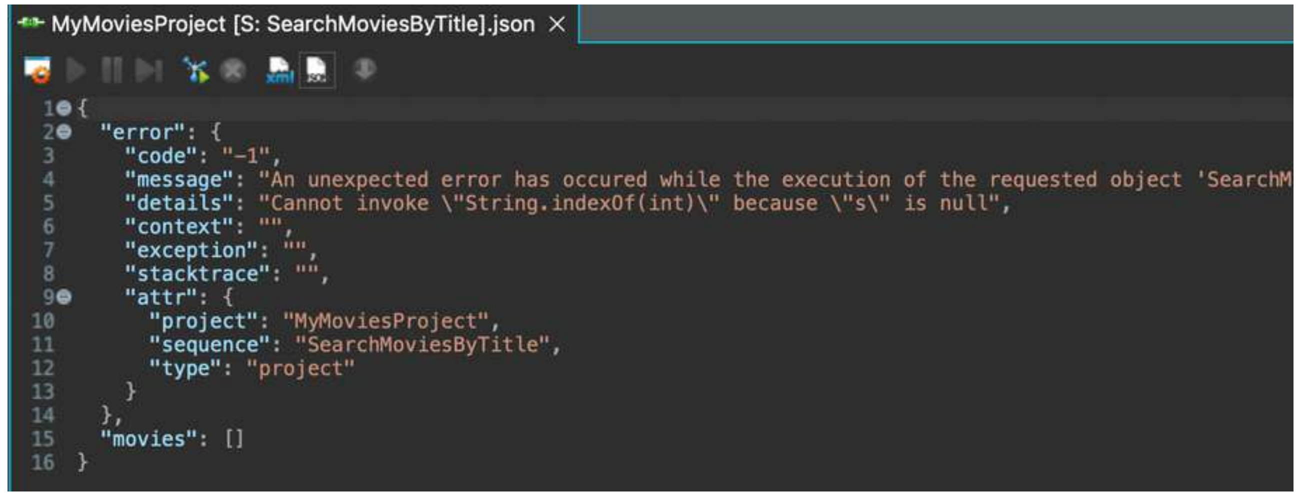
Original error message returned by the API



MyMoviesProject [C: HTTP_connector_MyMoviesProject].json

```
{
  "error": {
    "code": "-1",
    "message": "An unexpected error has occured while the execution of the requested object 'SearchM
    "details": "Cannot invoke \"String.indexOf(int)\" because \"s\" is null",
    "context": "",
    "exception": "com.twinsoft.convertigo.engine.EngineException",
    "stacktrace": "com.twinsoft.convertigo.engine.EngineException: An unexpected error has occured wl
    "attr": {
      "connector": "HTTP_connector_MyMoviesProject",
      "project": "MyMoviesProject",
      "transaction": "SearchMoviesByTitle",
      "type": "c8o"
    }
  }
}
```

6.6 Using the Return step

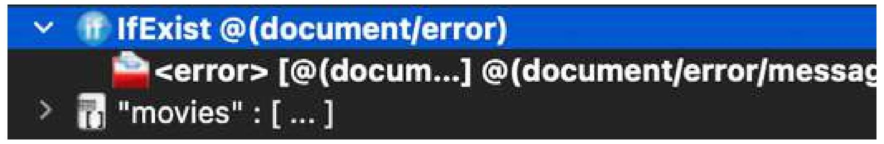

MyMoviesProject [S: SearchMoviesByTitle].json

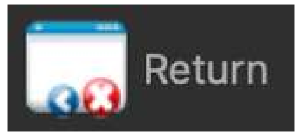
```
{
  "error": {
    "code": "-1",
    "message": "An unexpected error has occured while the execution of the requested object 'SearchM
    "details": "Cannot invoke \"String.indexOf(int)\" because \"s\" is null",
    "context": "",
    "exception": "",
    "stacktrace": "",
    "attr": {
      "project": "MyMoviesProject",
      "sequence": "SearchMoviesByTitle",
      "type": "project"
    }
  },
  "movies": []
}
```

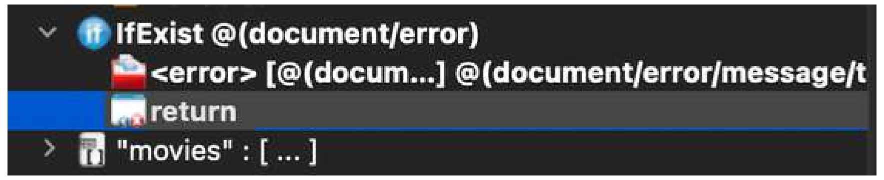
After the error structure step, the sequence didn’t stop and the following steps were executed.

That’s why we see an empty movies array after the error.

To stop the sequence after an error,


IfExist @(document/error)
<error> [@(docum...] @(document/error/messag
"movies" : [...]



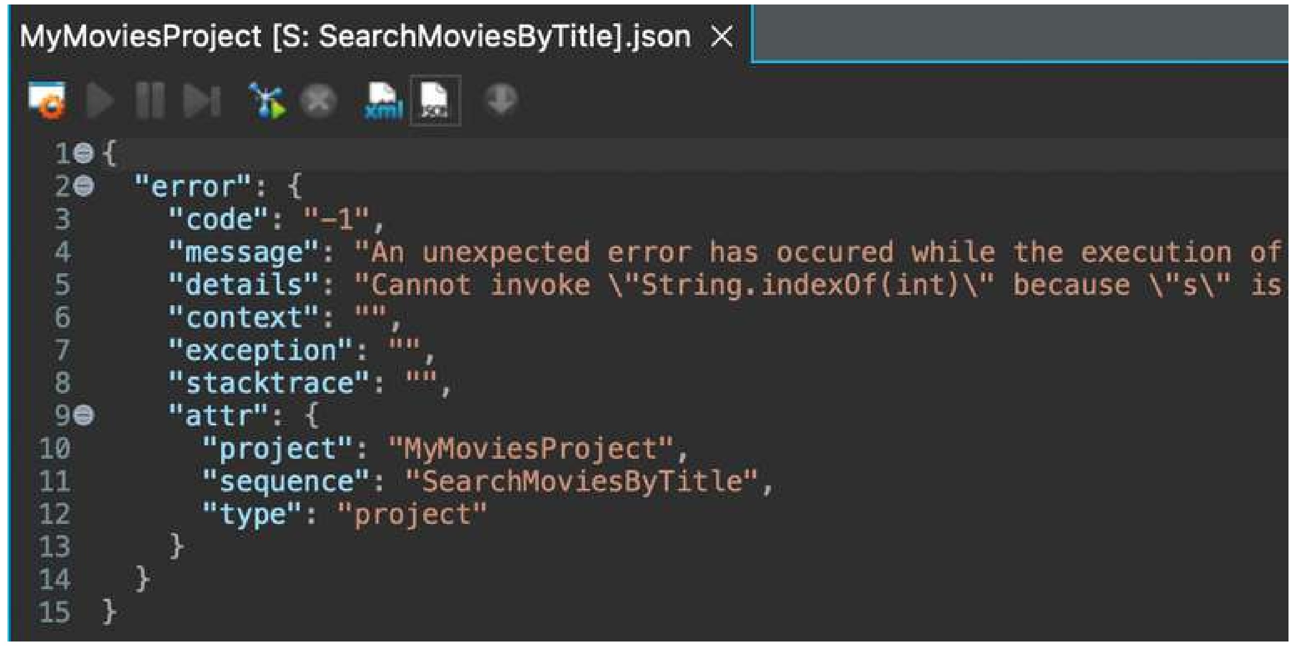

IfExist @(document/error)
<error> [@(docum...] @(document/error/message/t
return
"movies" : [...]

we add a **Return step**

after the Error Structure step

in the IfExist step.

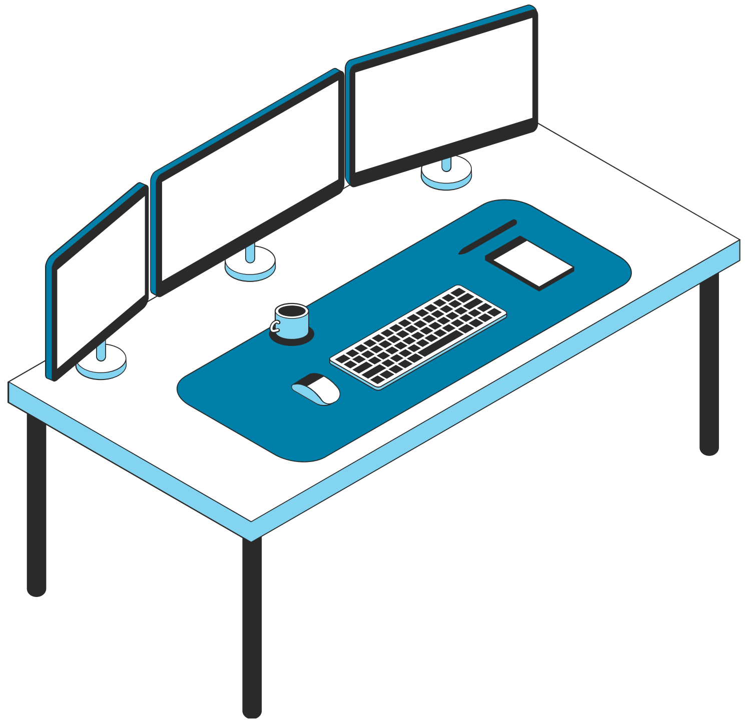
The sequence execution is stopped and the empty movies array disappears from the result of the sequence.


MyMoviesProject [S: SearchMoviesByTitle].json

```
{
  "error": {
    "code": "-1",
    "message": "An unexpected error has occured while the execution of
    "details": "Cannot invoke \"String.indexOf(int)\" because \"s\" is
    "context": "",
    "exception": "",
    "stacktrace": "",
    "attr": {
      "project": "MyMoviesProject",
      "sequence": "SearchMoviesByTitle",
      "type": "project"
    }
  }
}
```

7 – Collaboration with Git

How to share your projects with Git Versioning.



7.1 Git basics with Convertigo

7.2 Git Repositories View

7.3 Git Staging View

7.4 Compare mode

7.5 Commit your changes

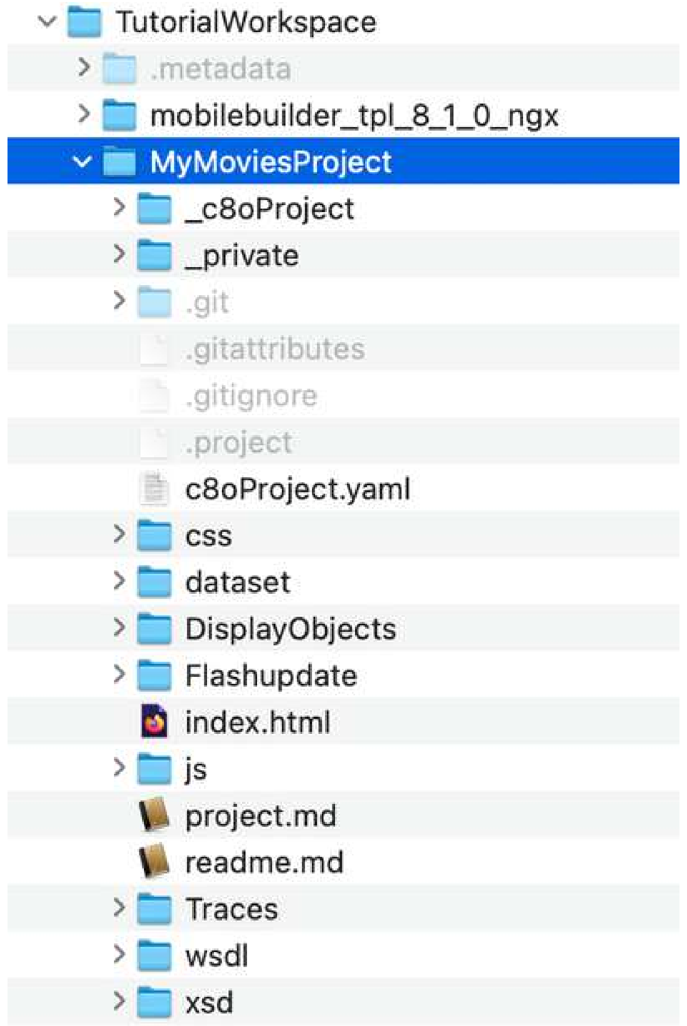
7.6 Clone a project

7.1 Git basics with Convertigo

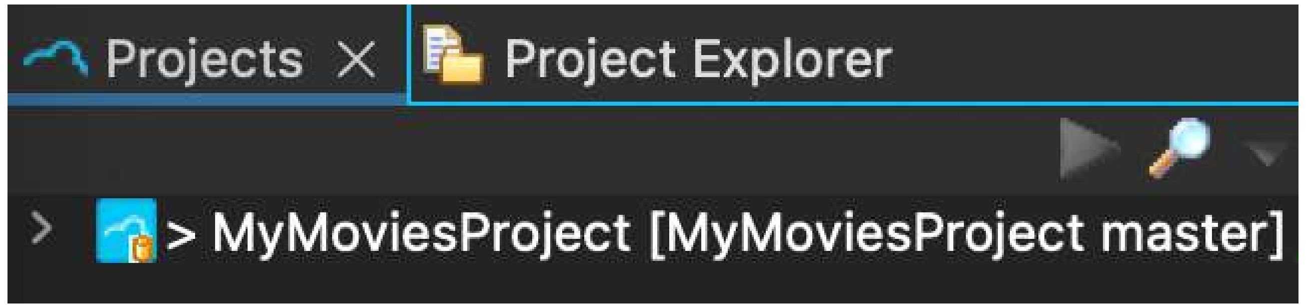
When you create a new project, a **Git Repository is automatically created**.

In the Projects folder, the name of your project

is followed by the **name of the branch** you're currently working on.



```
TutorialWorkspace
  .metadata
  mobilebuilder_tpl_8_1_0_ngx
  MyMoviesProject
    _c8oProject
    _private
    .git
    .gitattributes
    .gitignore
    .project
    c8oProject.yaml
    css
    dataset
    DisplayObjects
    Flashupdate
    index.html
    js
    project.md
    readme.md
    Traces
    wsdl
    xsd
```



> MyMoviesProject [MyMoviesProject master]

In the studio interface, two views are used to manage Git in your projects.

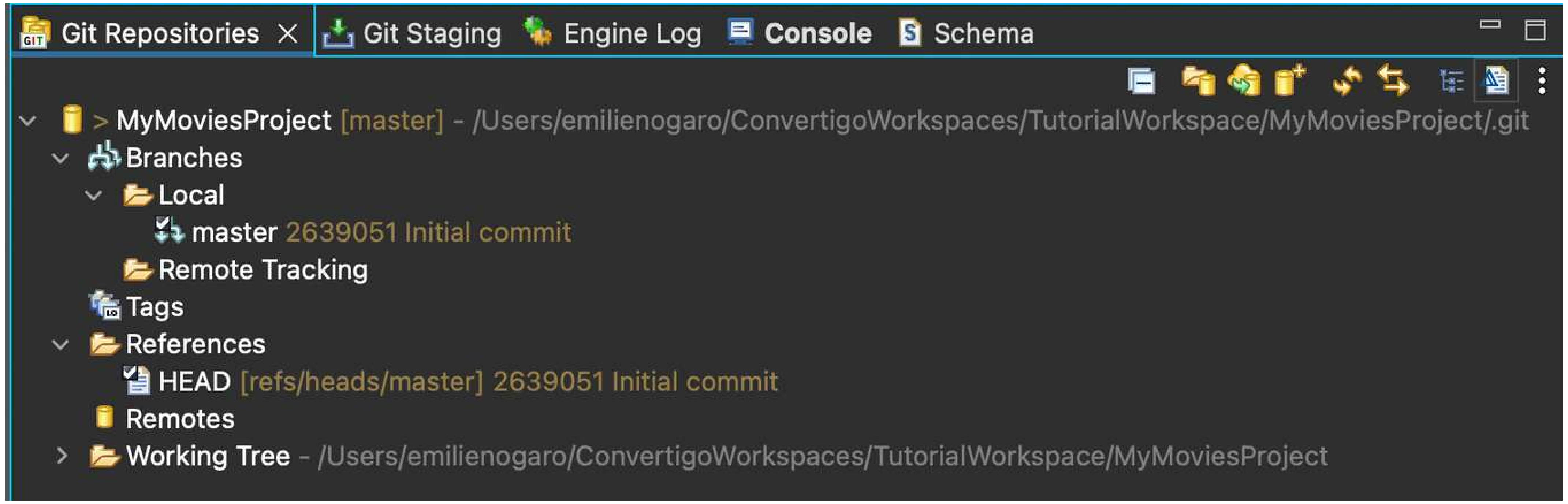
- Git Repositories

- Git Staging

7.2 Git Repositories View

In the **Git Repositories View**,

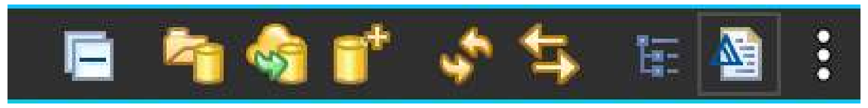
you can see the Git Repositories of **all the projects in your workspace**.

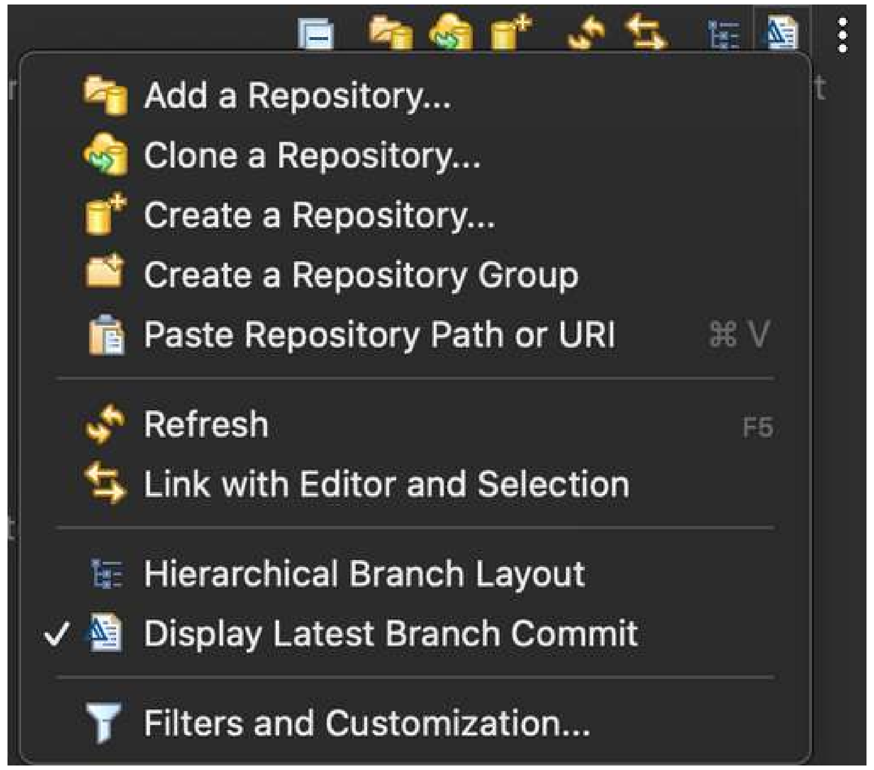


```
> MyMoviesProject [master] - /Users/emilienogaro/ConvertigoWorkspaces/TutorialWorkspace/MyMoviesProject/.git
  Branches
    Local
      master 2639051 Initial commit
    Remote Tracking
  Tags
  References
    HEAD [refs/heads/master] 2639051 Initial commit
  Remotes
  Working Tree - /Users/emilienogaro/ConvertigoWorkspaces/TutorialWorkspace/MyMoviesProject
```

It provides Eclipse features

to manage your Git repositories.

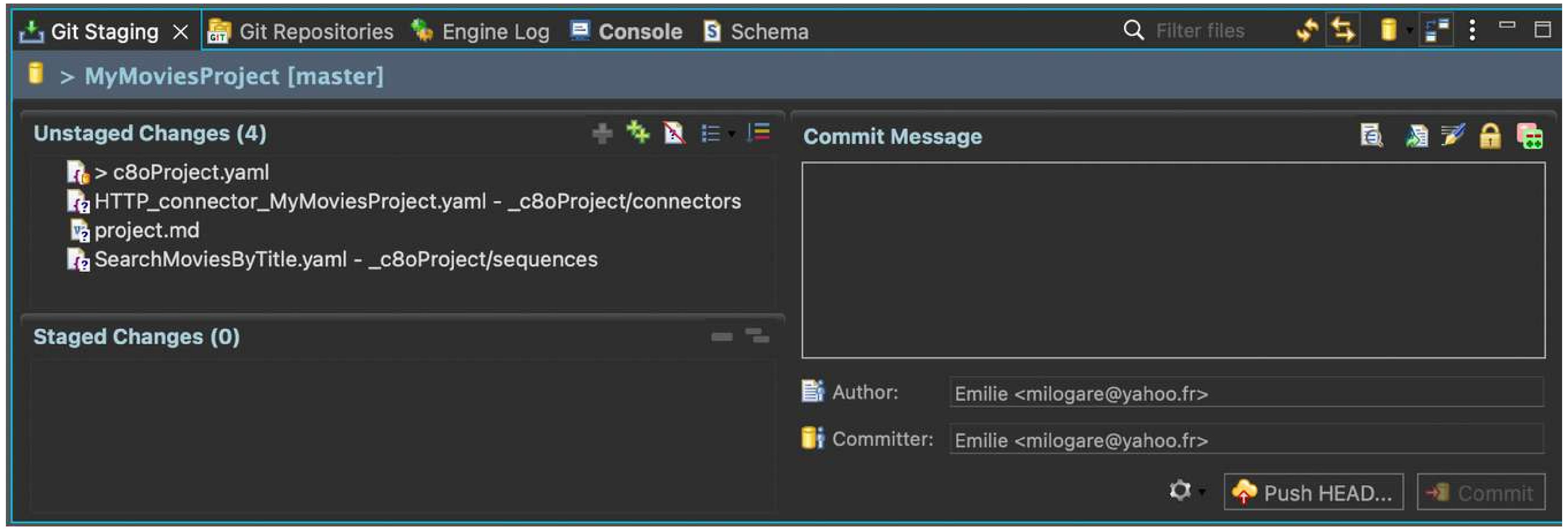



Add a Repository...
Clone a Repository...
Create a Repository...
Create a Repository Group
Paste Repository Path or URI ⌘ V
Refresh F5
Link with Editor and Selection
Hierarchical Branch Layout
✓ Display Latest Branch Commit
Filters and Customization...

7.3 Git Staging View

In the **Git Staging view**, you can manage your **git workflow**, and **commit your changes** to your **local and remote directories**.

The files that have been modified since the last commit are shown in **Unstaged changes**.

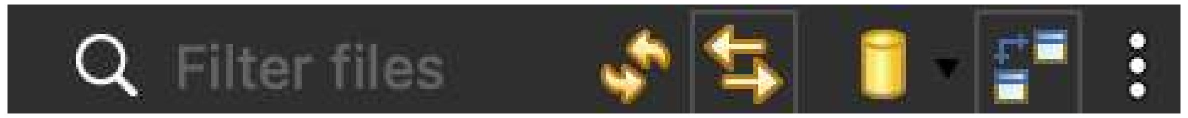


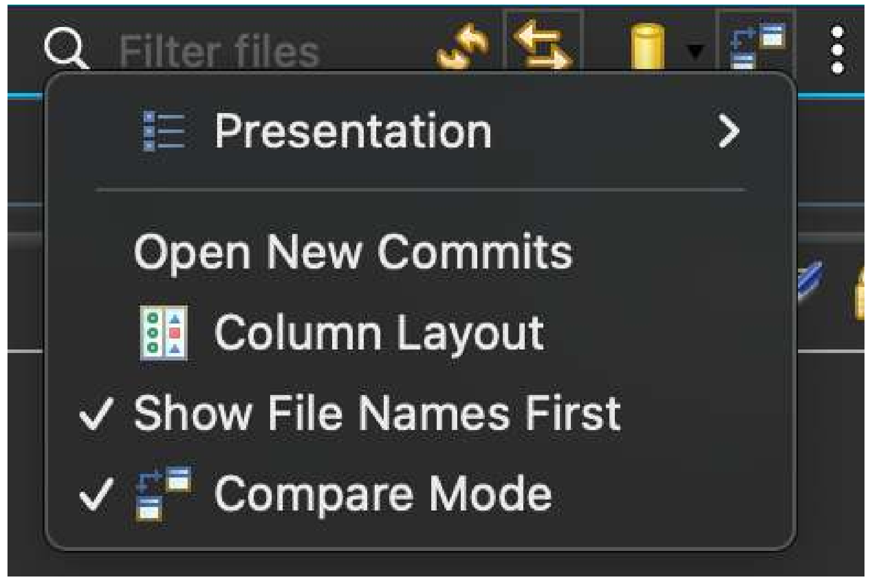
> c8oProject.yaml
HTTP_connector_MyMoviesProject.yaml - _c8oProject/connectors
project.md
SearchMoviesByTitle.yaml - _c8oProject/sequences

Author: Emilie <milogare@yahoo.fr>

Committer: Emilie <milogare@yahoo.fr>

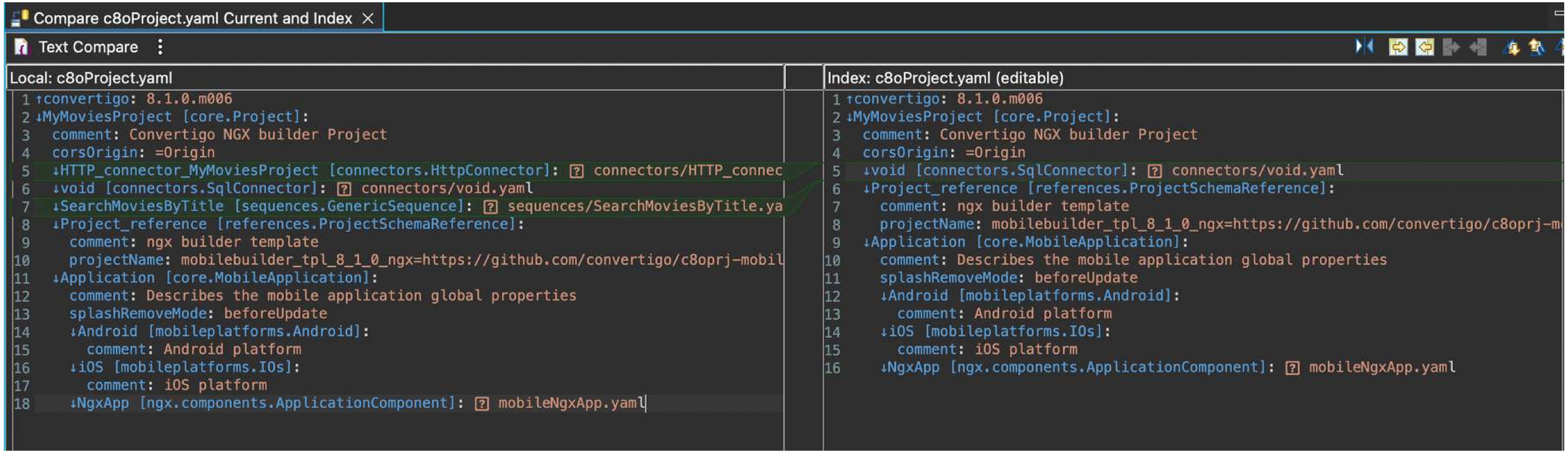
It provides Eclipse features to manage your Git workflow.





7.4 Compare Mode

With the **Compare Mode**, you can display the **differences with the previous commit**, and **resolve conflicts** when necessary.

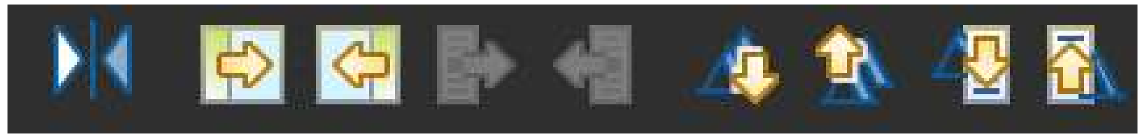


Local: c8oProject.yaml

```
↑convertigo: 8.1.0.m006
↓MyMoviesProject [core.Project]:
  comment: Convertigo NGX builder Project
  corsOrigin: =Origin
  ↓HTTP_connector_MyMoviesProject [connectors.HttpConnector]: ? connectors/HTTP_connec
  ↓void [connectors.SqlConnector]: ? connectors/void.yaml
  ↓SearchMoviesByTitle [sequences.GenericSequence]: ? sequences/SearchMoviesByTitle.ya
  ↓Project_reference [references.ProjectSchemaReference]:
    comment: ngx builder template
    projectName: mobilebuilder_tpl_8_1_0_ngx=https://github.com/convertigo/c8oprj-mobil
  ↓Application [core.MobileApplication]:
    comment: Describes the mobile application global properties
    splashRemoveMode: beforeUpdate
    ↓Android [mobileplatforms.Android]:
      comment: Android platform
    ↓iOS [mobileplatforms.IOs]:
      comment: iOS platform
    ↓NgxApp [ngx.components.ApplicationComponent]: ? mobileNgxApp.yaml
```

Index: c8oProject.yaml (editable)

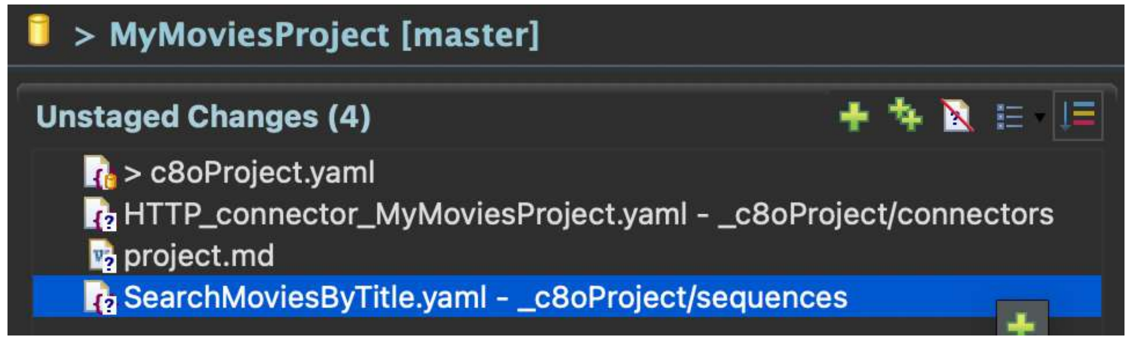
```
↑convertigo: 8.1.0.m006
↓MyMoviesProject [core.Project]:
  comment: Convertigo NGX builder Project
  corsOrigin: =Origin
  ↓void [connectors.SqlConnector]: ? connectors/void.yaml
  ↓Project_reference [references.ProjectSchemaReference]:
    comment: ngx builder template
    projectName: mobilebuilder_tpl_8_1_0_ngx=https://github.com/convertigo/c8oprj-m
  ↓Application [core.MobileApplication]:
    comment: Describes the mobile application global properties
    splashRemoveMode: beforeUpdate
    ↓Android [mobileplatforms.Android]:
      comment: Android platform
    ↓iOS [mobileplatforms.IOs]:
      comment: iOS platform
    ↓NgxApp [ngx.components.ApplicationComponent]: ? mobileNgxApp.yaml
```



These icons represent **features** allowing you to **manipulate the files** : navigate to the next or previous changes, swap the views, copy changes from one view to the other...

7.4 Compare Mode

Right-click twice on a file in the Staging view to open the compare mode.

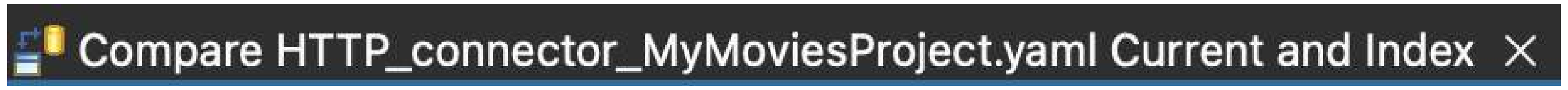


> MyMoviesProject [master]

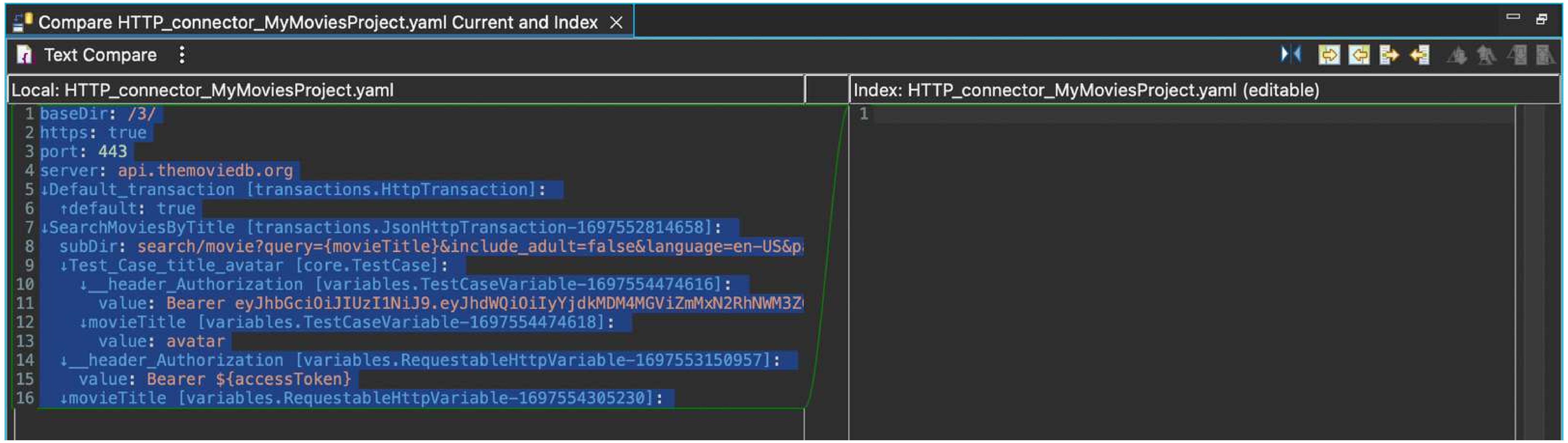
Unstaged Changes (4)

- > c8oProject.yaml
- HTTP_connector_MyMoviesProject.yaml - _c8oProject/connectors
- project.md
- SearchMoviesByTitle.yaml - _c8oProject/sequences

HTTP_connector_MyMoviesProject.yaml - _c8oProject/connectors



Compare HTTP_connector_MyMoviesProject.yaml Current and Index



Compare HTTP_connector_MyMoviesProject.yaml Current and Index

Text Compare

Local: HTTP_connector_MyMoviesProject.yaml

Index: HTTP_connector_MyMoviesProject.yaml (editable)

```
baseDir: /3/
https: true
port: 443
server: api.themoviedb.org
↓Default_transaction [transactions.HttpTransaction]:
  ↑default: true
↓SearchMoviesByTitle [transactions.JsonHttpTransaction-1697552814658]:
  subDir: search/movie?query={movieTitle}&include_adult=false&language=en-US&p
  ↓Test_Case_title_avatar [core.TestCase]:
    ↓__header_Authorization [variables.TestCaseVariable-1697554474616]:
      value: Bearer eyJhbGciOiJIUzI1NiJ9.eyJhdWQiOiIyYjdkMDM4MGViZmMxN2RhNWM3Z
    ↓movieTitle [variables.TestCaseVariable-1697554474618]:
      value: avatar
  ↓__header_Authorization [variables.RequestableHttpVariable-1697553150957]:
    value: Bearer ${accessToken}
  ↓movieTitle [variables.RequestableHttpVariable-1697554305230]:
```

You can see the changes since the last commit :

Here the **index view** is empty because the connector was created after the last commit

7.5 Create a repository

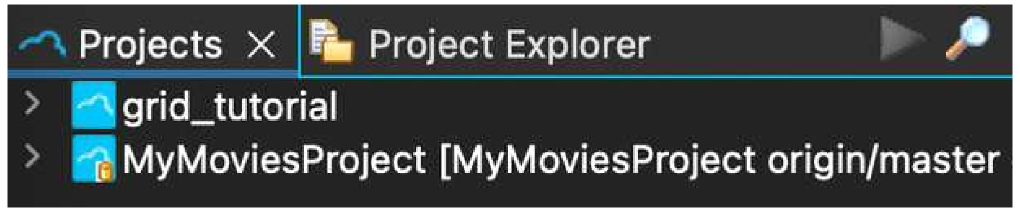
convertigo

When you **create a new project** in your workspace, a **Git Repository is automatically created**.

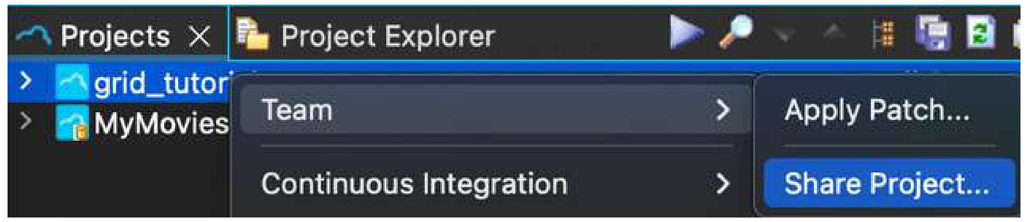
But if you **import a project from a .car file**, you have to **create it manually**.

Let’s say we want to create a Git Repository for the the project grid_tutorial

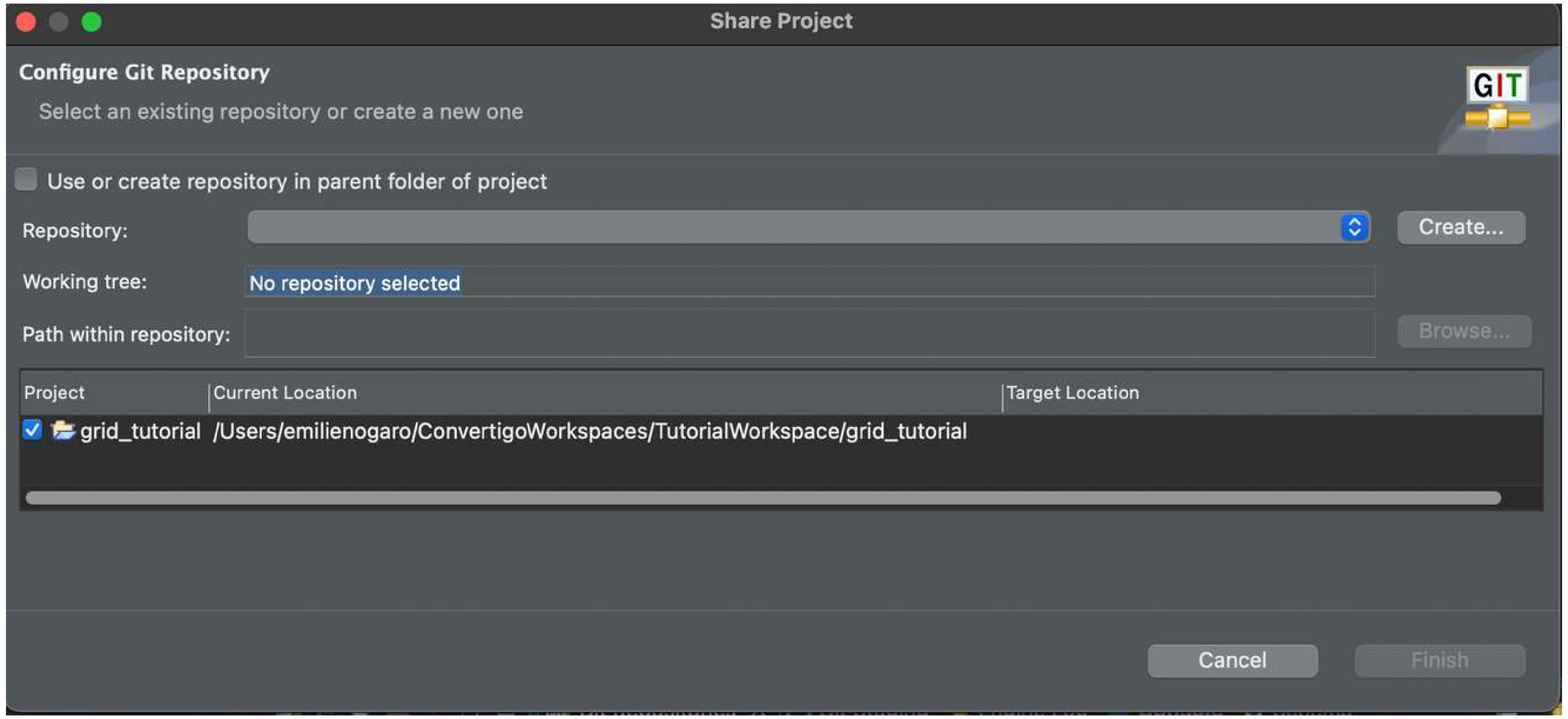
Right-click on the project grid_tutorial in the Projects view.



Select Team > then Share Project



In the Configure Git Repository window of the Share Project Window, click on Create to create the repository



Configure Git Repository

Select an existing repository or create a new one

☐ Use or create repository in parent folder of project

Repository:

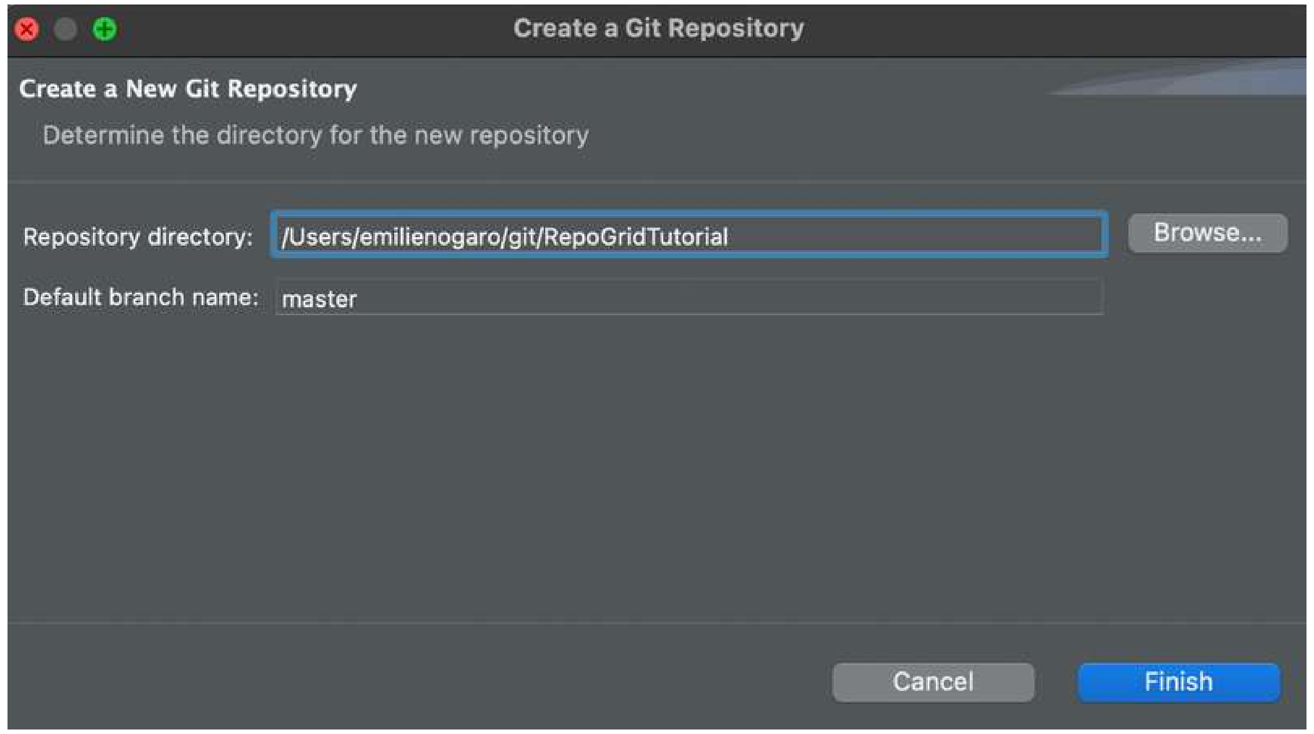
Working tree: No repository selected

Path within repository:

Project	Current Location	Target Location
☑ grid_tutorial	/Users/emilienogaro/ConvertigoWorkspaces/TutorialWorkspace/grid_tutorial	

7.5 Create a repository

In the **Create a new Git Repository window**, change the repository name (repository by default) to RepoGridTutorial. Then click on **Finish**.



Create a Git Repository

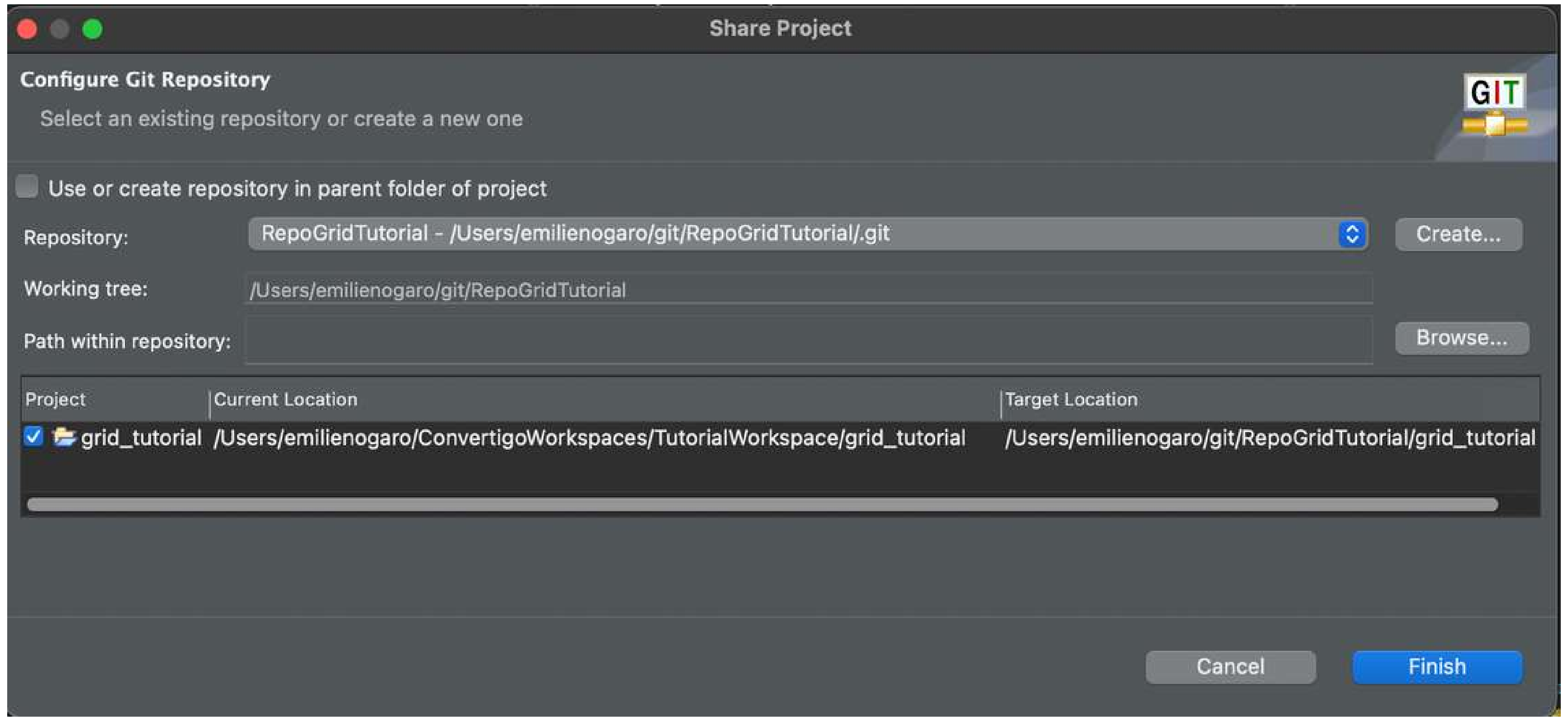
Create a New Git Repository

Determine the directory for the new repository

Repository directory: /Users/emilienogaro/git/RepoGridTutorial

Default branch name: master

In the **Configure Git Repository window**, you can see the **repository name and its path**. Click on **Finish**.



Share Project

Configure Git Repository

Select an existing repository or create a new one

☐ Use or create repository in parent folder of project

Repository: RepoGridTutorial - /Users/emilienogaro/git/RepoGridTutorial/.git

Working tree: /Users/emilienogaro/git/RepoGridTutorial

Path within repository:

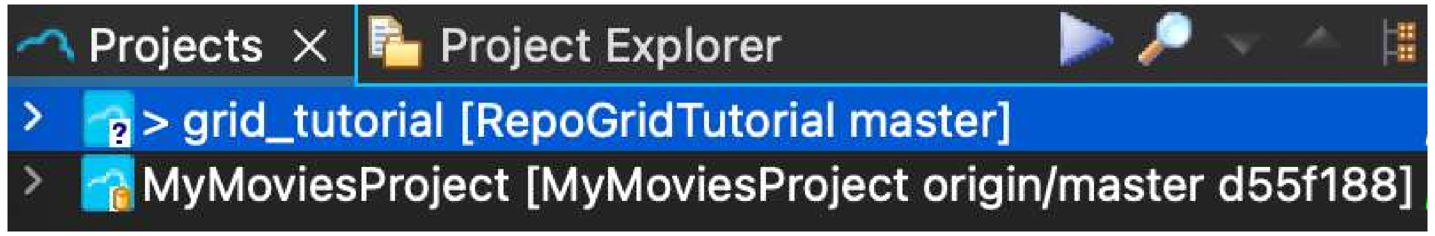
Project	Current Location	Target Location
☑ grid_tutorial	/Users/emilienogaro/ConvertigoWorkspaces/TutorialWorkspace/grid_tutorial	/Users/emilienogaro/git/RepoGridTutorial/grid_tutorial

7.5 Create a repository

In the **Projects view**,

the **repository name** and the **branch name**

appears after the project name.

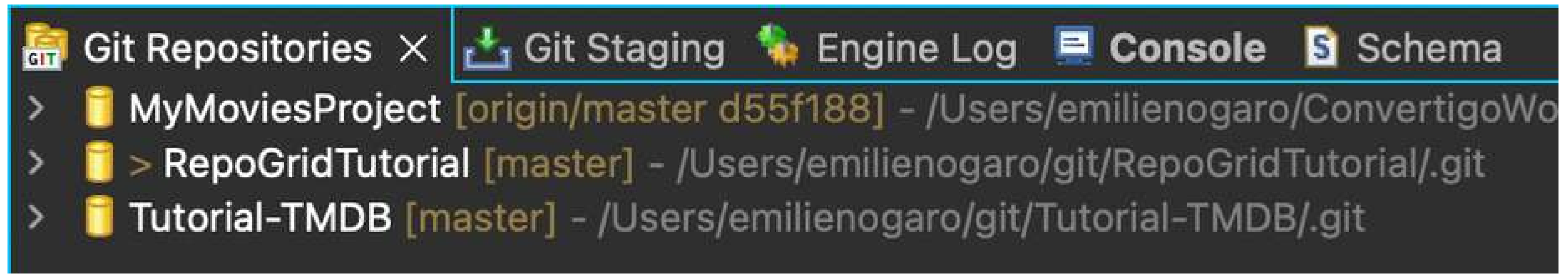


> grid_tutorial [RepoGridTutorial master]
MyMoviesProject [MyMoviesProject origin/master d55f188]

In the **Git Repositories view**,

the **repository name**, the **branch name**

and the **path to the Git repository** appears.

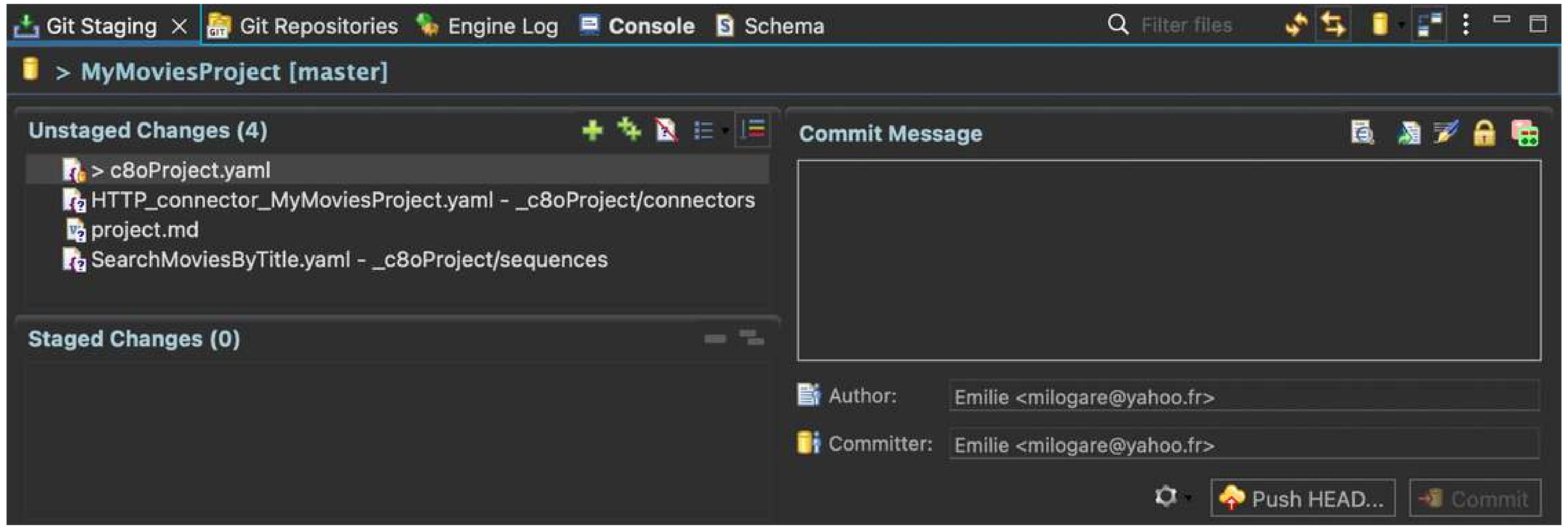


MyMoviesProject [origin/master d55f188] - /Users/emilienogaro/ConvertigoWo
> RepoGridTutorial [master] - /Users/emilienogaro/git/RepoGridTutorial/.git
Tutorial-TMDB [master] - /Users/emilienogaro/git/Tutorial-TMDB/.git

7.6 Commit your changes

Let's say you have **made a few changes** in your project and you want to **commit them on a Git repository**.

Go the Git Staging view and stage your files.



> MyMoviesProject [master]

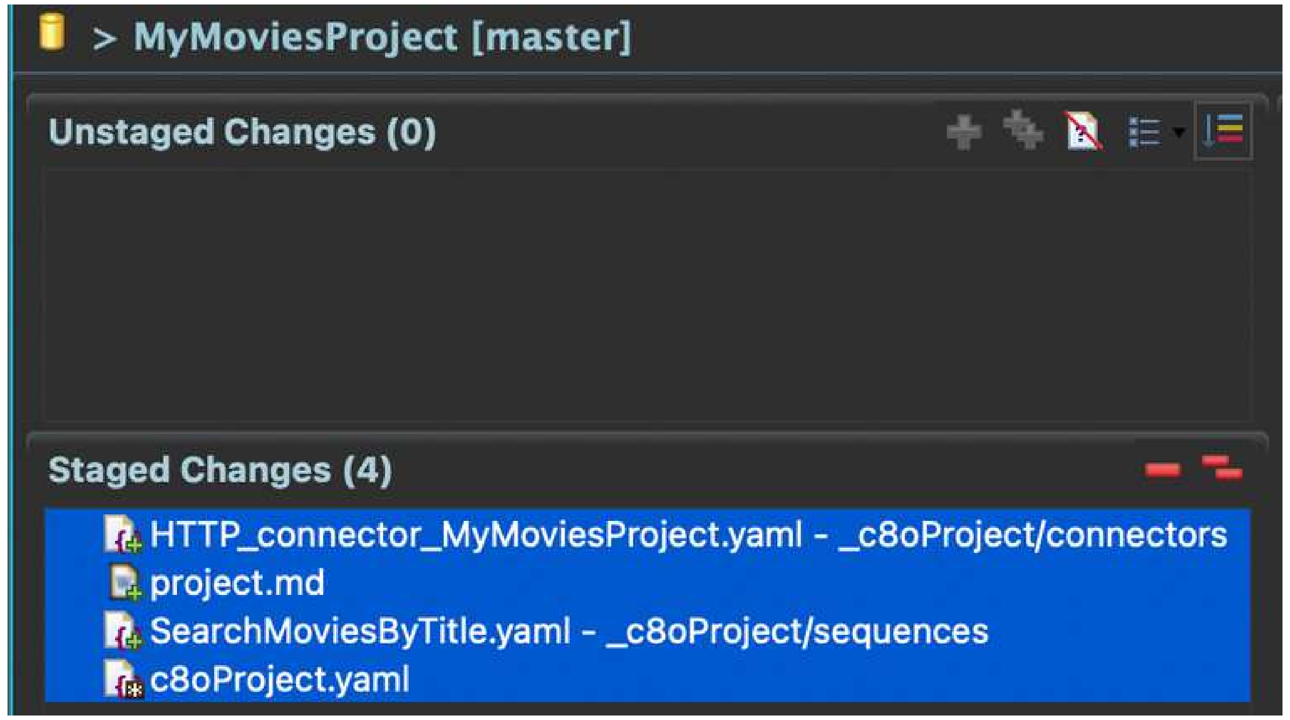
Unstaged Changes (4)
- > c8oProject.yaml
- HTTP_connector_MyMoviesProject.yaml - _c8oProject/connectors
- project.md
- SearchMoviesByTitle.yaml - _c8oProject/sequences

Staged Changes (0)

Commit Message

Author: Emilie <milogare@yahoo.fr>

Committer: Emilie <milogare@yahoo.fr>



> MyMoviesProject [master]

Unstaged Changes (0)

Staged Changes (4)
- HTTP_connector_MyMoviesProject.yaml - _c8oProject/connectors
- project.md
- SearchMoviesByTitle.yaml - _c8oProject/sequences
- c8oProject.yaml

Stage your files with the green cross

✚ One by One

✚✚ Or All at Once

You can also Unstage them with the red line

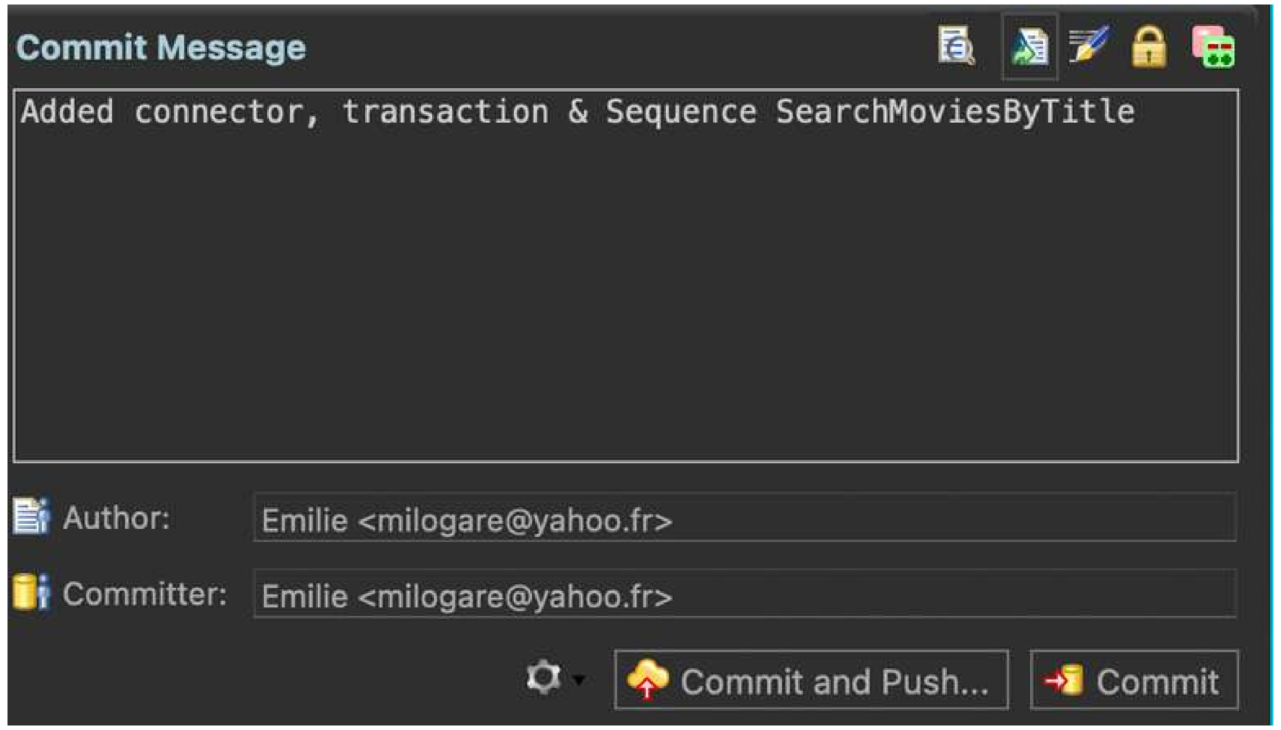
▬ One by One

▬▬ Or All at Once

7.6 Commit your changes

Add a commit message

and click on **Commit**.



Commit Message

```
Added connector, transaction & Sequence SearchMoviesByTitle
```

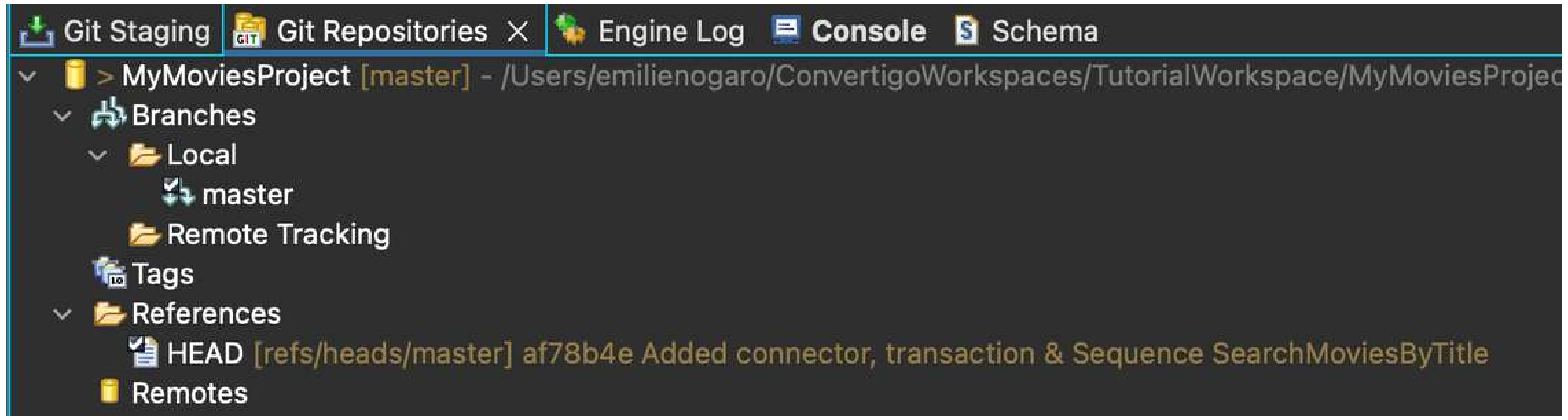
Author: Emilie <milogare@yahoo.fr>

Committer: Emilie <milogare@yahoo.fr>

Your changes have been **committed to your local Git repository**

In the **Git Repositories view**,

you can see the **latest commit** in the **References folder**.



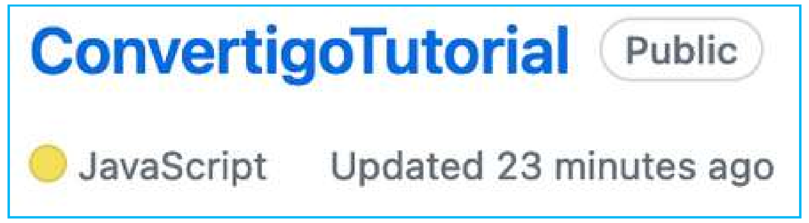
```
> MyMoviesProject [master] - /Users/emilienogaro/ConvertigoWorkspaces/TutorialWorkspace/MyMoviesProjec
  Branches
    Local
      master
    Remote Tracking
  Tags
  References
    HEAD [refs/heads/master] af78b4e Added connector, transaction & Sequence SearchMoviesByTitle
  Remotes
```

7.6 Commit your changes

At this stage, only your **local Git repository** has been initialized.

Let’s add a **remote repository** to your project.

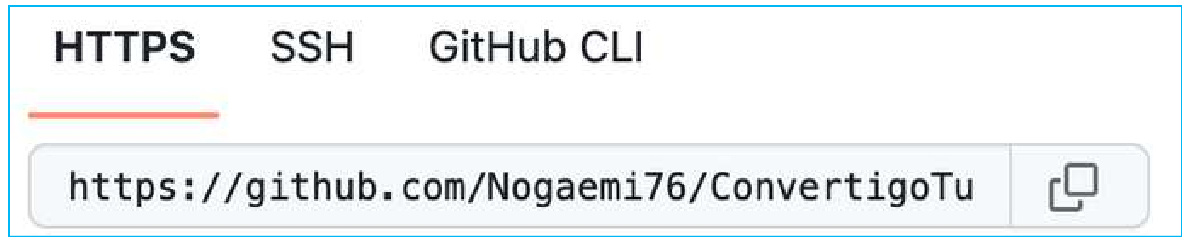
Create an **empty remote repository in GitHub or GitLab.**



ConvertigoTutorial Public

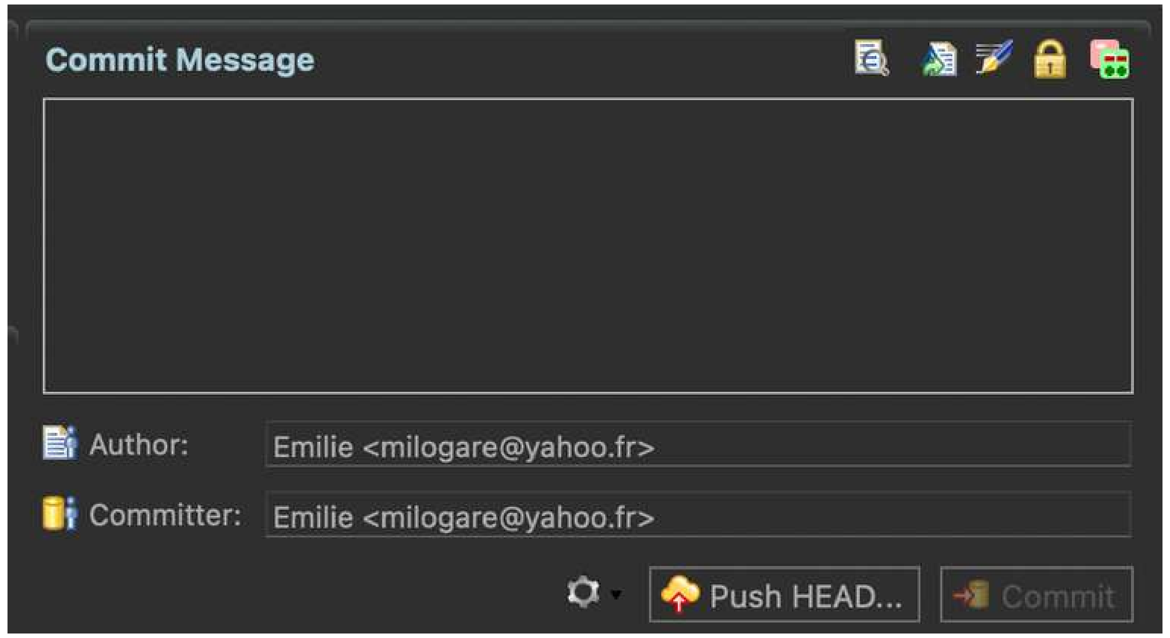
● JavaScript Updated 23 minutes ago

Copy your repo URI to the clipboard.



```
https://github.com/Nogaemi76/ConvertigoTu
```

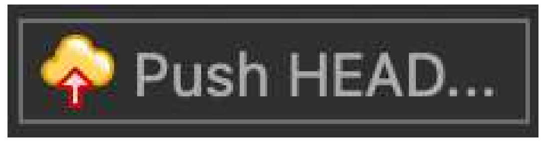
In the **Git Staging view**



Author: Emilie <milogare@yahoo.fr>

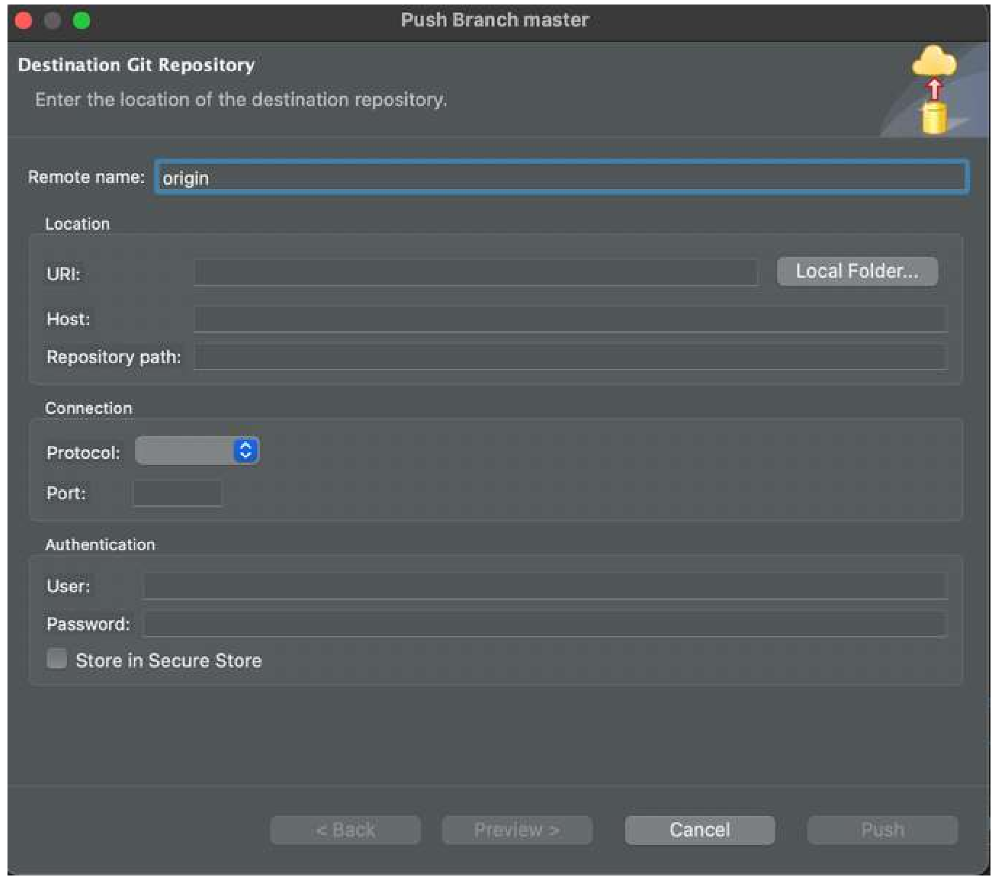
Committer: Emilie <milogare@yahoo.fr>

Click on **Push HEAD**.



7.6 Commit your changes

In the **Destination Git Repository window** of the **Push Branch Master window**, paste the **URI in the URI field**.



Push Branch master

Destination Git Repository

Enter the location of the destination repository.

Remote name: origin

Location

URI:

Host:

Repository path:

Connection

Protocol:

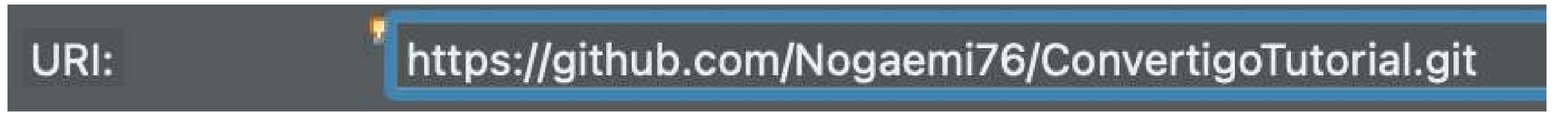
Port:

Authentication

User:

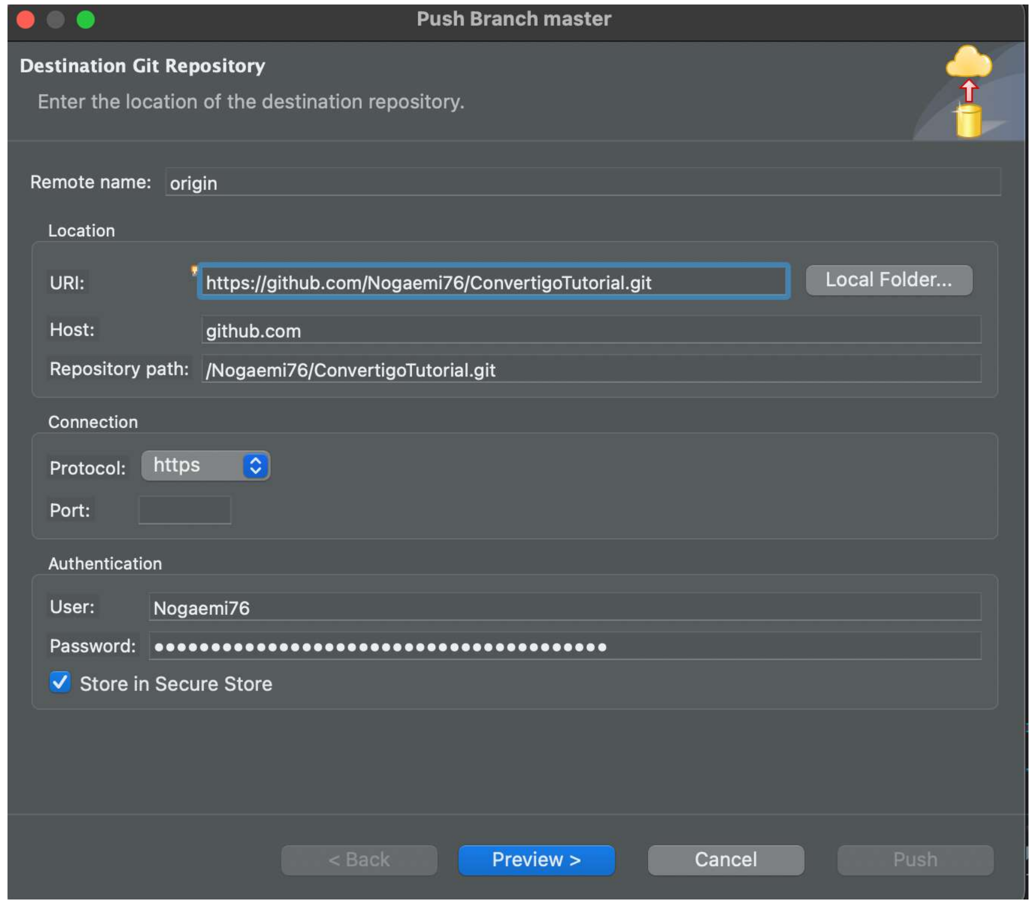
Password:

□ Store in Secure Store



URI: https://github.com/Nogaemi76/ConvertigoTutorial.git

The other fields will update automatically



Push Branch master

Destination Git Repository

Enter the location of the destination repository.

Remote name: origin

Location

URI: https://github.com/Nogaemi76/ConvertigoTutorial.git

Host: github.com

Repository path: /Nogaemi76/ConvertigoTutorial.git

Connection

Protocol: https

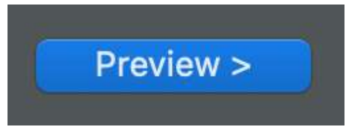
Port:

Authentication

User: Nogaemi76

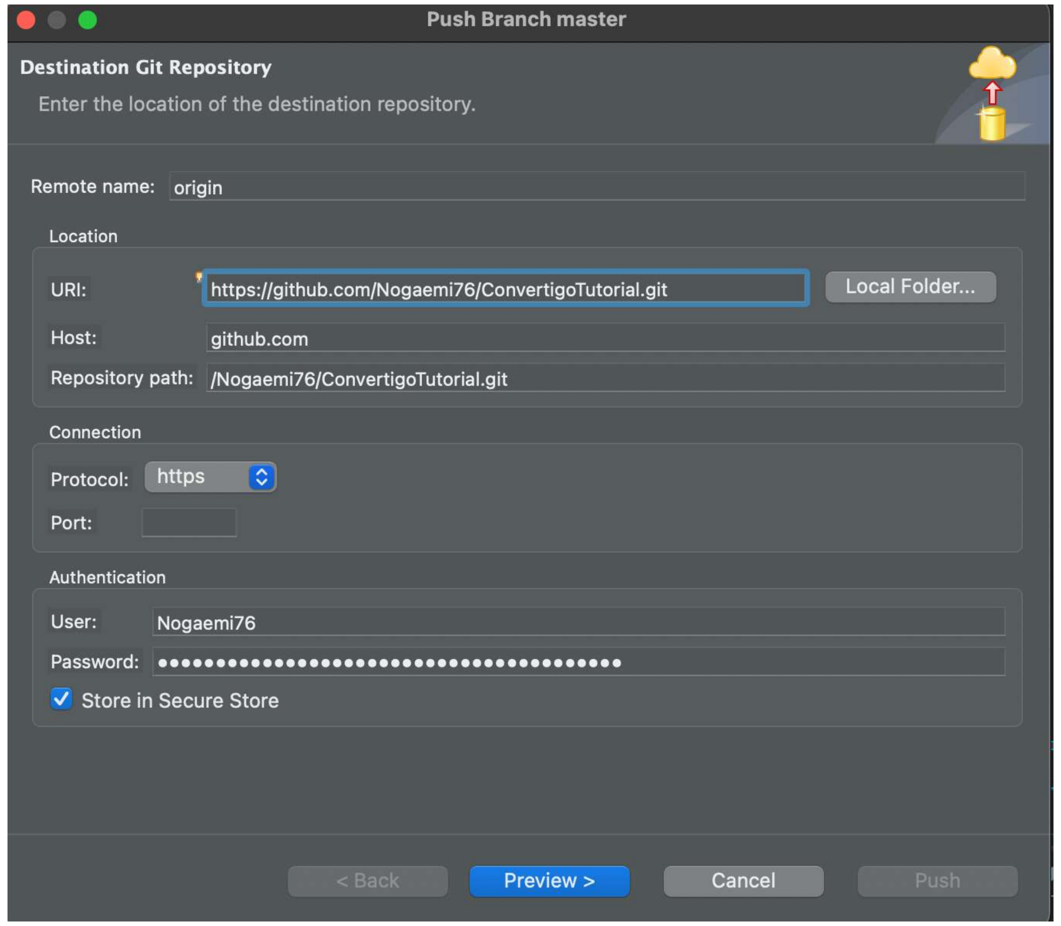
Password: ●●

☑ Store in Secure Store



Click on **Preview>**.

7.6 Commit your changes



Push Branch master

Destination Git Repository

Enter the location of the destination repository.

Remote name: origin

Location

URI: https://github.com/Nogaemi76/ConvertigoTutorial.git

Host: github.com

Repository path: /Nogaemi76/ConvertigoTutorial.git

Connection

Protocol: https

Port:

Authentication

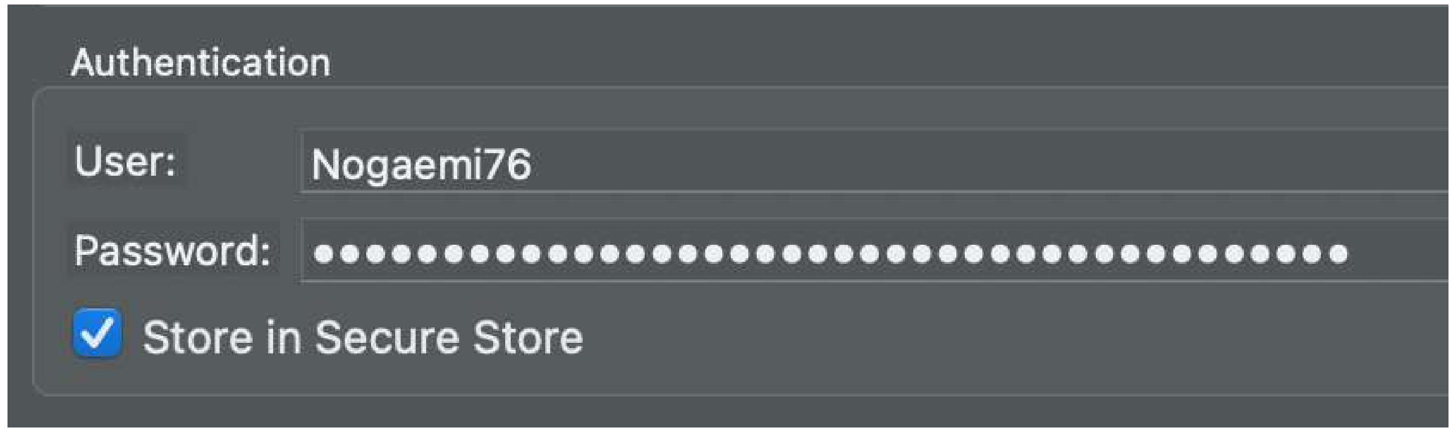
User: Nogaemi76

Password: ••

☑ Store in Secure Store

Reminder : In the Authentication part of the Push Branch Master window.

- User is your GitHub Username
- Password is a Personal access token from GitHub

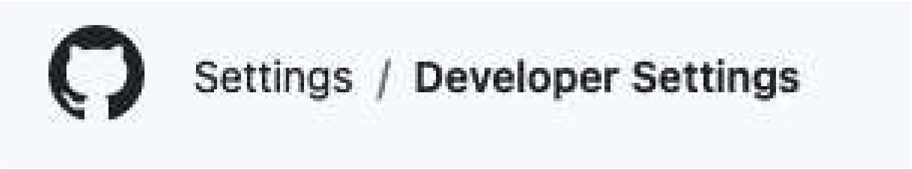


Authentication

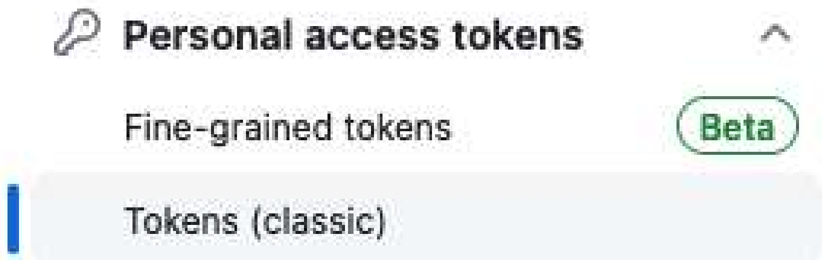
User: Nogaemi76

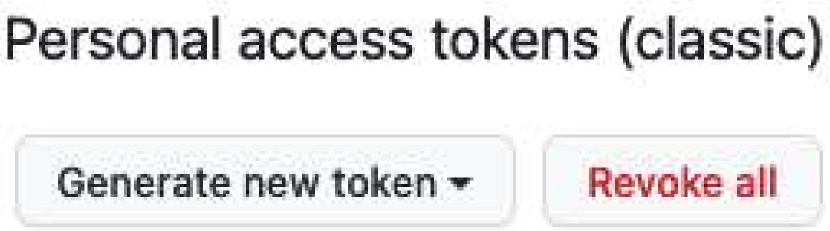
Password: ••

☑ Store in Secure Store



Settings / **Developer Settings**





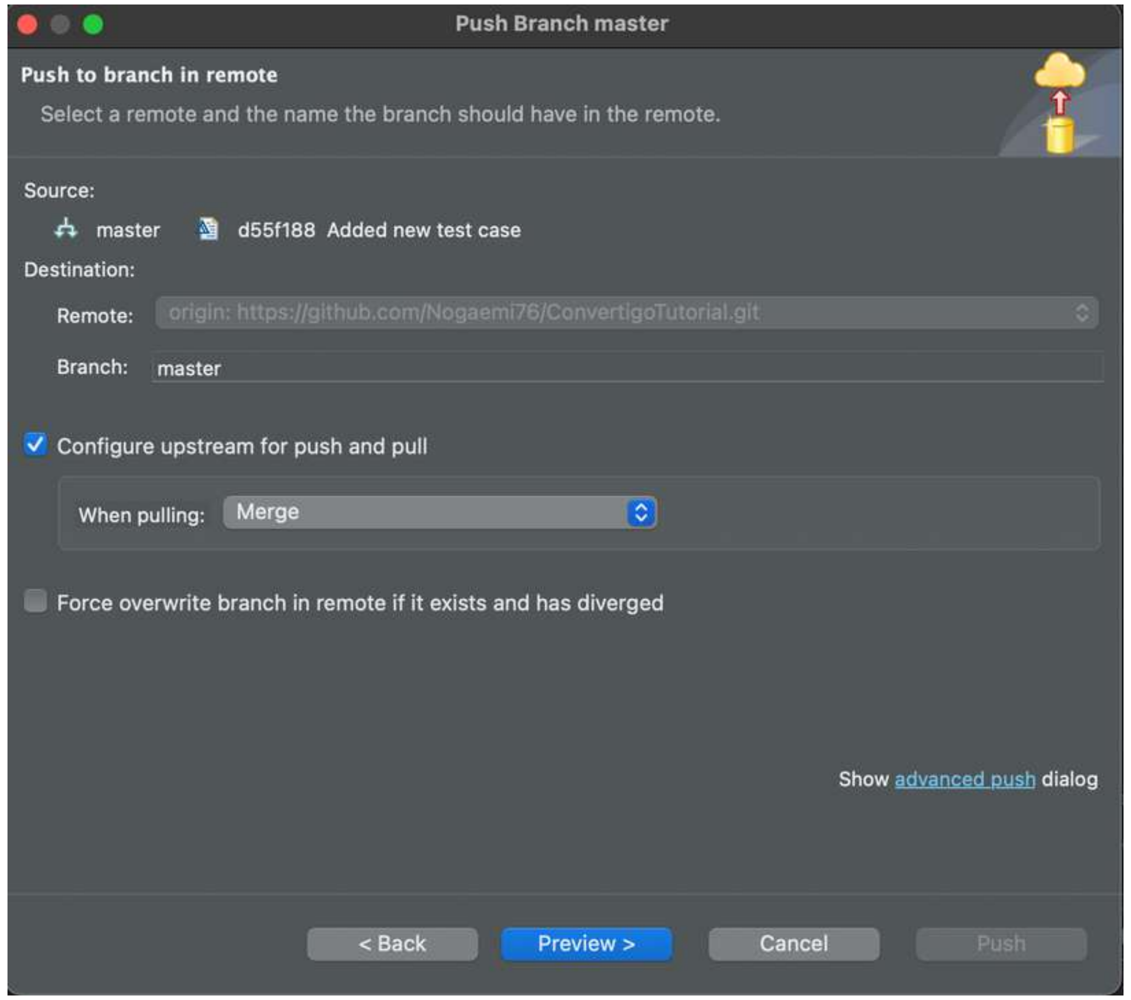
Personal access tokens (classic)

7.6 Commit your changes

The **Push to branch in remote** window appears.

You can change the remote branch if necessary.

Click on **Preview >**.



Push Branch master

Push to branch in remote

Select a remote and the name the branch should have in the remote.

Source:

master d55f188 Added new test case

Destination:

Remote: origin: https://github.com/Nogaemi76/ConvertigoTutorial.git

Branch: master

☑ Configure upstream for push and pull

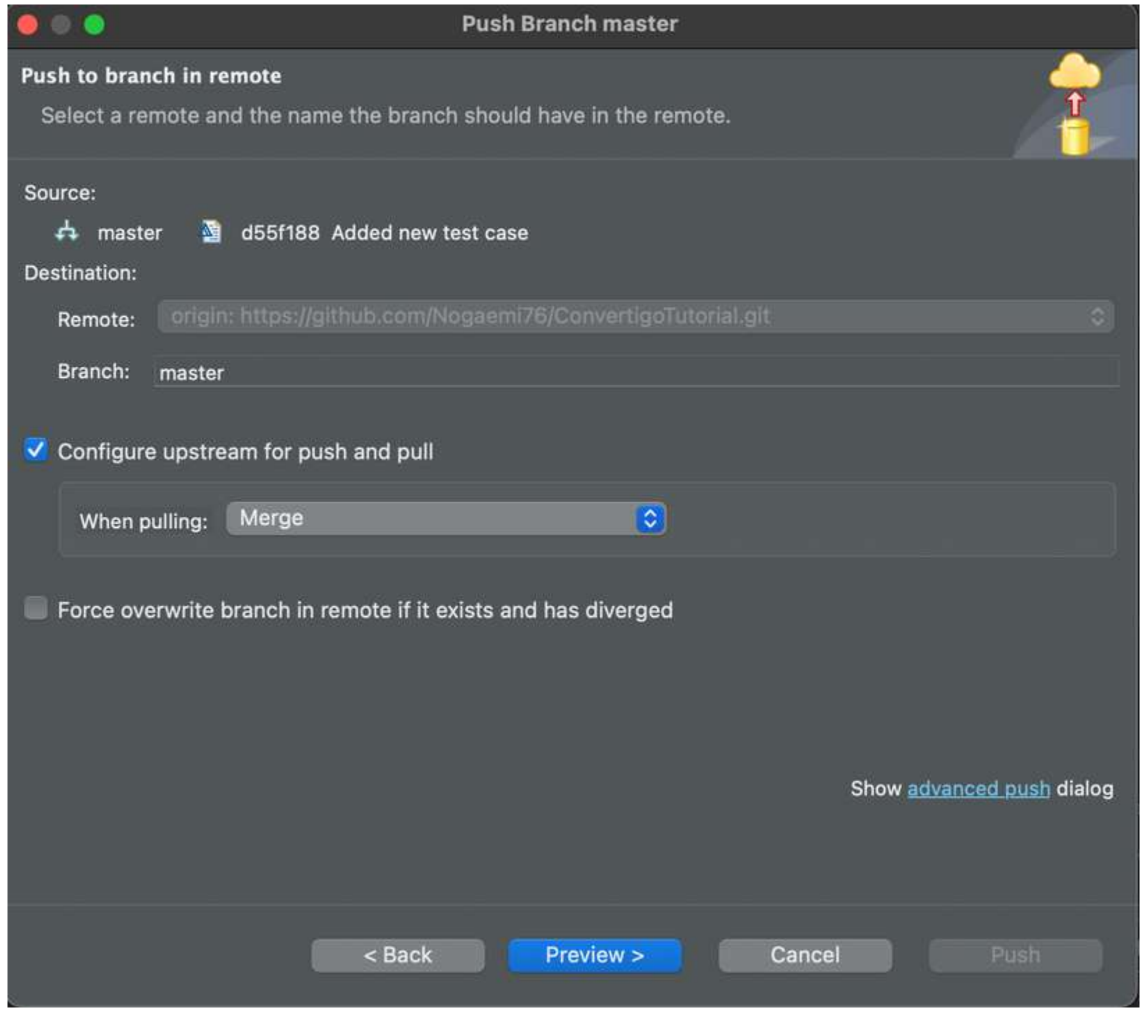
When pulling: Merge

□ Force overwrite branch in remote if it exists and has diverged

Show advanced push dialog

The **Push to branch in remote window** appears.

Click on **Preview >**



Push Branch master

Push to branch in remote

Select a remote and the name the branch should have in the remote.

Source:

master d55f188 Added new test case

Destination:

Remote: origin: https://github.com/Nogaemi76/ConvertigoTutorial.git

Branch: master

☑ Configure upstream for push and pull

When pulling: Merge

□ Force overwrite branch in remote if it exists and has diverged

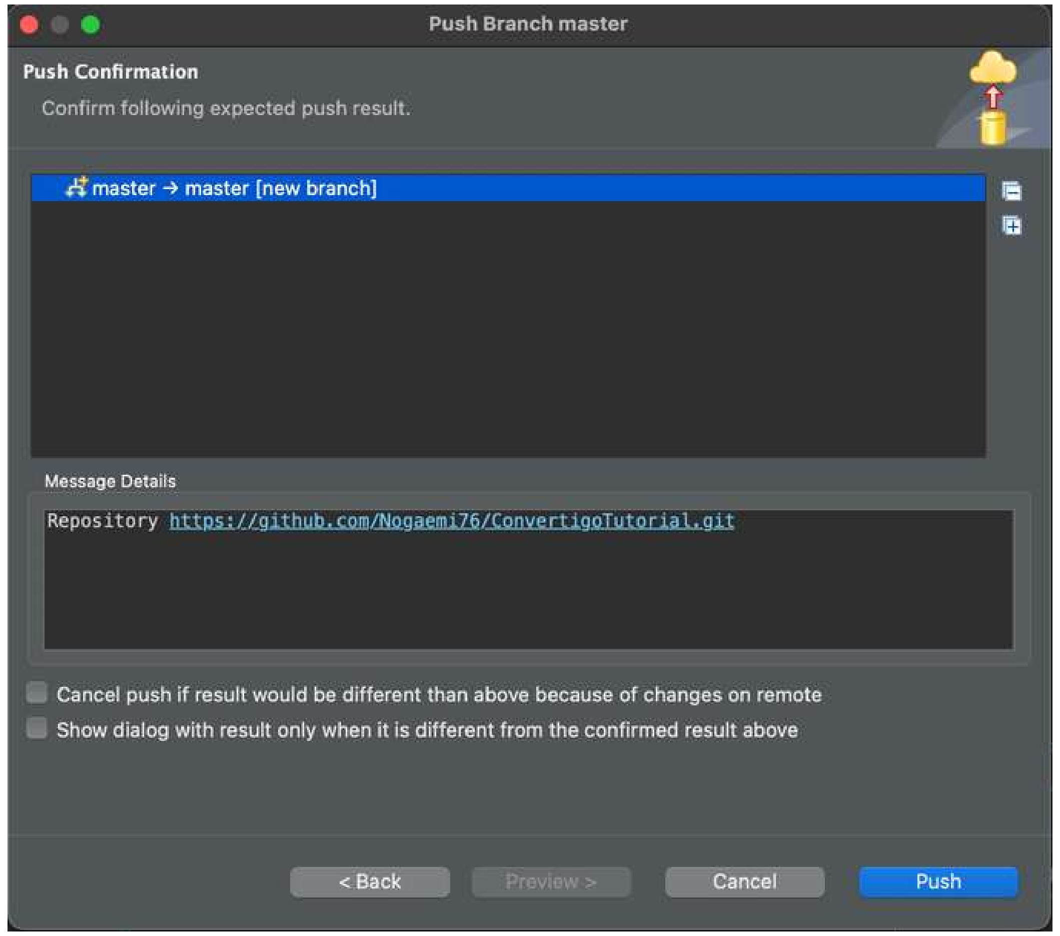
Show advanced push dialog

7.6 Commit your changes

The **Push Confirmation window** appears.

Click on **Push**

to **push your project on your remote repository**.



Push Branch master

Push Confirmation

Confirm following expected push result.

master → master [new branch]

Message Details

```
Repository https://github.com/Nogaemi76/ConvertigoTutorial.git
```

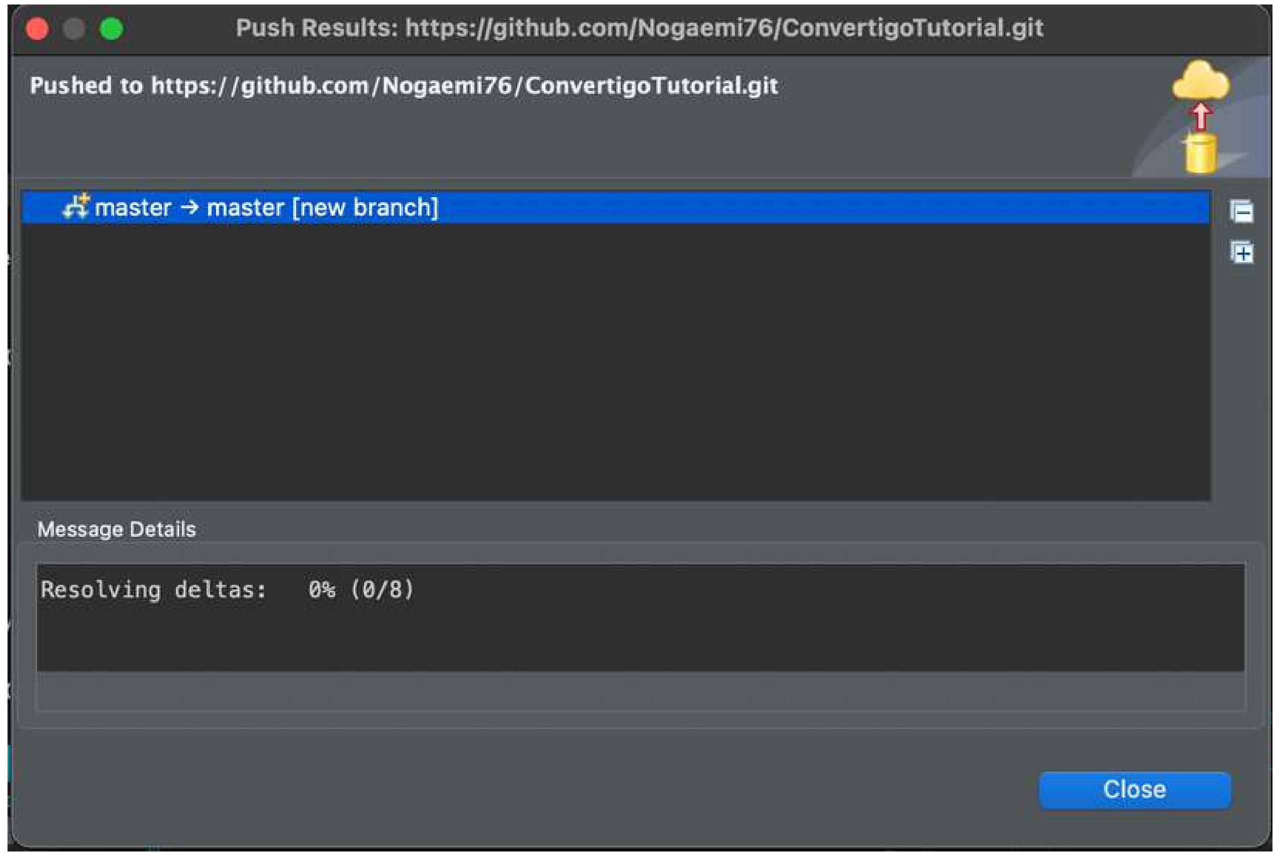
□ Cancel push if result would be different than above because of changes on remote

□ Show dialog with result only when it is different from the confirmed result above

A **Push Results window** appears

to confirm that your project has been pushed

on your remote repository.



Push Results: https://github.com/Nogaemi76/ConvertigoTutorial.git

Pushed to https://github.com/Nogaemi76/ConvertigoTutorial.git

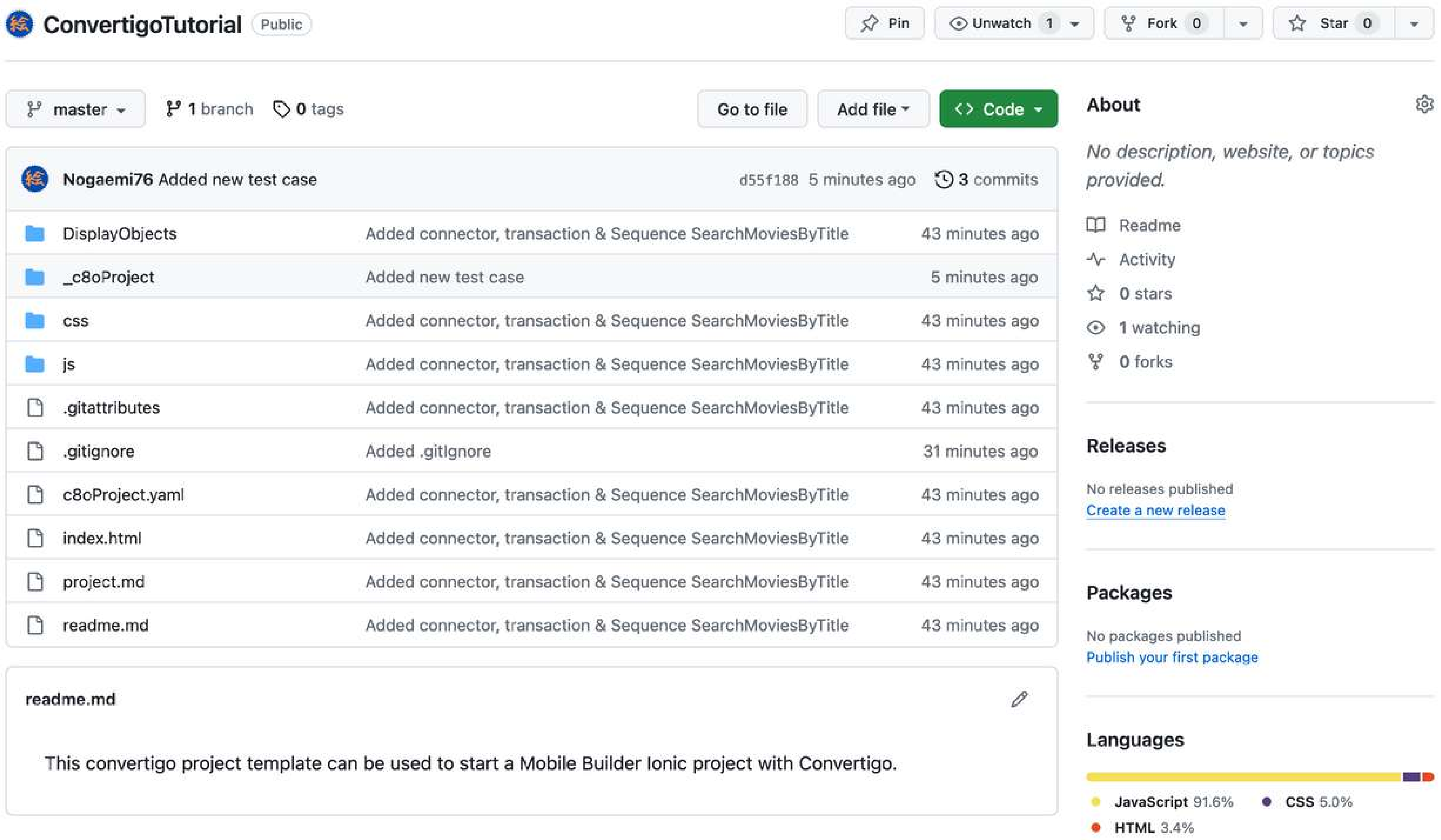
master → master [new branch]

Message Details

```
Resolving deltas:   0% (0/8)
```

7.6 Commit your changes

Your project appears in your **remote repository**.



ConvertigoTutorial Public

Pin · Unwatch 1 · Fork 0 · Star 0

master · 1 branch · 0 tags · Go to file · Add file · Code

Nogaemi76 Added new test case · d55f188 5 minutes ago · 3 commits

DisplayObjects	Added connector, transaction & Sequence SearchMoviesByTitle	43 minutes ago
_c8oProject	Added new test case	5 minutes ago
css	Added connector, transaction & Sequence SearchMoviesByTitle	43 minutes ago
js	Added connector, transaction & Sequence SearchMoviesByTitle	43 minutes ago
.gitattributes	Added connector, transaction & Sequence SearchMoviesByTitle	43 minutes ago
.gitignore	Added .gitIgnore	31 minutes ago
c8oProject.yaml	Added connector, transaction & Sequence SearchMoviesByTitle	43 minutes ago
index.html	Added connector, transaction & Sequence SearchMoviesByTitle	43 minutes ago
project.md	Added connector, transaction & Sequence SearchMoviesByTitle	43 minutes ago
readme.md	Added connector, transaction & Sequence SearchMoviesByTitle	43 minutes ago

readme.md

This convertigo project template can be used to start a Mobile Builder Ionic project with Convertigo.

About

No description, website, or topics provided.

Readme

Activity

0 stars

1 watching

0 forks

Releases

No releases published

Create a new release

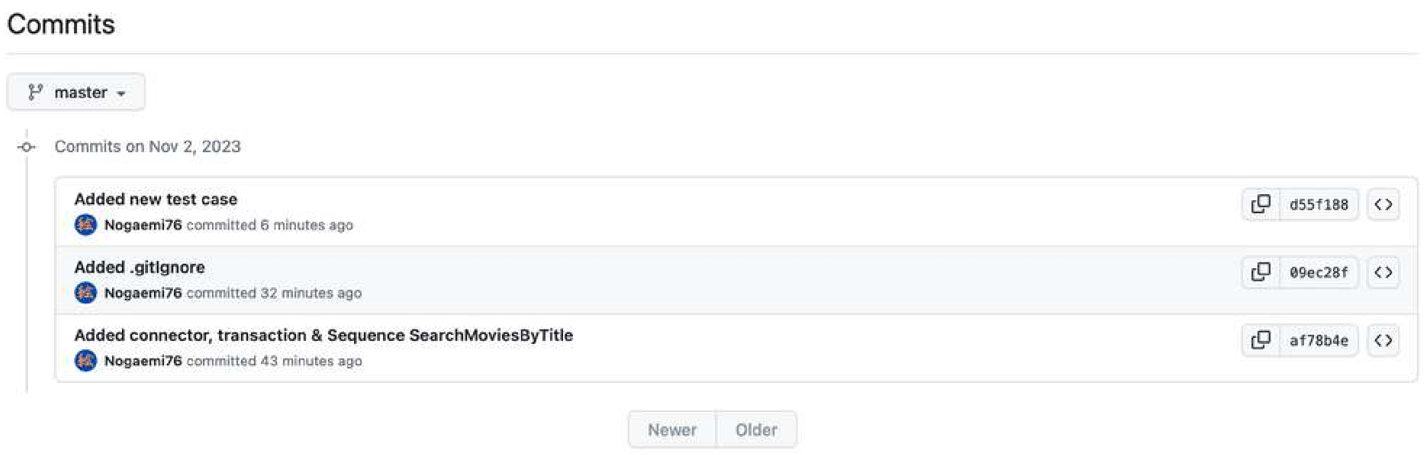
Packages

No packages published

Publish your first package

Languages

JavaScript 91.6% · CSS 5.0% · HTML 3.4%



Commits

master

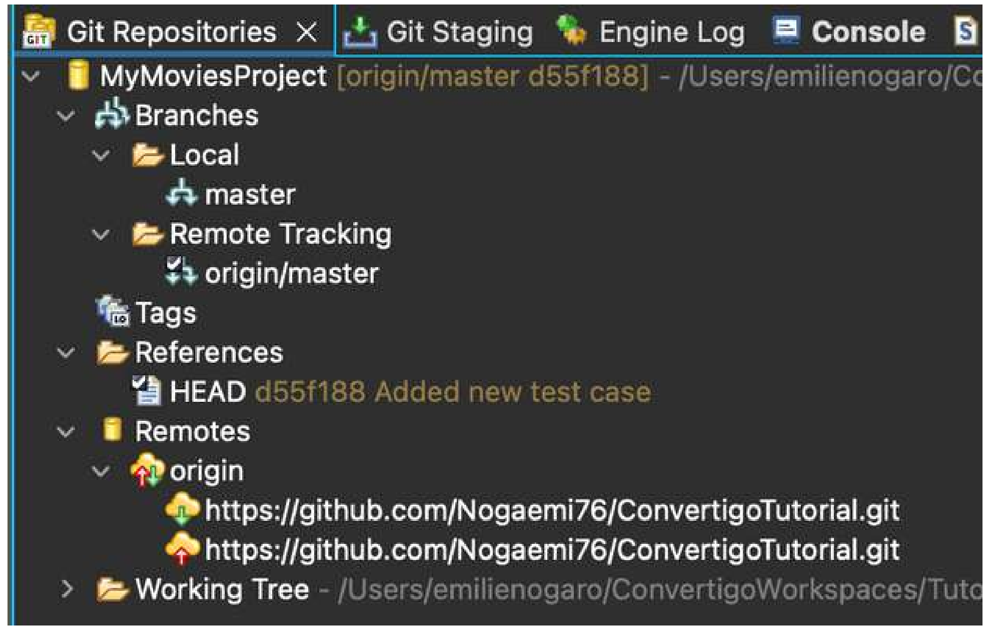
Commits on Nov 2, 2023

Added new test case Nogaemi76 committed 6 minutes ago	d55f188
Added .gitIgnore Nogaemi76 committed 32 minutes ago	09ec28f
Added connector, transaction & Sequence SearchMoviesByTitle Nogaemi76 committed 43 minutes ago	af78b4e

Newer · Older

The **Remote branch** appears

in the **Git Repository view**.



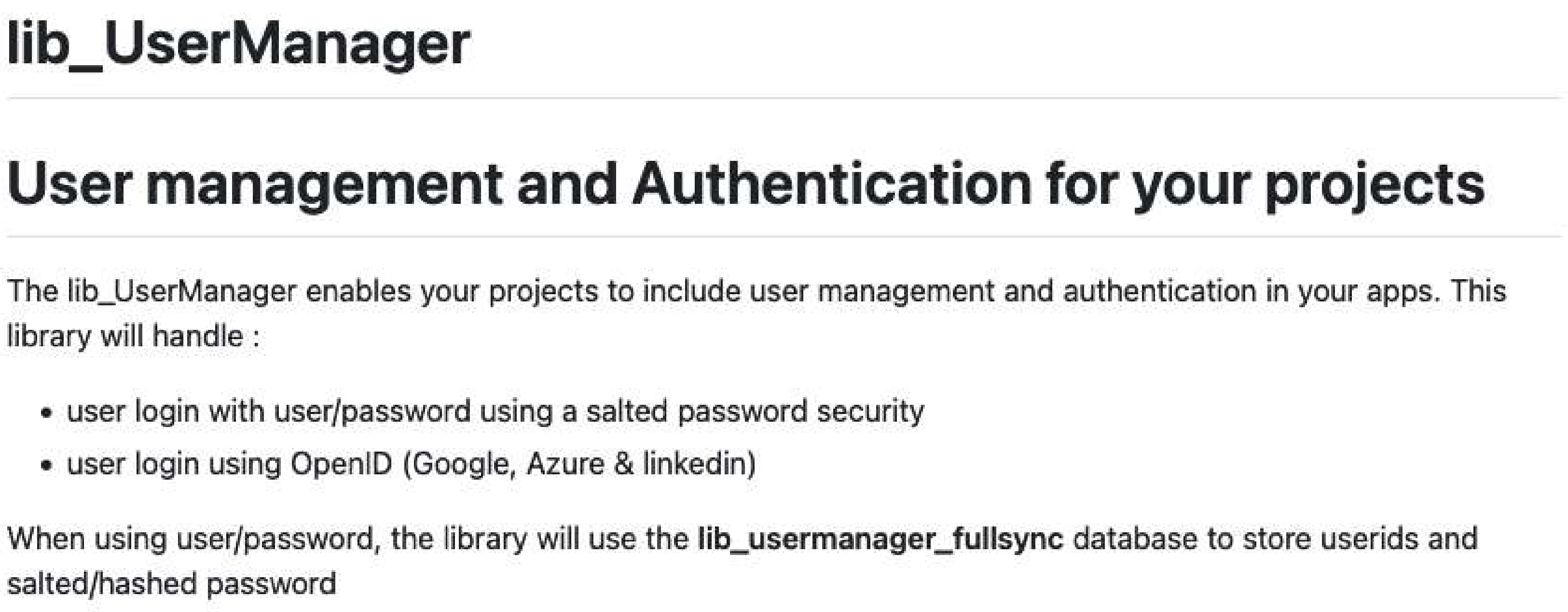
```
MyMoviesProject [origin/master d55f188] - /Users/emilienogaro/C
  Branches
    Local
      master
    Remote Tracking
      origin/master
  Tags
  References
    HEAD d55f188 Added new test case
  Remotes
    origin
      https://github.com/Nogaemi76/ConvertigoTutorial.git
      https://github.com/Nogaemi76/ConvertigoTutorial.git
  Working Tree - /Users/emilienogaro/ConvertigoWorkspaces/Tuto
```

7.7 Clone a project

Let’s say you want to clone a project in your studio.

For example, you want to use the **library lib_UserManager developed by Convertigo**.

It is used to **include user management and authentication** in a **Convertigo project**.



lib_UserManager

User management and Authentication for your projects

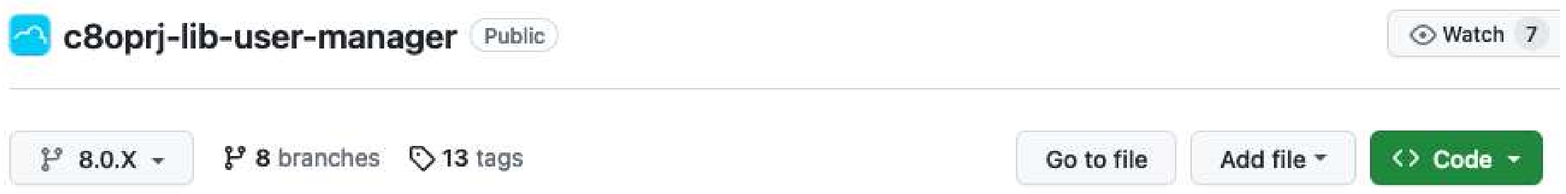
The lib_UserManager enables your projects to include user management and authentication in your apps. This library will handle :

- user login with user/password using a salted password security
- user login using OpenID (Google, Azure & linkedin)

When using user/password, the library will use the **lib_usermanager_fullsync** database to store userids and salted/hashed password

You can find the repository in GitHub :

https://github.com/convertigo/c8oprj-lib-user-manager



7.7 Clone a project

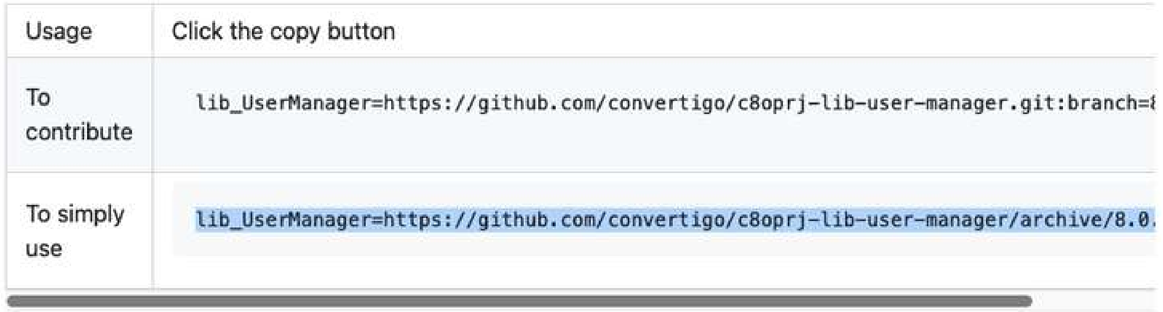
As explained in the ReadMe of lib_UserManager in GitHub,

the **simplest way** to clone a project is

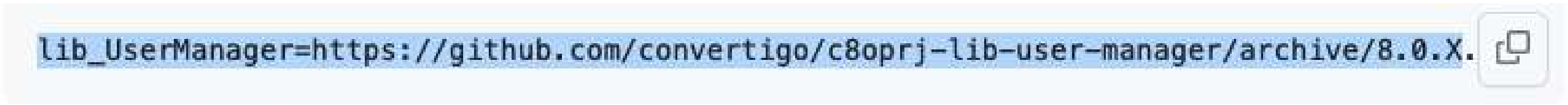
- NOT by using the Git Repositories view (more complex eclipse-based process).
- by **using the Convertigo project import Wizard** in the **Project view** (customized process developed by Convertigo).

Copy the project url from the ReadMe of the repo in GitHub :

lib_UserManager=https://github.com/convertigo/c8oprj-lib-user-manager/archive/8.0.X.zip

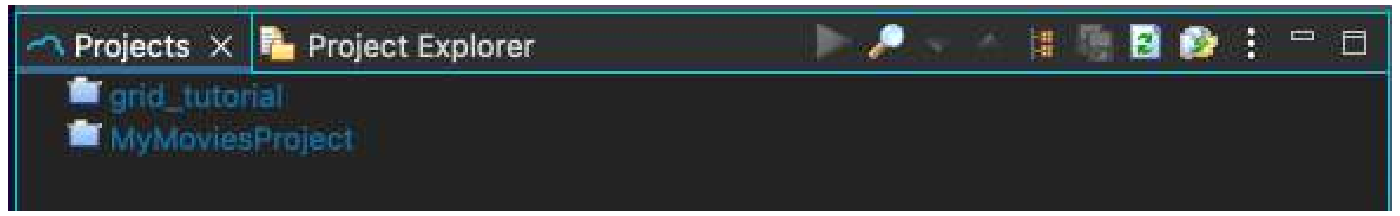


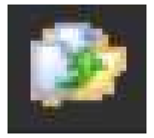
Usage	Click the copy button
To contribute	`lib_UserManager=https://github.com/convertigo/c8oprj-lib-user-manager.git:branch=`
To simply use	`lib_UserManager=https://github.com/convertigo/c8oprj-lib-user-manager/archive/8.0.`



```
lib_UserManager=https://github.com/convertigo/c8oprj-lib-user-manager/archive/8.0.X.
```

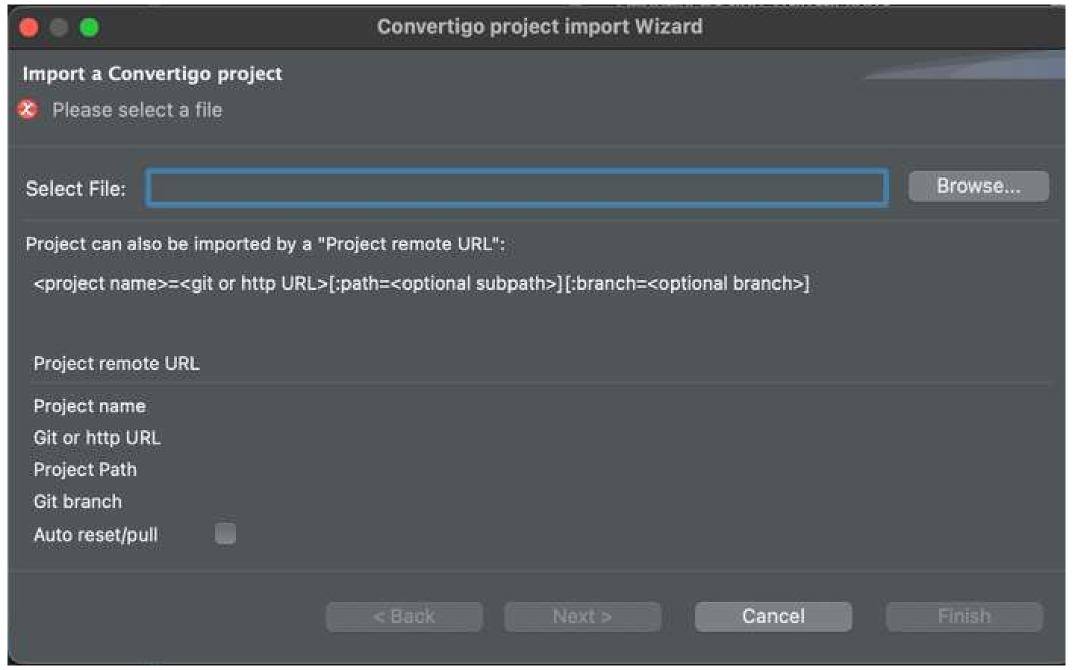
Click on the **Import a project in treeview** button to open the **Convertigo project import Wizard**.





7.7 Clone a project

The **Convertigo project import Wizard** opens.



Import a Convertigo project

Please select a file

Select File:

Project can also be imported by a "Project remote URL":

<project name>=<git or http URL>[:path=<optional subpath>][:branch=<optional branch>]

Project remote URL

Project name

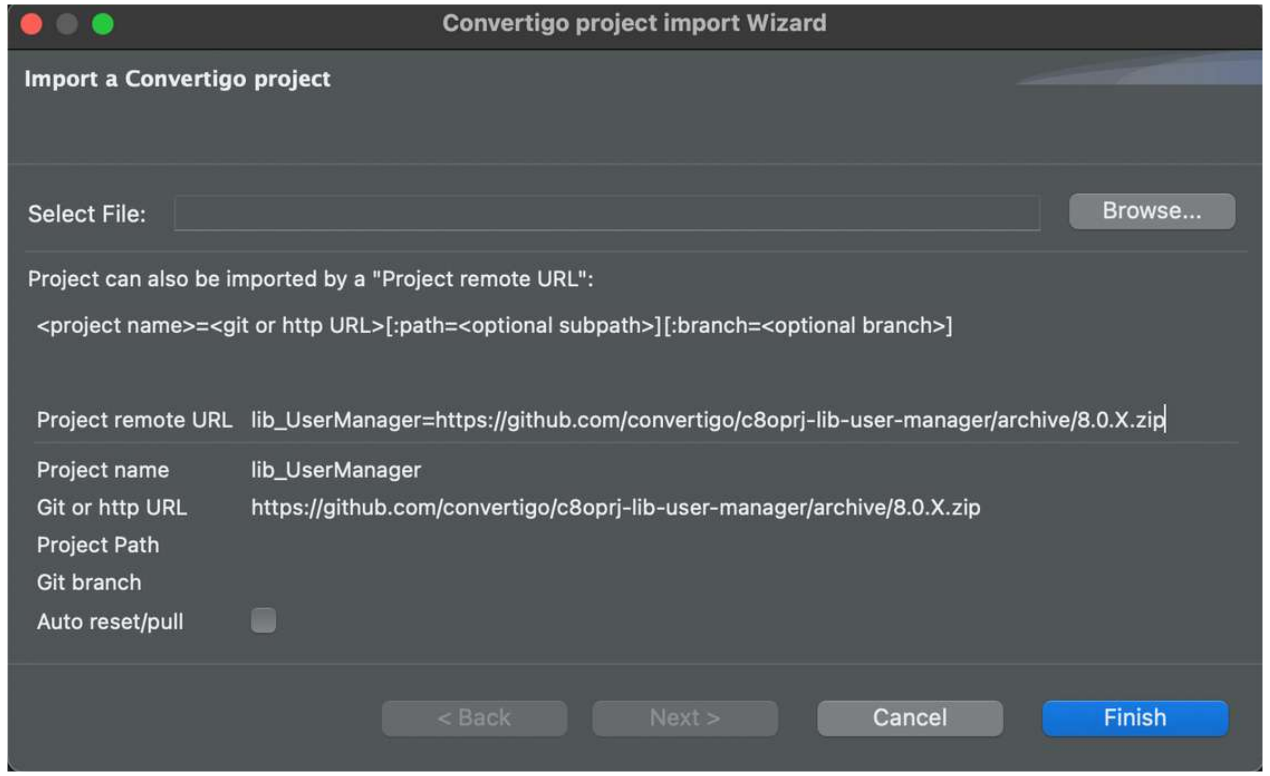
Git or http URL

Project Path

Git branch

Auto reset/pull □

Paste the **project url in the Project remote URL field** and click on **Finish**.



Import a Convertigo project

Select File:

Project can also be imported by a "Project remote URL":

<project name>=<git or http URL>[:path=<optional subpath>][:branch=<optional branch>]

Project remote URL lib_UserManager=https://github.com/convertigo/c8oprj-lib-user-manager/archive/8.0.X.zip

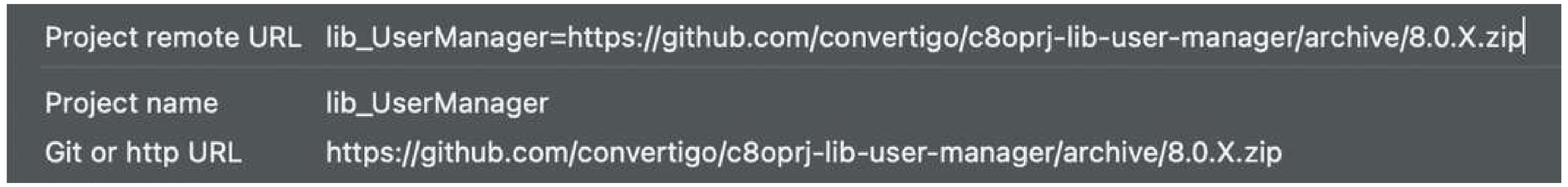
Project name lib_UserManager

Git or http URL https://github.com/convertigo/c8oprj-lib-user-manager/archive/8.0.X.zip

Project Path

Git branch

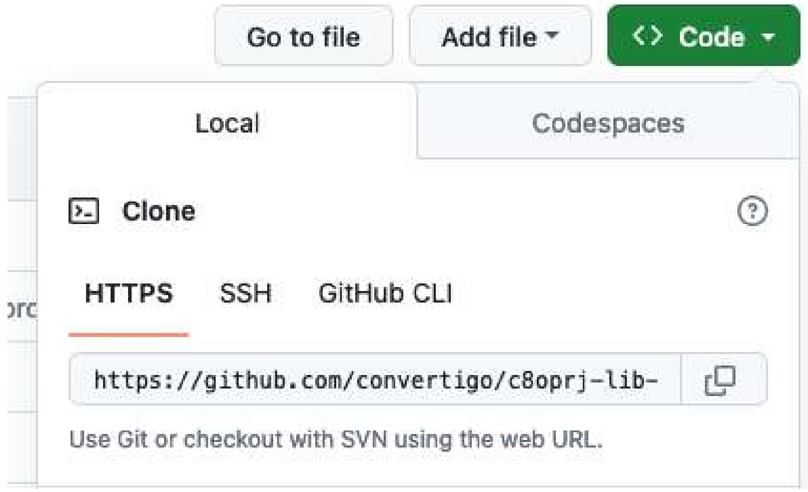
Auto reset/pull □



Project remote URL lib_UserManager=https://github.com/convertigo/c8oprj-lib-user-manager/archive/8.0.X.zip

Project name lib_UserManager

Git or http URL https://github.com/convertigo/c8oprj-lib-user-manager/archive/8.0.X.zip



https://github.com/convertigo/c8oprj-lib-

Use Git or checkout with SVN using the web URL.

Important :

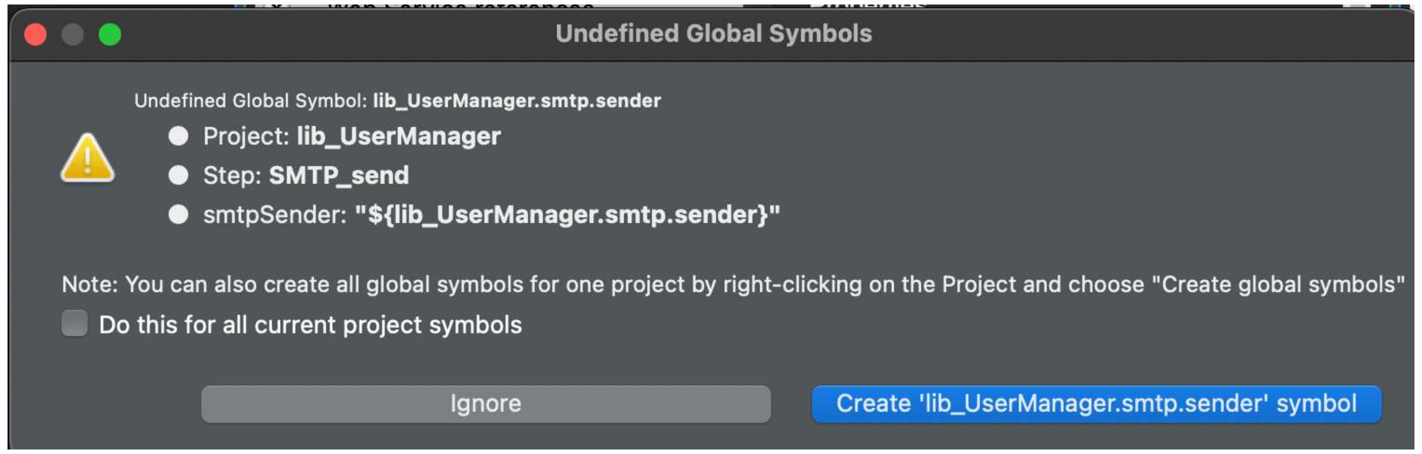
Usually, when **cloning a GitHub repo**, you copy it from the usual repo url and the project name is not already present in the url.

In that case, you have to **include it manually in the project name field**.

7.7 Clone a project

If the cloned project has symbols,

the **Undefined Global Symbol window** appears.



Undefined Global Symbols

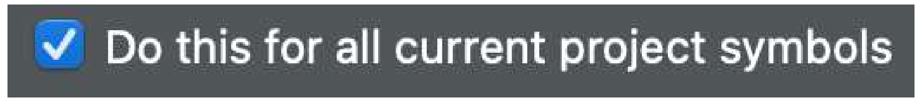
Undefined Global Symbol: **lib_UserManager.smtp.sender**

- Project: **lib_UserManager**
- Step: **SMTP_send**
- smtpSender: **"${lib_UserManager.smtp.sender}"**

Note: You can also create all global symbols for one project by right-clicking on the Project and choose "Create global symbols"

□ Do this for all current project symbols

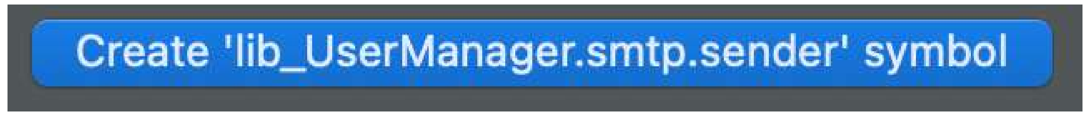
Select **Do this for all current symbols**.



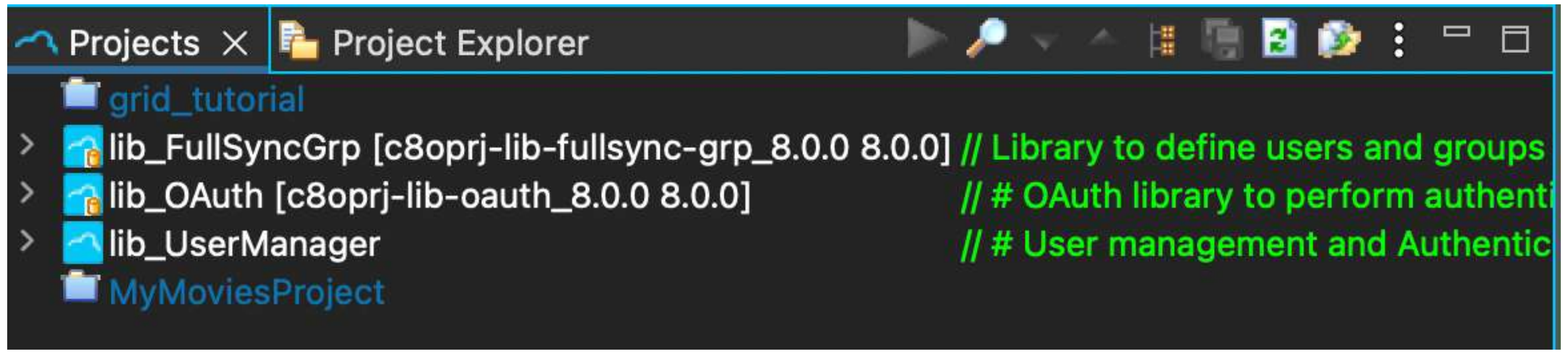
☑ Do this for all current project symbols

Click on Create 'XXX' symbol

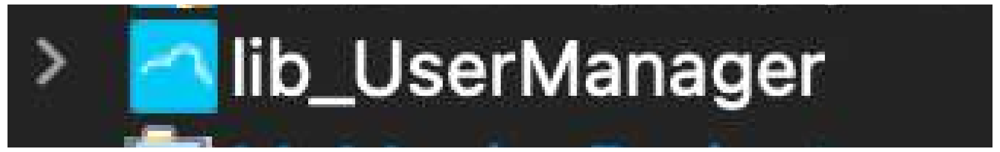
('XXX' depends on the symbol name).

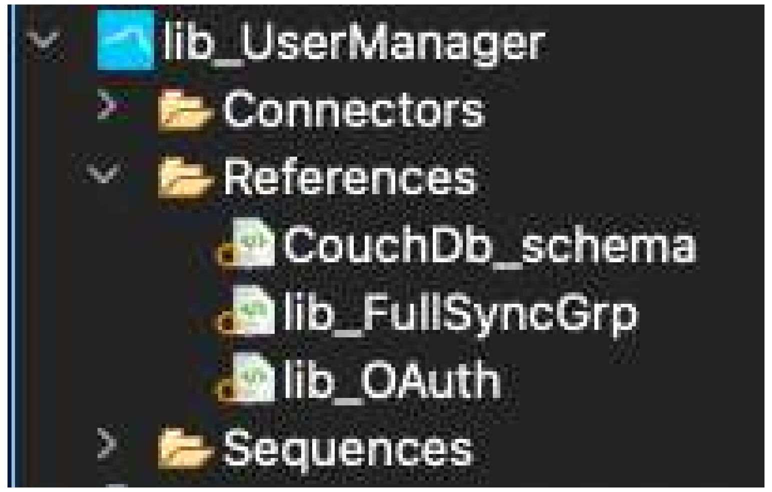


The **library is imported** and appears in the **Projects view**.



- grid_tutorial
- lib_FullSyncGrp [c8oprj-lib-fullsync-grp_8.0.0 8.0.0] // Library to define users and groups
- lib_OAuth [c8oprj-lib-oauth_8.0.0 8.0.0] // # OAuth library to perform authenti
- lib_UserManager // # User management and Authentic
- MyMoviesProject





- lib_UserManager
  - Connectors
  - References
    - CouchDb_schema
    - lib_FullSyncGrp
    - lib_OAuth
  - Sequences

Important :

The library lib_UserManager

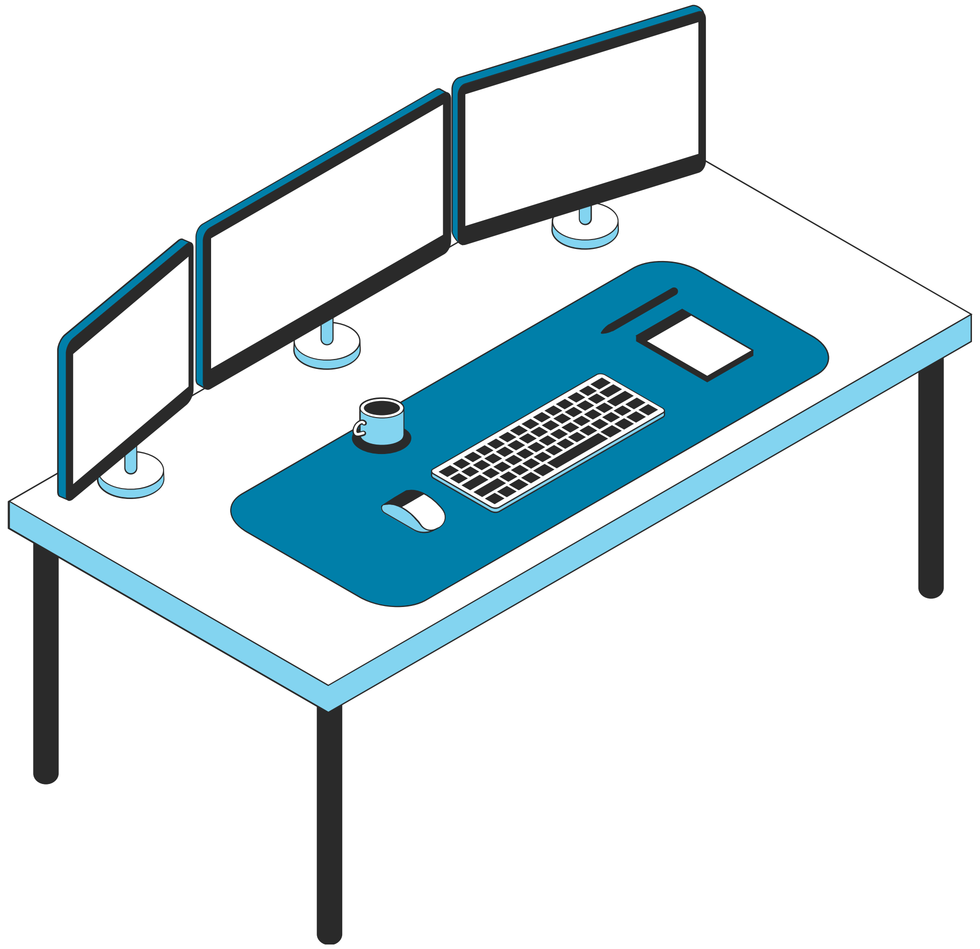
uses other libraries

(as shown in References folder)

and **they were imported** as well.

8 – Test platform

How to test your backend.



8.1 Access the Test platform

8.2 Test a transaction

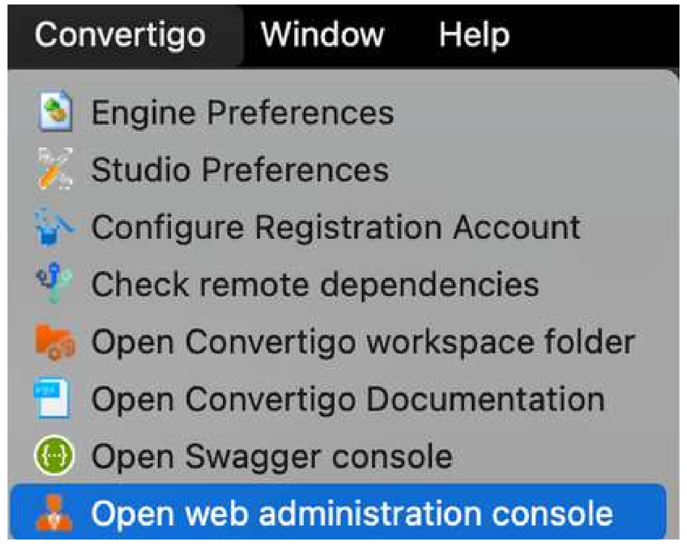
8.3 Test a sequence

8.1 Access the Test platform

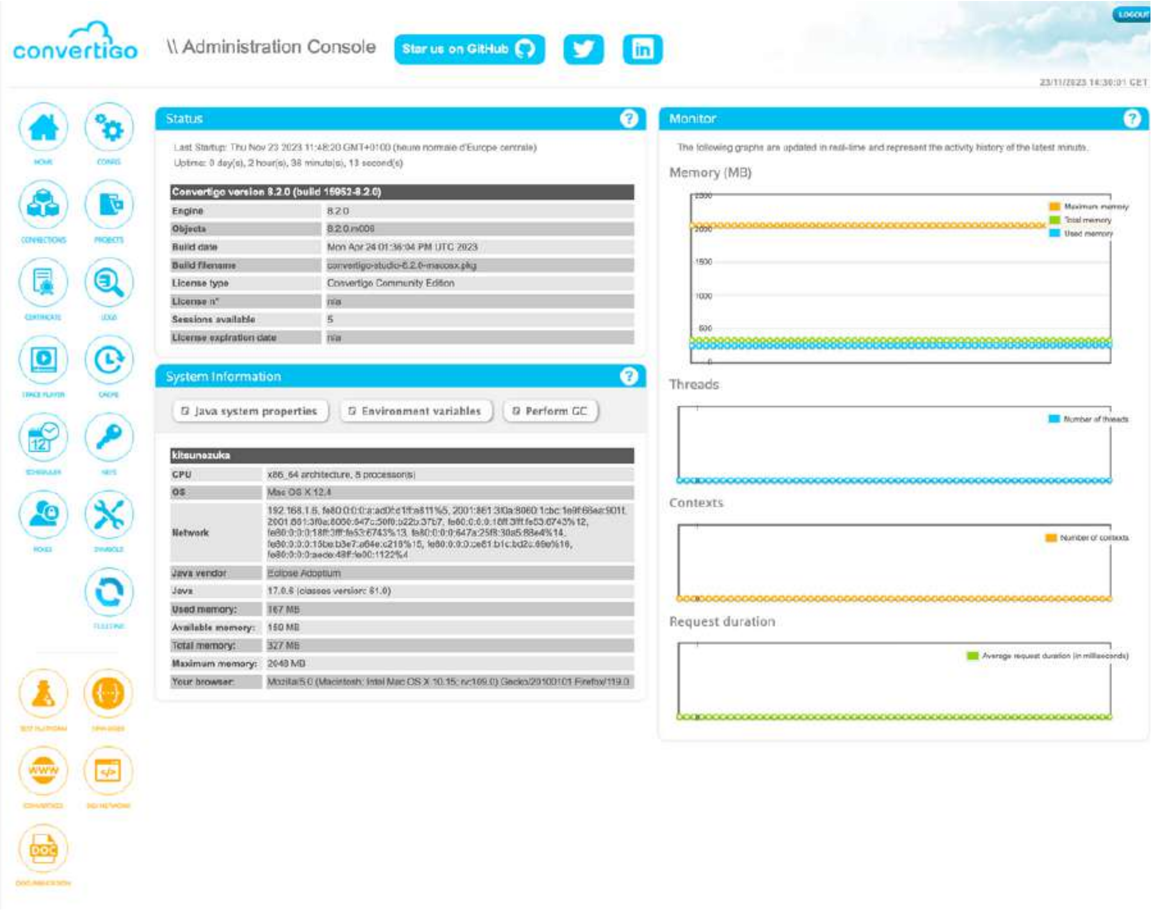
Convertigo provides a Test platform to test your backend and your frontend.

To access the Test platform

Open the **web administration console**.


Convertigo
Window
Help
Engine Preferences
Studio Preferences
Configure Registration Account
Check remote dependencies
Open Convertigo workspace folder
Open Convertigo Documentation
Open Swagger console
Open web administration console

In the web administration console



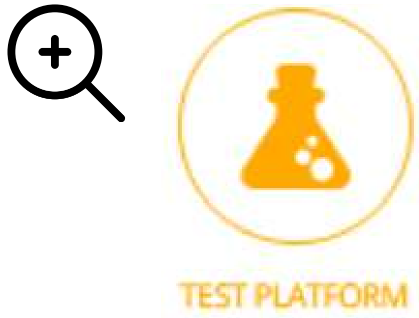
Last Startup: Thu Nov 23 2023 11:48:20 GMT+0100 (heure normale d'Europe centrale)
Uptime: 0 day(s), 2 hour(s), 38 minute(s), 13 second(s)

Convertigo version 8.2.0 (build 16952-8.2.0)	
Engine	8.2.0
Objects	8.2.0.m008
Build date	Mon Apr 24 01:36:04 PM UTC 2023
Build filename	convertigo-studio-8.2.0-macosx.pkg
License type	Convertigo Community Edition
License n°	n/a
Sessions available	5
License expiration date	n/a

kitsunezuka	
CPU	x86_64 architecture, 8 processor(s)
OS	Mac OS X 12.4
Network	192.168.1.6, fe80:0:0:0:a:ad0:c1ff:e811%5, 2001:861:3f0a:8060:1cbc:fe9f:88ea:5011, 2001:861:3f0a:8060:647c:50f0:b22b:37b7, fe80:0:0:0:18ff:3fff:fe53:6743%12, fe80:0:0:0:18ff:3fff:fe53:6743%13, fe80:0:0:0:647a:25f8:30a5:88e4%14, fe80:0:0:0:15be:b3e7:a64e:c218%15, fe80:0:0:0:ce81:b1c:bd2c:69e%16, fe80:0:0:0:aede:48ff:fe00:1122%4
Java vendor	Eclipse Adoptium
Java	17.0.8 (classes version: 61.0)
Used memory:	167 MB
Available memory:	150 MB
Total memory:	327 MB
Maximum memory:	2048 MB
Your browser:	Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:109.0) Gecko/20100101 Firefox/119.0

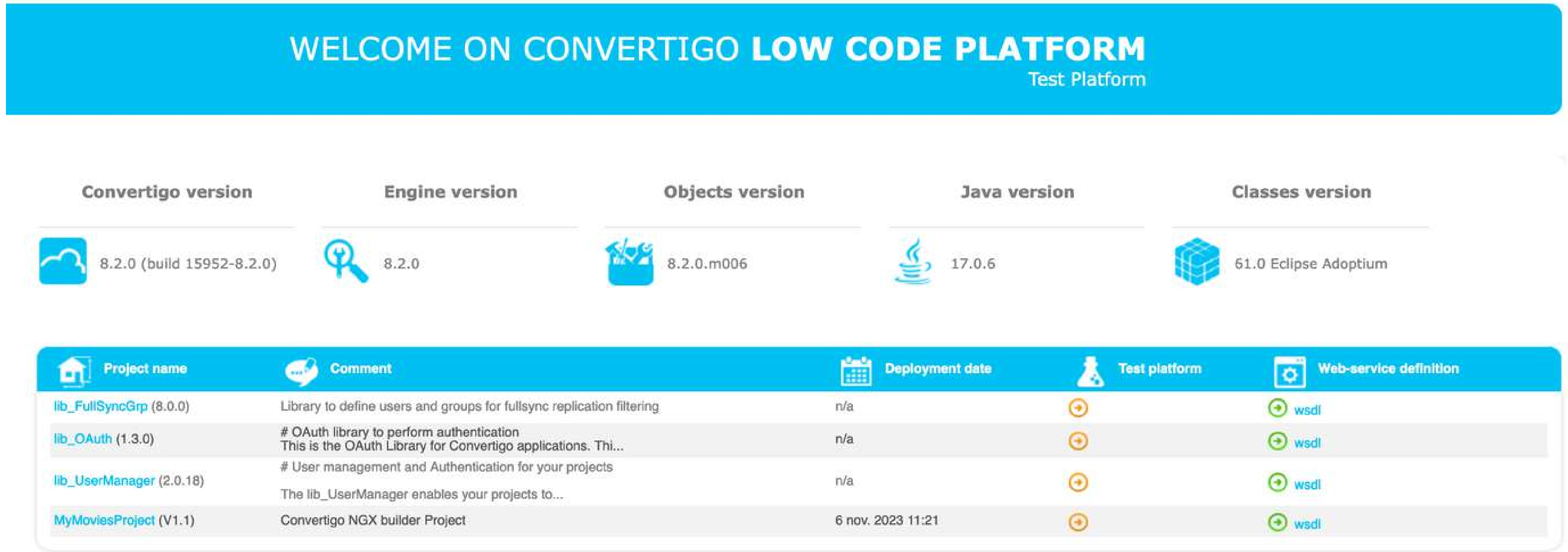
Click on

the **Test platform icon**.



8.1 Access the Test platform

In the Test platform are displayed **all the projects of your workspace**.

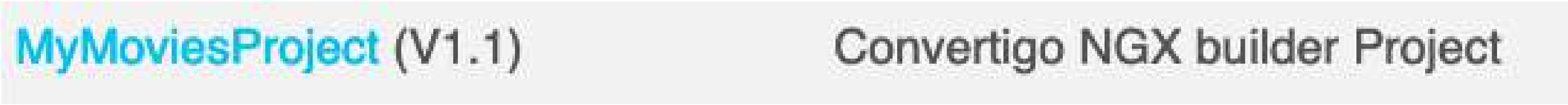


WELCOME ON CONVERTIGO **LOW CODE PLATFORM**
Test Platform

Convertigo version	Engine version	Objects version	Java version	Classes version
8.2.0 (build 15952-8.2.0)	8.2.0	8.2.0.m006	17.0.6	61.0 Eclipse Adoptium

Project name	Comment	Deployment date	Test platform	Web-service definition
lib_FullSyncGrp (8.0.0)	Library to define users and groups for fullsync replication filtering	n/a	➜	➜ wsdl
lib_OAuth (1.3.0)	# OAuth library to perform authentication This is the OAuth Library for Convertigo applications. Thi...	n/a	➜	➜ wsdl
lib_UserManager (2.0.18)	# User management and Authentication for your projects The lib_UserManager enables your projects to...	n/a	➜	➜ wsdl
MyMoviesProject (V1.1)	Convertigo NGX builder Project	6 nov. 2023 11:21	➜	➜ wsdl

Click on MyMoviesProject to select it.

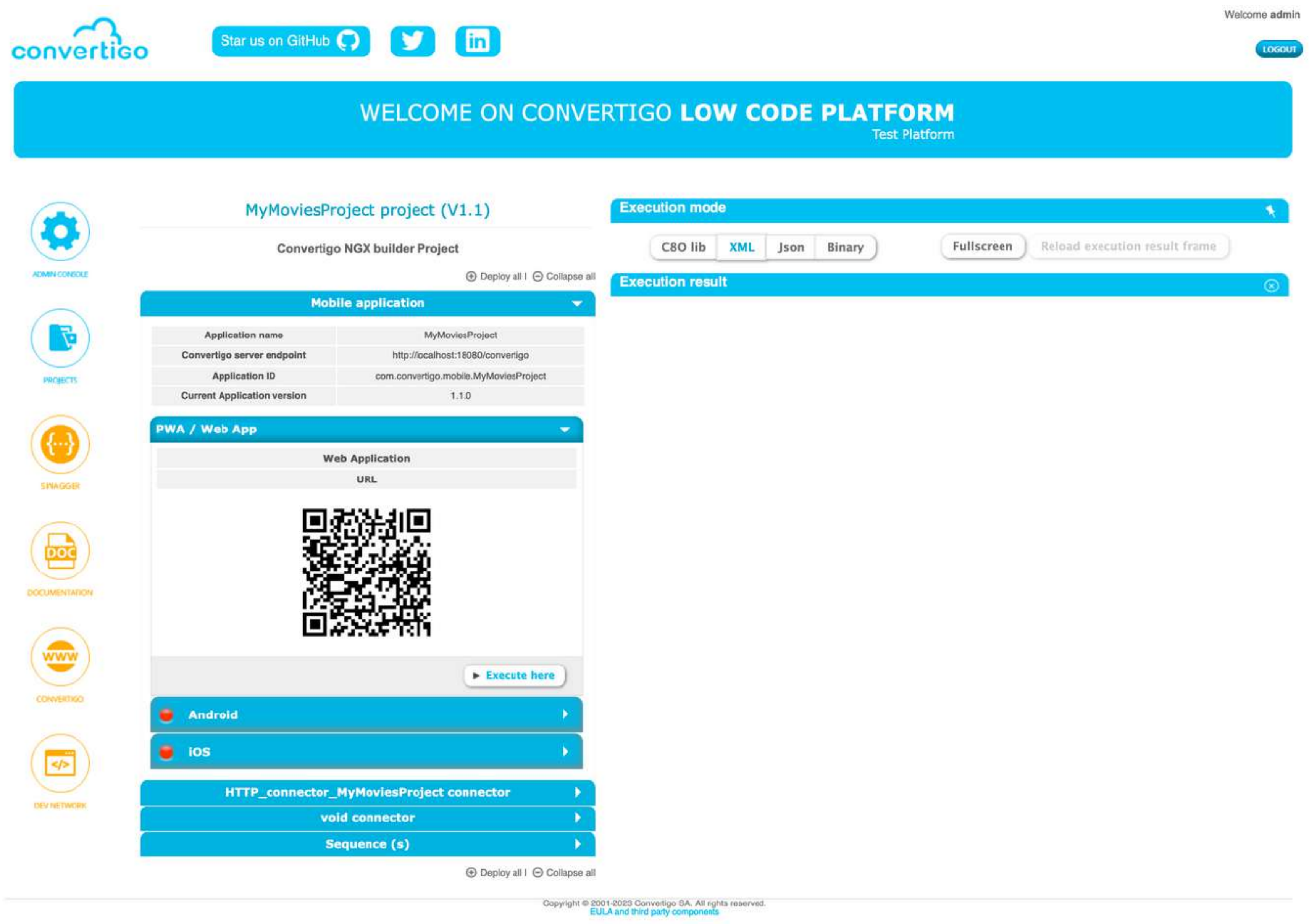


MyMoviesProject (V1.1) Convertigo NGX builder Project

8.1 Access the Test platform

In the MyMoviesProject page of the Test platform,

we can see **all the transactions and sequences** of the project.



MyMoviesProject project (V1.1)

Convertigo NGX builder Project

⊕ Deploy all I ⊖ Collapse all

Mobile application

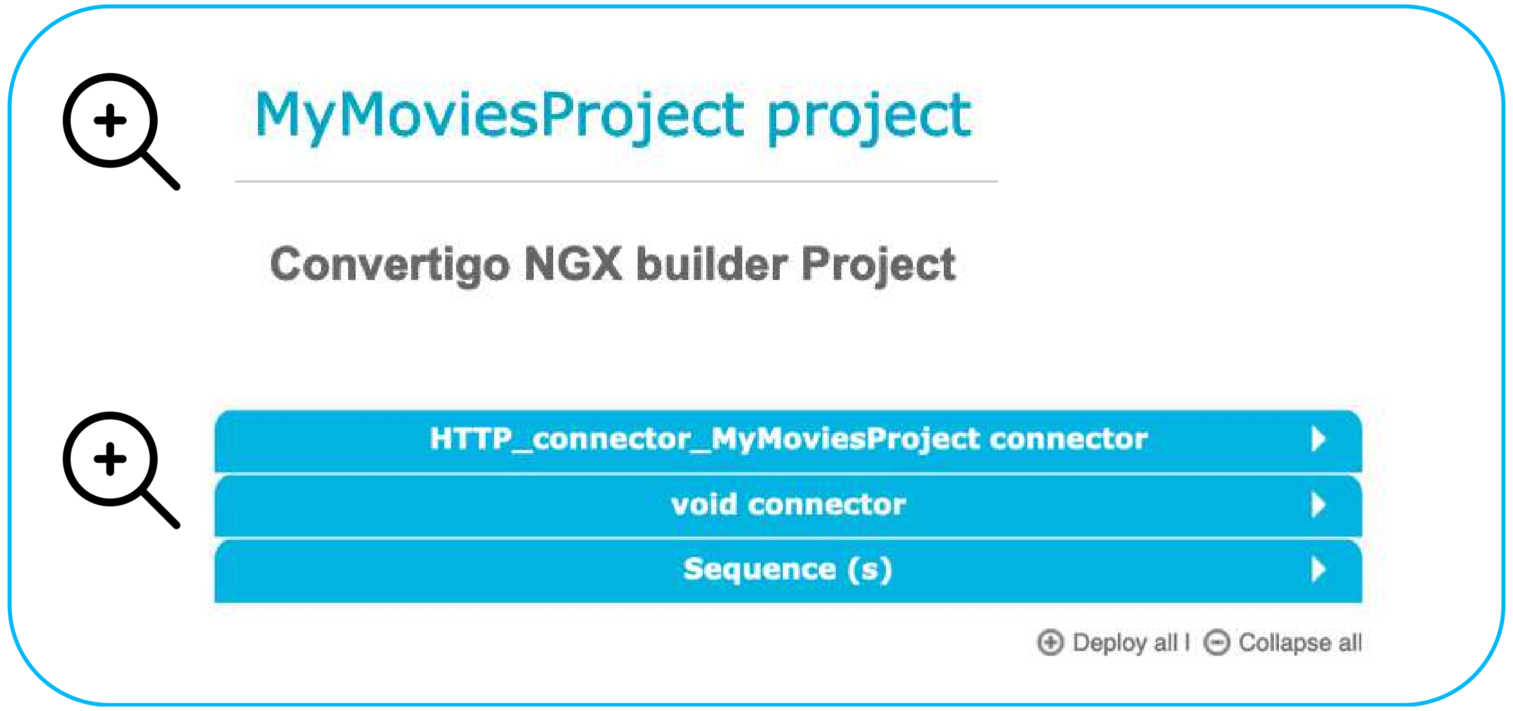
Application name	MyMoviesProject
Convertigo server endpoint	http://localhost:18080/convertigo
Application ID	com.convertigo.mobile.MyMoviesProject
Current Application version	1.1.0

PWA / Web App

Web Application

URL

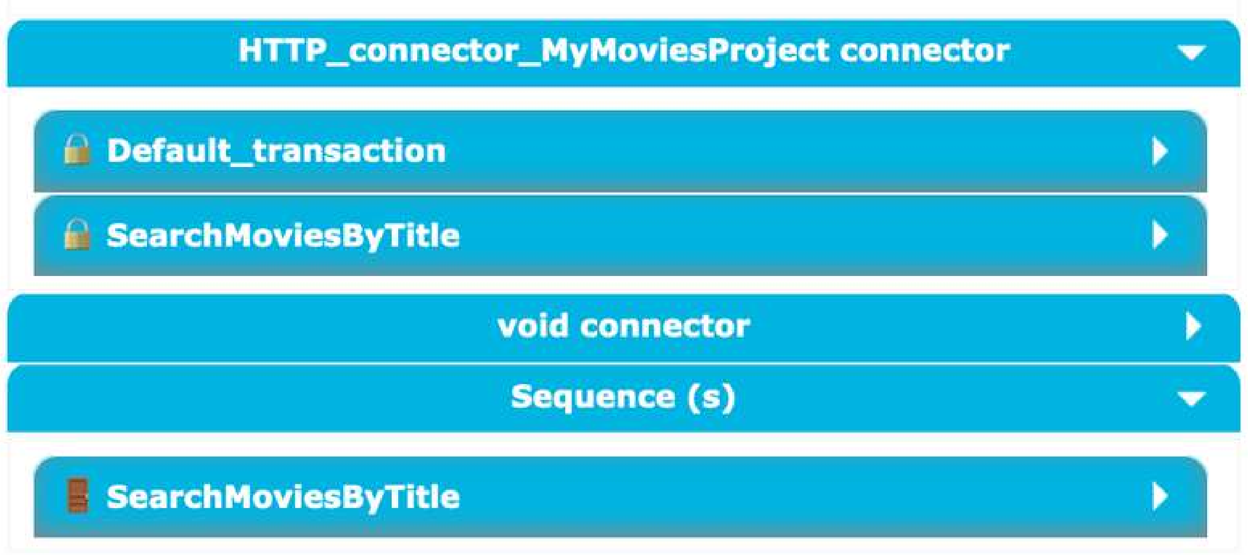
Copyright © 2001-2023 Convertigo SA. All rights reserved.
EULA and third party components



MyMoviesProject project

Convertigo NGX builder Project

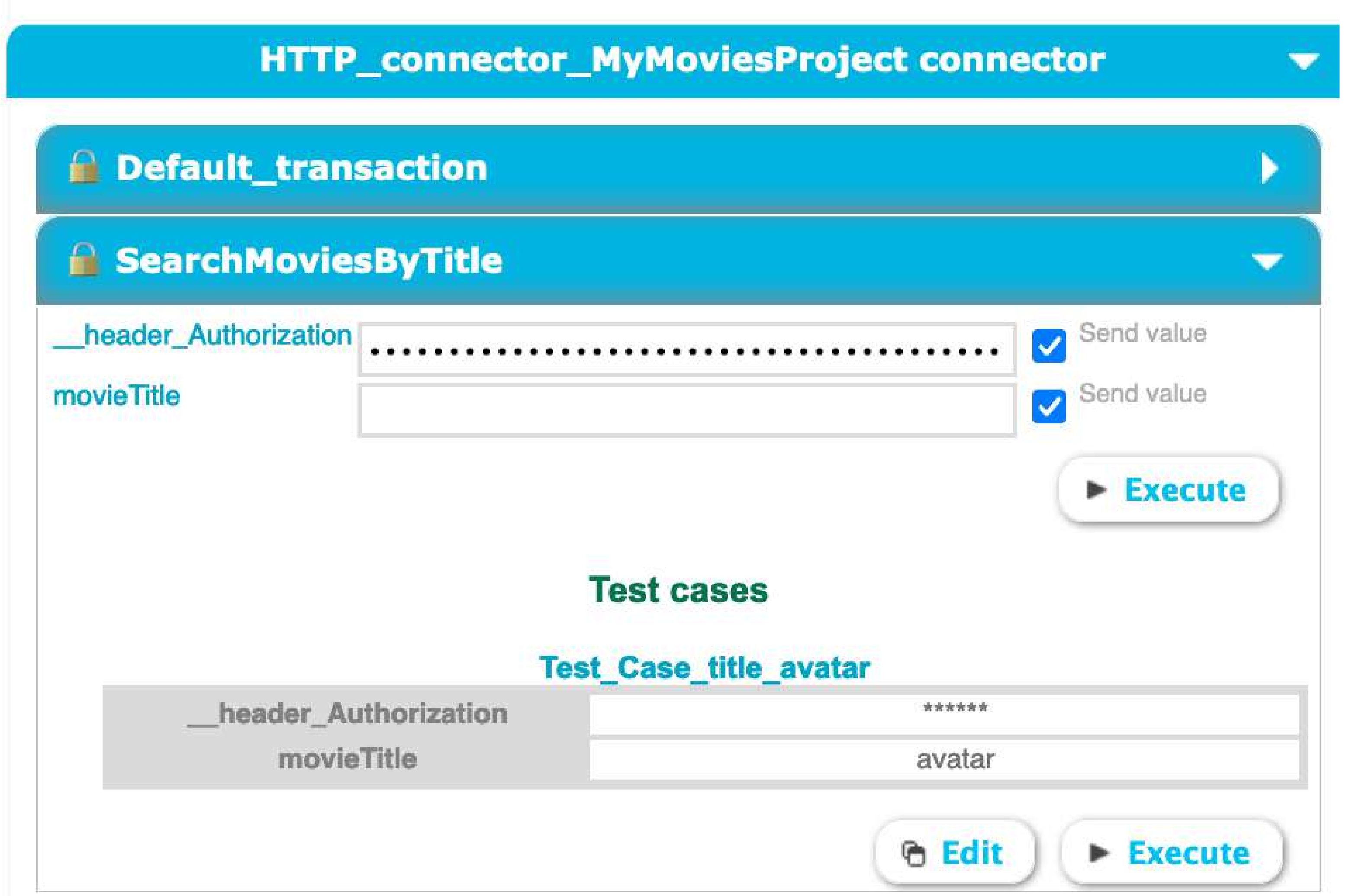
HTTP_connector_MyMoviesProject connector
void connector
Sequence (s)
⊕ Deploy all I ⊖ Collapse all



8.2 Test a transaction

Let’s test our SearchMoviesByTitle transaction.

When we deploy the transaction tab, we can see 2 parts:



__header_Authorization .... ☑ Send value

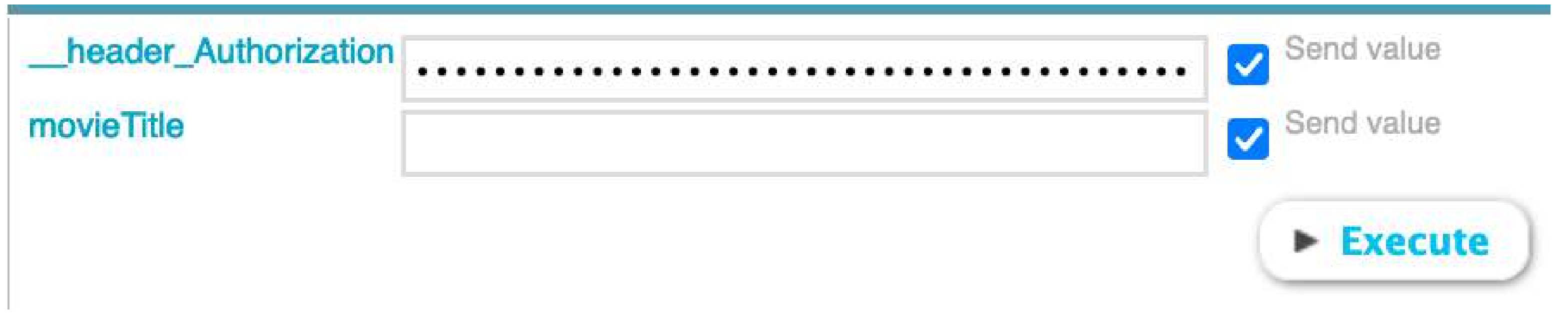
movieTitle ☑ Send value

Test cases

Test_Case_title_avatar

__header_Authorization	******
movieTitle	avatar

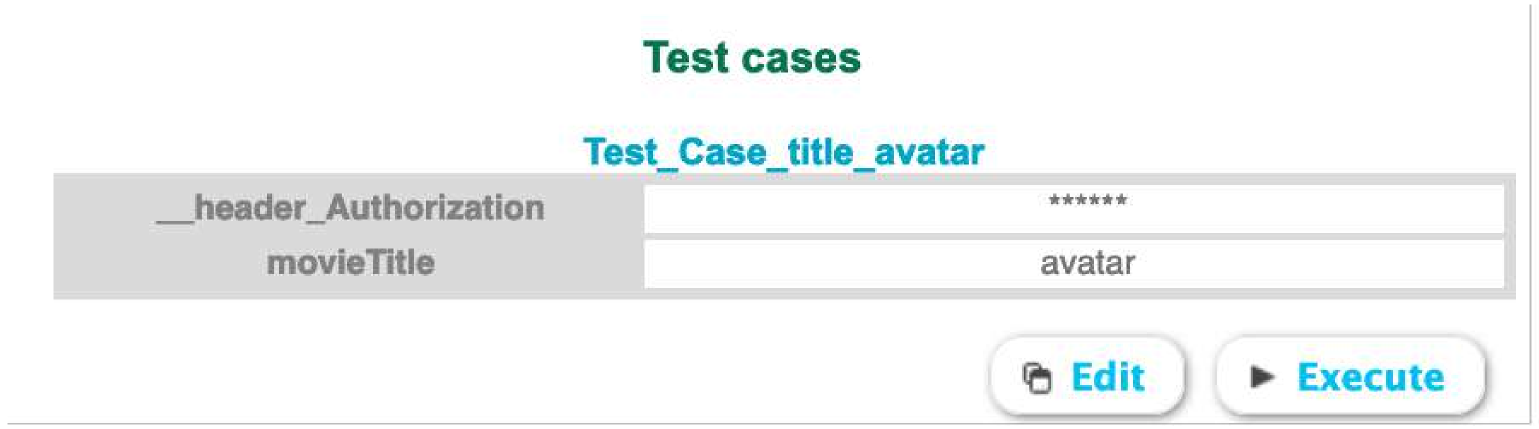
- an **editor with our transaction variables** where we can enter a movieTitle variable.



__header_Authorization .... ☑ Send value

movieTitle ☑ Send value

- the **test case we created in our transaction**.



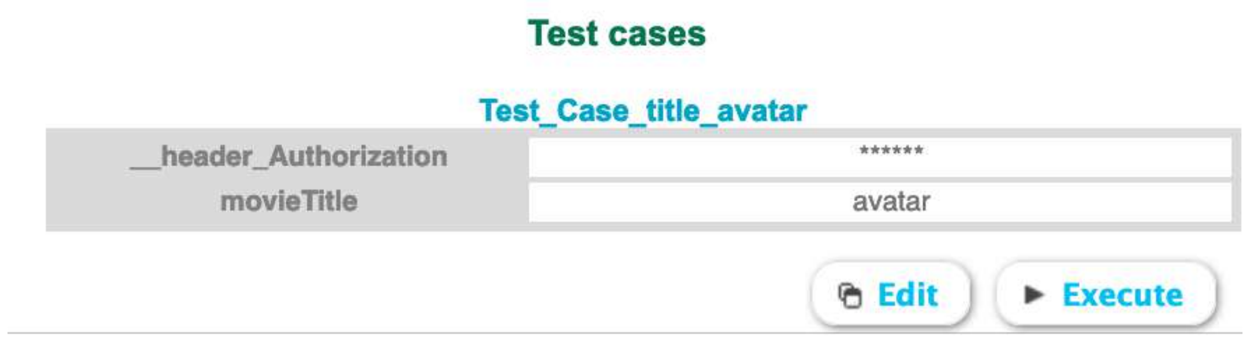
Test cases

Test_Case_title_avatar

__header_Authorization	******
movieTitle	avatar

8.2 Test a transaction

Let’s try the **test case** we created in our project with “avatar” as value for the movieTitle variable.

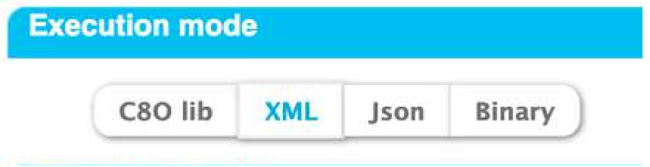


Test cases

Test_Case_title_avatar

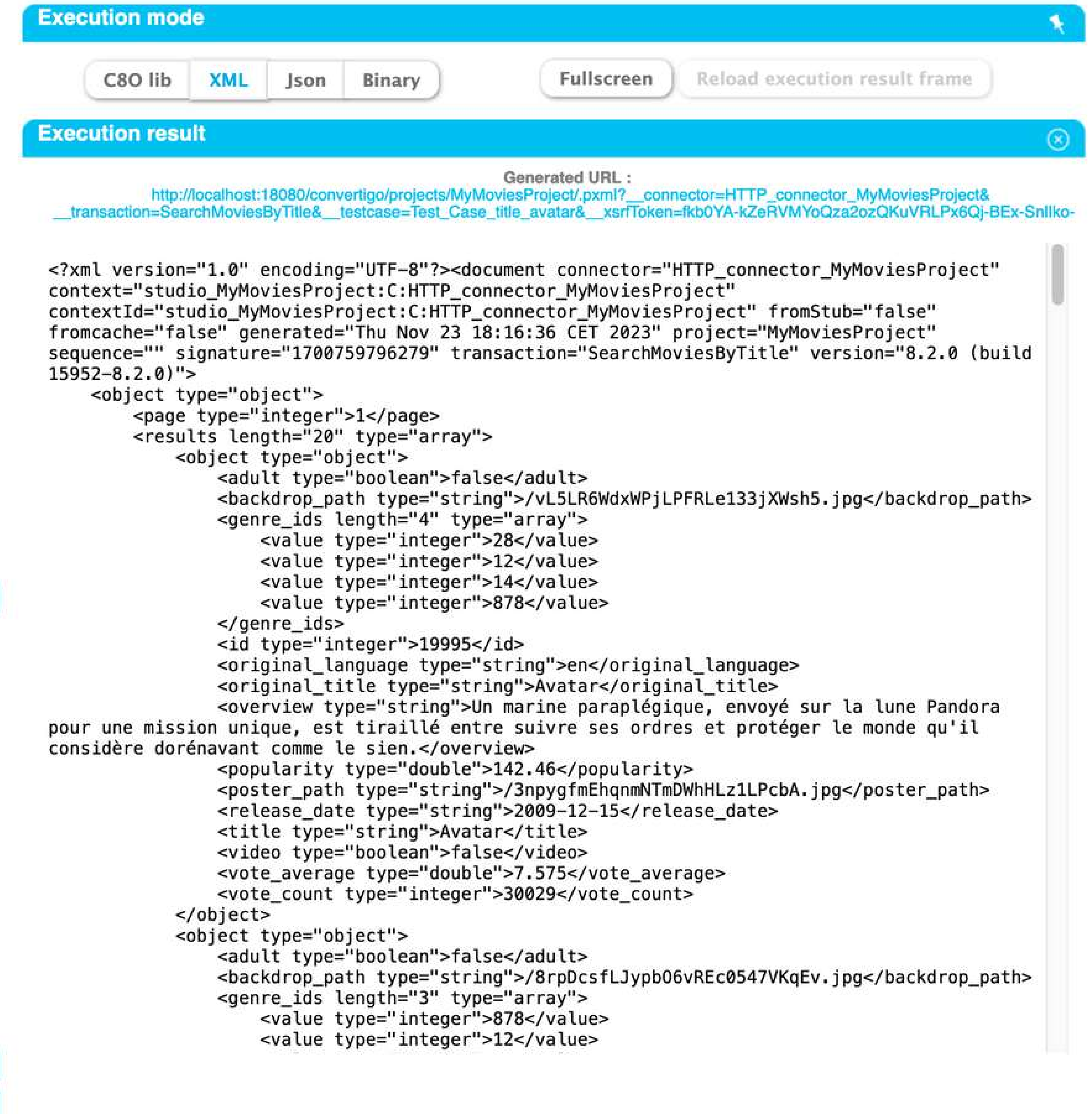
__header_Authorization	******
movieTitle	avatar

The result will be **displayed in XML** by default.



Click on **Execute** to run the test case.

The result is **displayed in XML**.

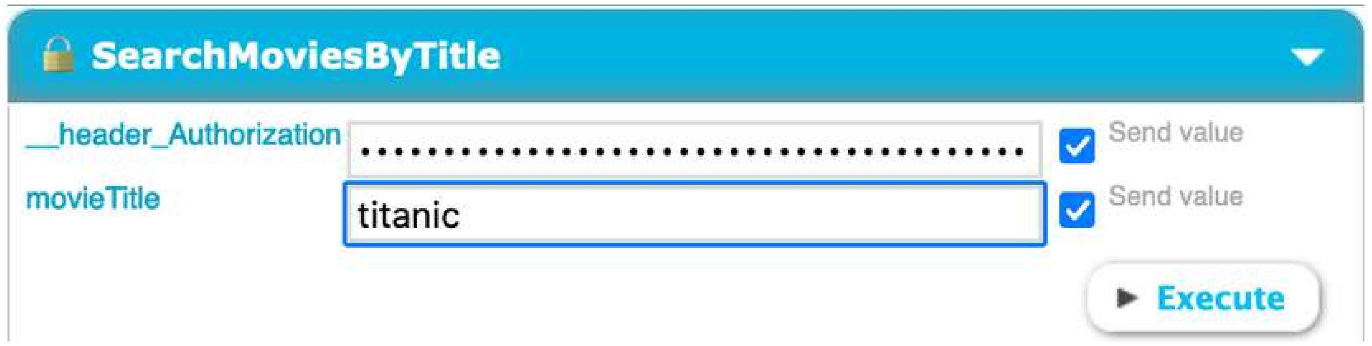


Generated URL :
http://localhost:18080/convertigo/projects/MyMoviesProject/.pxml?__connector=HTTP_connector_MyMoviesProject&__transaction=SearchMoviesByTitle&__testcase=Test_Case_title_avatar&__xsrfToken=fkb0YA-kZeRVMYoQza2ozQKuVRLPx6Qj-BEx-SnIlko-

```
<?xml version="1.0" encoding="UTF-8"?><document connector="HTTP_connector_MyMoviesProject"
context="studio_MyMoviesProject:C:HTTP_connector_MyMoviesProject"
contextId="studio_MyMoviesProject:C:HTTP_connector_MyMoviesProject" fromStub="false"
fromcache="false" generated="Thu Nov 23 18:16:36 CET 2023" project="MyMoviesProject"
sequence="" signature="1700759796279" transaction="SearchMoviesByTitle" version="8.2.0 (build
15952-8.2.0)">
    <object type="object">
        <page type="integer">1</page>
        <results length="20" type="array">
            <object type="object">
                <adult type="boolean">false</adult>
                <backdrop_path type="string">/vL5LR6WdxWPjLPFRLe133jXWsh5.jpg</backdrop_path>
                <genre_ids length="4" type="array">
                    <value type="integer">28</value>
                    <value type="integer">12</value>
                    <value type="integer">14</value>
                    <value type="integer">878</value>
                </genre_ids>
                <id type="integer">19995</id>
                <original_language type="string">en</original_language>
                <original_title type="string">Avatar</original_title>
                <overview type="string">Un marine paraplégique, envoyé sur la lune Pandora
pour une mission unique, est tiraillé entre suivre ses ordres et protéger le monde qu'il
considère dorénavant comme le sien.</overview>
                <popularity type="double">142.46</popularity>
                <poster_path type="string">/3npygfmEhqnmNTmDWhHLz1LPcbA.jpg</poster_path>
                <release_date type="string">2009-12-15</release_date>
                <title type="string">Avatar</title>
                <video type="boolean">false</video>
                <vote_average type="double">7.575</vote_average>
                <vote_count type="integer">30029</vote_count>
            </object>
            <object type="object">
                <adult type="boolean">false</adult>
                <backdrop_path type="string">/8rpDcsfLJypbO6vREc0547VKqEv.jpg</backdrop_path>
                <genre_ids length="3" type="array">
                    <value type="integer">878</value>
                    <value type="integer">12</value>
```

8.2 Test a transaction

Let’s try the editor with our transaction variables with “titanic” as value for the movieTitle variable.

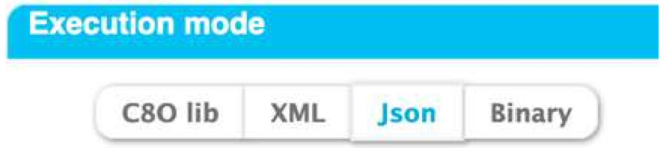


SearchMoviesByTitle

__header_Authorization: •• ☑ Send value

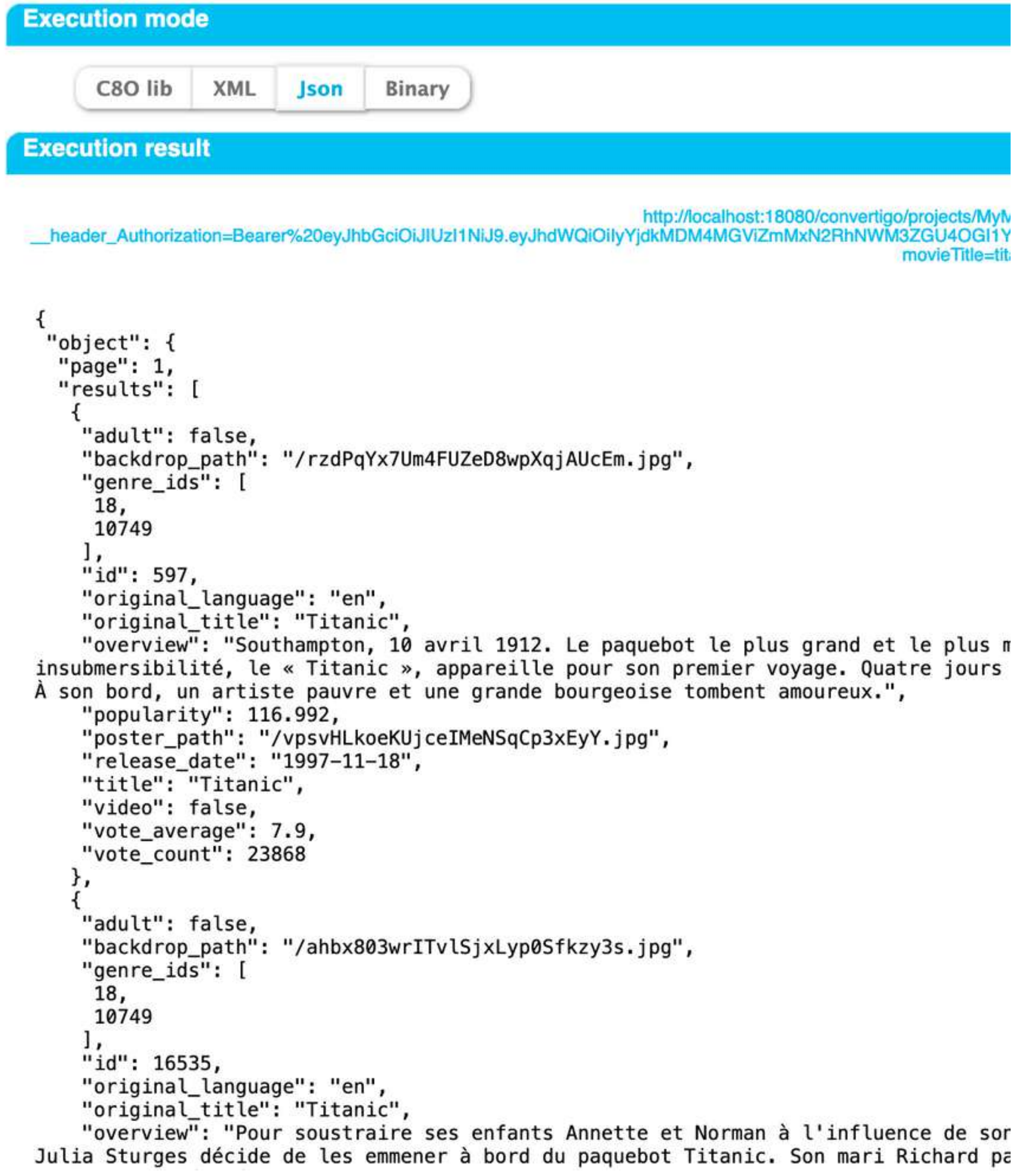
movieTitle: titanic ☑ Send value

Let’s change the **Execution mode to Json.**



Click on **Execute** to run the test case.

The result is **displayed in JSON**.



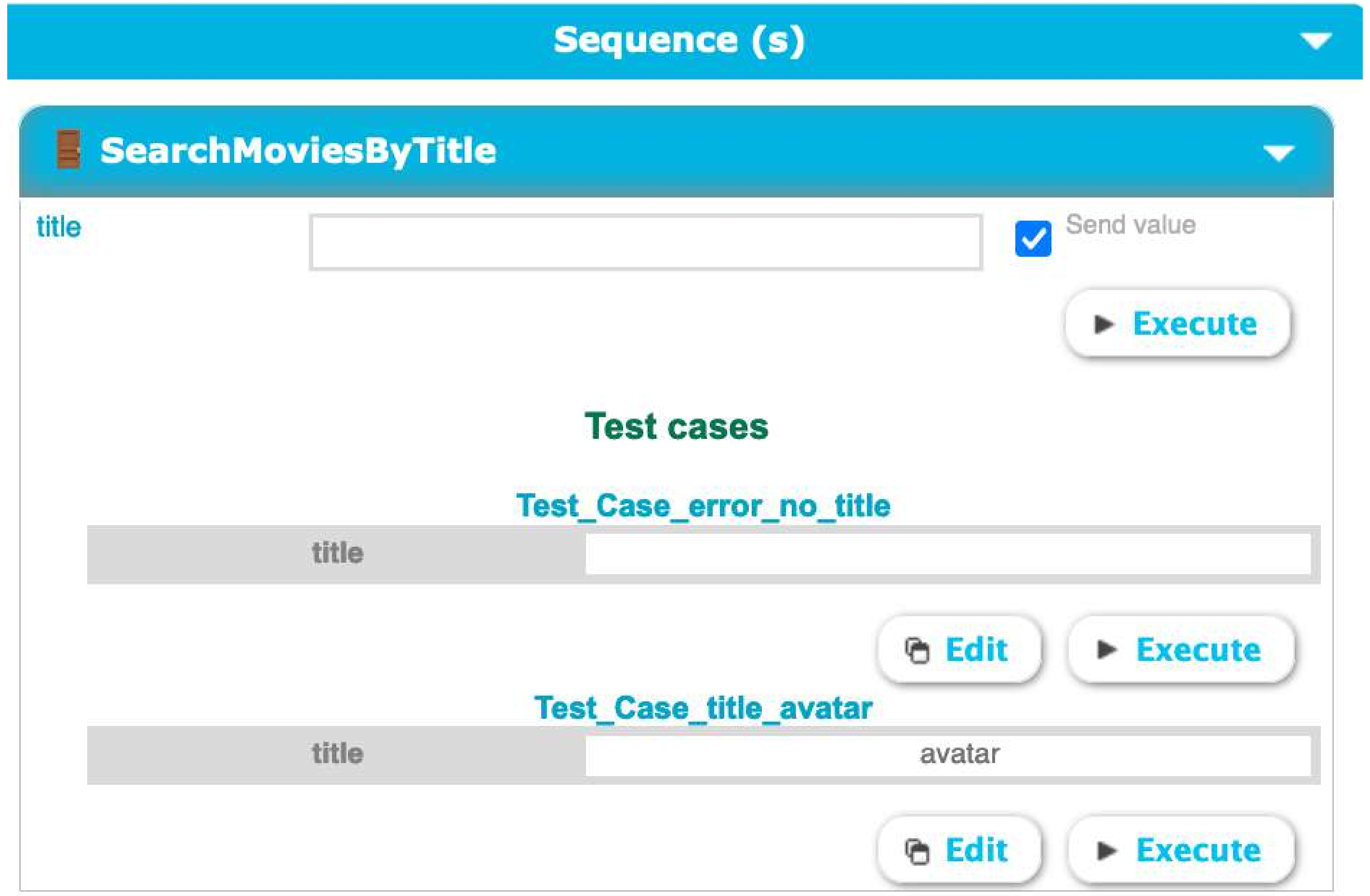
http://localhost:18080/convertigo/projects/MyM
__header_Authorization=Bearer%20eyJhbGciOiJIUzI1NiJ9.eyJhdWQiOilyYjdkMDM4MGViZmMxN2RhNWM3ZGU4OGI1Y
movieTitle=tit

```
{
 "object": {
  "page": 1,
  "results": [
   {
    "adult": false,
    "backdrop_path": "/rzdPqYx7Um4FUZeD8wpXqjAUcEm.jpg",
    "genre_ids": [
     18,
     10749
    ],
    "id": 597,
    "original_language": "en",
    "original_title": "Titanic",
    "overview": "Southampton, 10 avril 1912. Le paquebot le plus grand et le plus m
insubmersibilité, le « Titanic », appareille pour son premier voyage. Quatre jours
À son bord, un artiste pauvre et une grande bourgeoise tombent amoureux.",
    "popularity": 116.992,
    "poster_path": "/vpsvHLkoeKUjceIMeNSqCp3xEyY.jpg",
    "release_date": "1997-11-18",
    "title": "Titanic",
    "video": false,
    "vote_average": 7.9,
    "vote_count": 23868
   },
   {
    "adult": false,
    "backdrop_path": "/ahbx803wrITvlSjxLyp0Sfkzy3s.jpg",
    "genre_ids": [
     18,
     10749
    ],
    "id": 16535,
    "original_language": "en",
    "original_title": "Titanic",
    "overview": "Pour soustraire ses enfants Annette et Norman à l'influence de sor
Julia Sturges décide de les emmener à bord du paquebot Titanic. Son mari Richard pa
```

8.3 Test a sequence

Now, let’s test our SearchMoviesByTitle sequence.

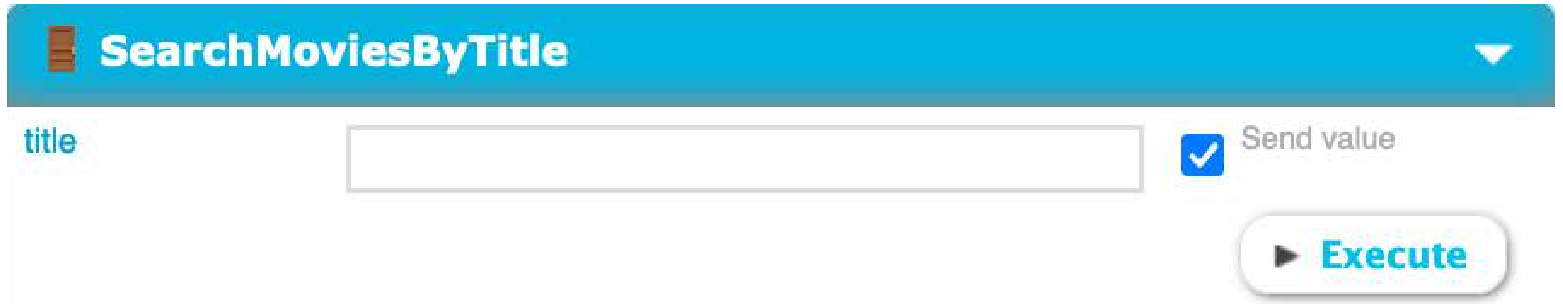
Here again, we can see 2 parts:



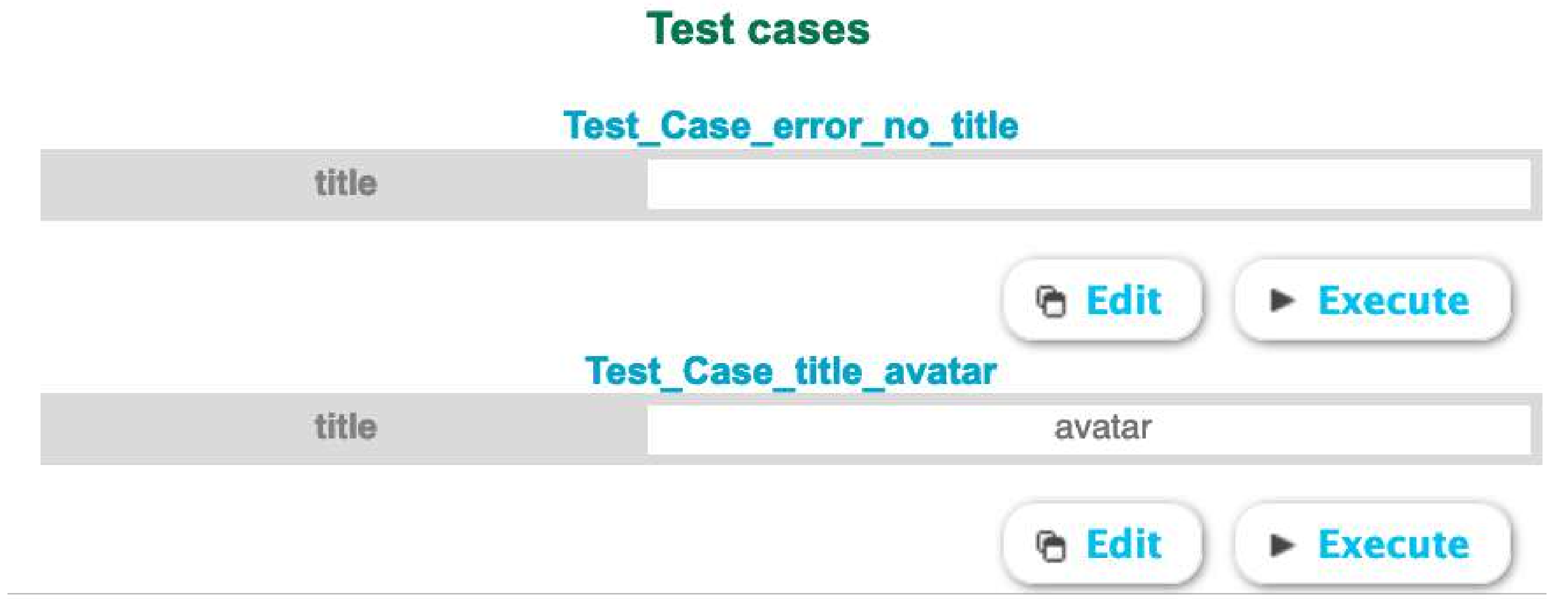
title	

title	avatar

- an editor with our sequence variable where we can enter a title variable.



- the test cases we created in our transaction.

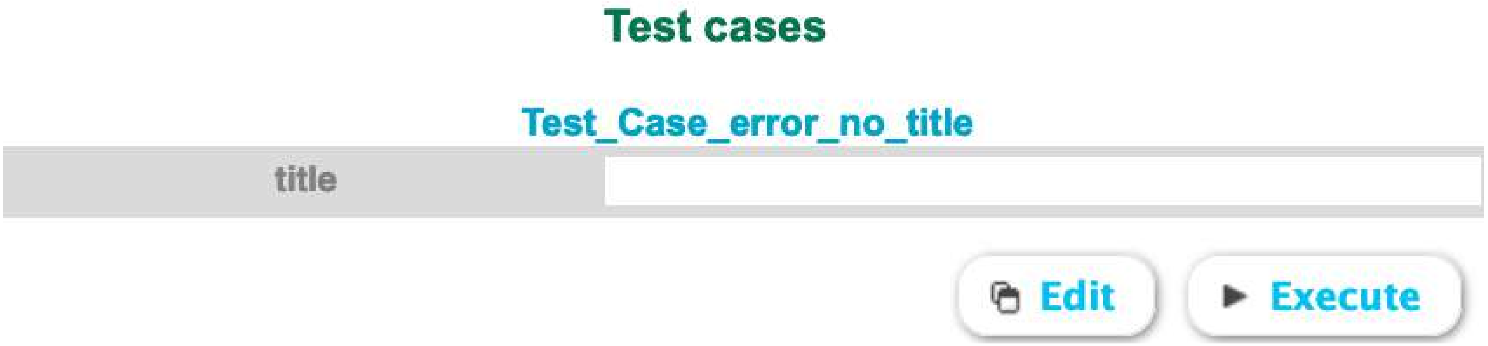


title	

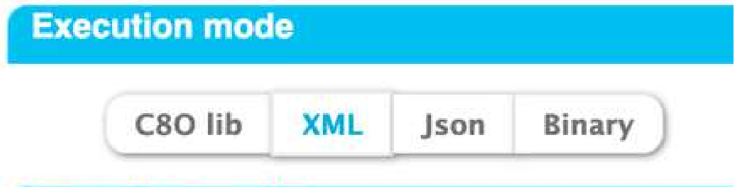
title	avatar

8.3 Test a sequence

Let’s try the error test case we created in our project with no value for the title variable.

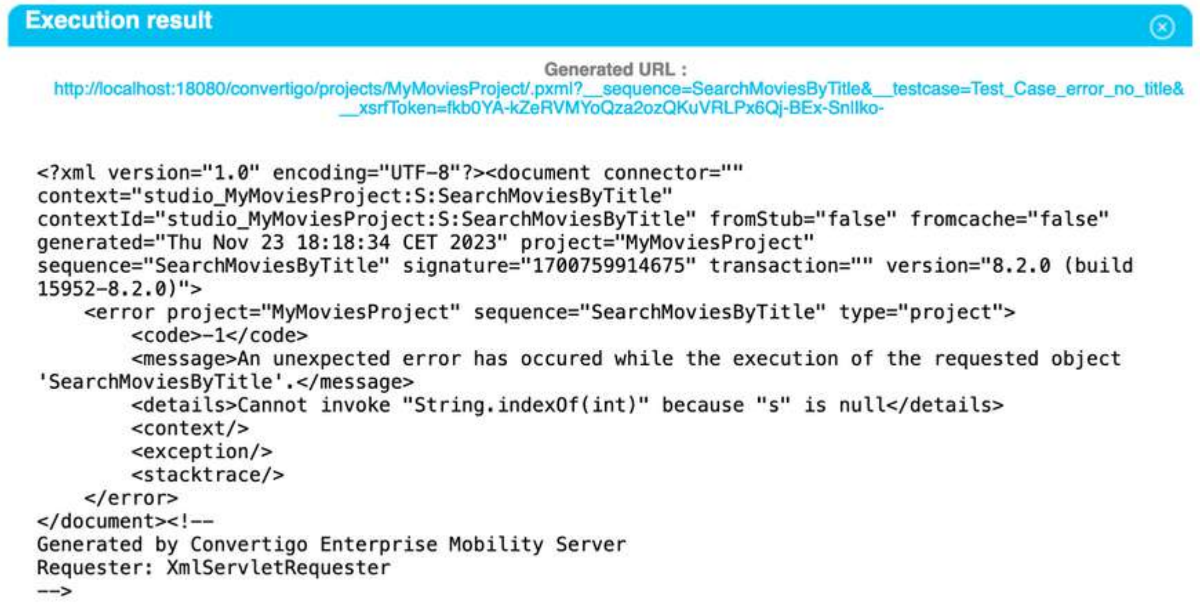


The **Execution mode** is in **XML**.



Click on **Execute** to run the test case.

The result is **displayed in XML**.

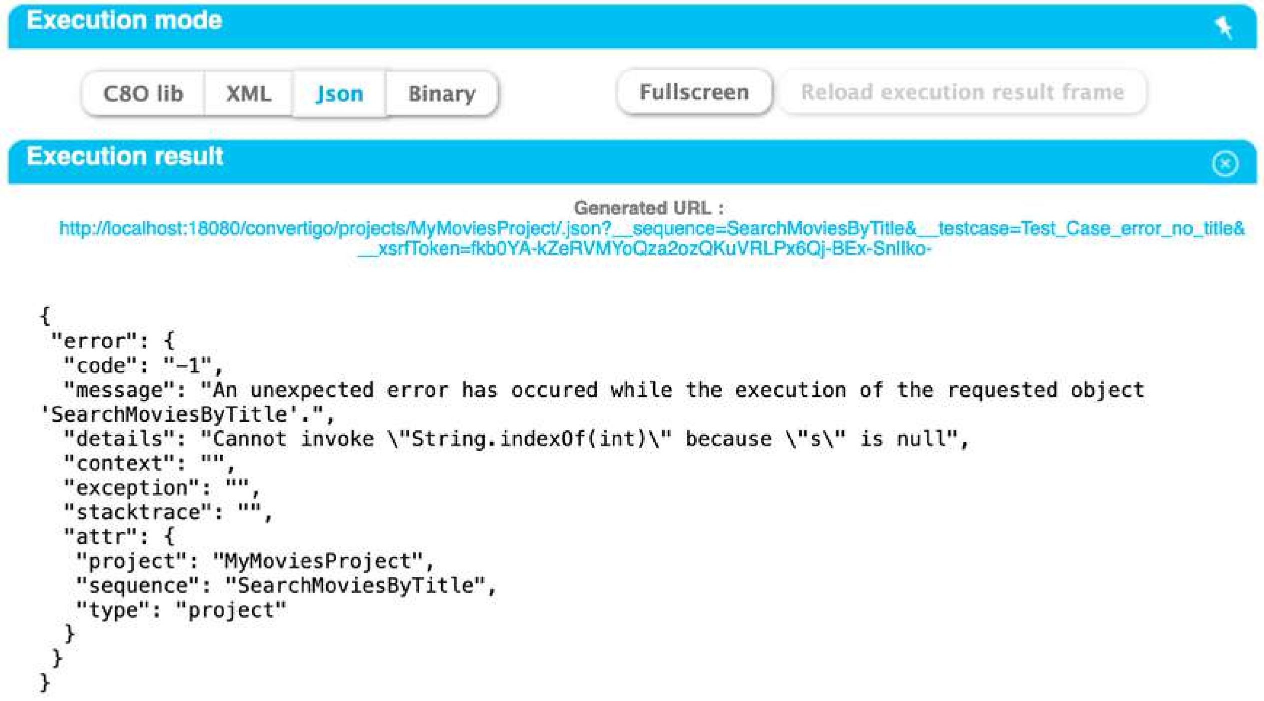


Generated URL :
http://localhost:18080/convertigo/projects/MyMoviesProject/.pxml?__sequence=SearchMoviesByTitle&__testcase=Test_Case_error_no_title&__xsrfToken=fkb0YA-kZeRVMYoQza2ozQKuVRLPx6Qj-BEx-Snllko-

```
<?xml version="1.0" encoding="UTF-8"?><document connector=""
context="studio_MyMoviesProject:S:SearchMoviesByTitle"
contextId="studio_MyMoviesProject:S:SearchMoviesByTitle" fromStub="false" fromcache="false"
generated="Thu Nov 23 18:18:34 CET 2023" project="MyMoviesProject"
sequence="SearchMoviesByTitle" signature="1700759914675" transaction="" version="8.2.0 (build
15952-8.2.0)">
    <error project="MyMoviesProject" sequence="SearchMoviesByTitle" type="project">
        <code>-1</code>
        <message>An unexpected error has occured while the execution of the requested object
'SearchMoviesByTitle'.</message>
        <details>Cannot invoke "String.indexOf(int)" because "s" is null</details>
        <context/>
        <exception/>
        <stacktrace/>
    </error>
</document><!--
Generated by Convertigo Enterprise Mobility Server
Requester: XmlServletRequester
-->
```

Let’s change the **Execution mode** to **Json**, and execute the test again.

The result is **displayed in JSON**.

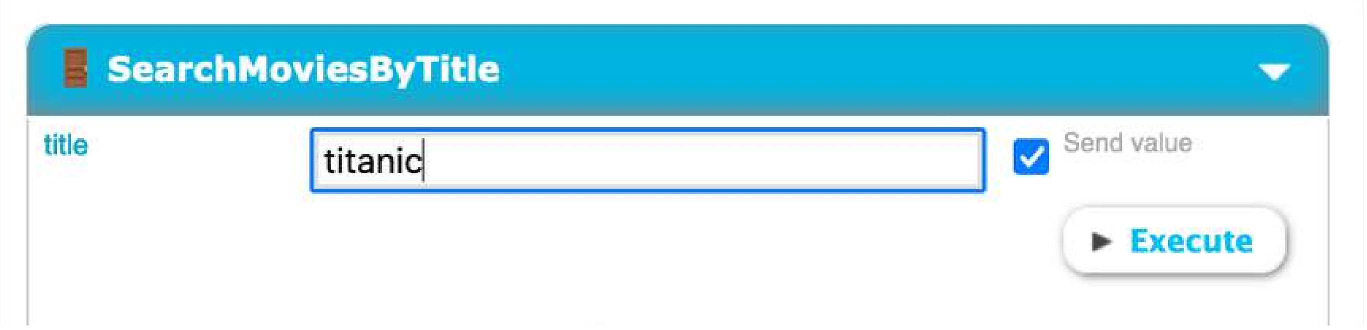


Generated URL :
http://localhost:18080/convertigo/projects/MyMoviesProject/.json?__sequence=SearchMoviesByTitle&__testcase=Test_Case_error_no_title&__xsrfToken=fkb0YA-kZeRVMYoQza2ozQKuVRLPx6Qj-BEx-Snllko-

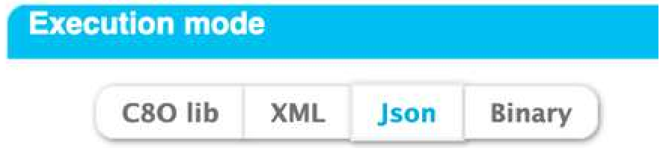
```
{
 "error": {
  "code": "-1",
  "message": "An unexpected error has occured while the execution of the requested object
'SearchMoviesByTitle'.",
  "details": "Cannot invoke \"String.indexOf(int)\" because \"s\" is null",
  "context": "",
  "exception": "",
  "stacktrace": "",
  "attr": {
   "project": "MyMoviesProject",
   "sequence": "SearchMoviesByTitle",
   "type": "project"
  }
 }
}
```

8.3 Test a sequence

Let’s try the editor with our sequence variable with “titanic” as value for the title variable.

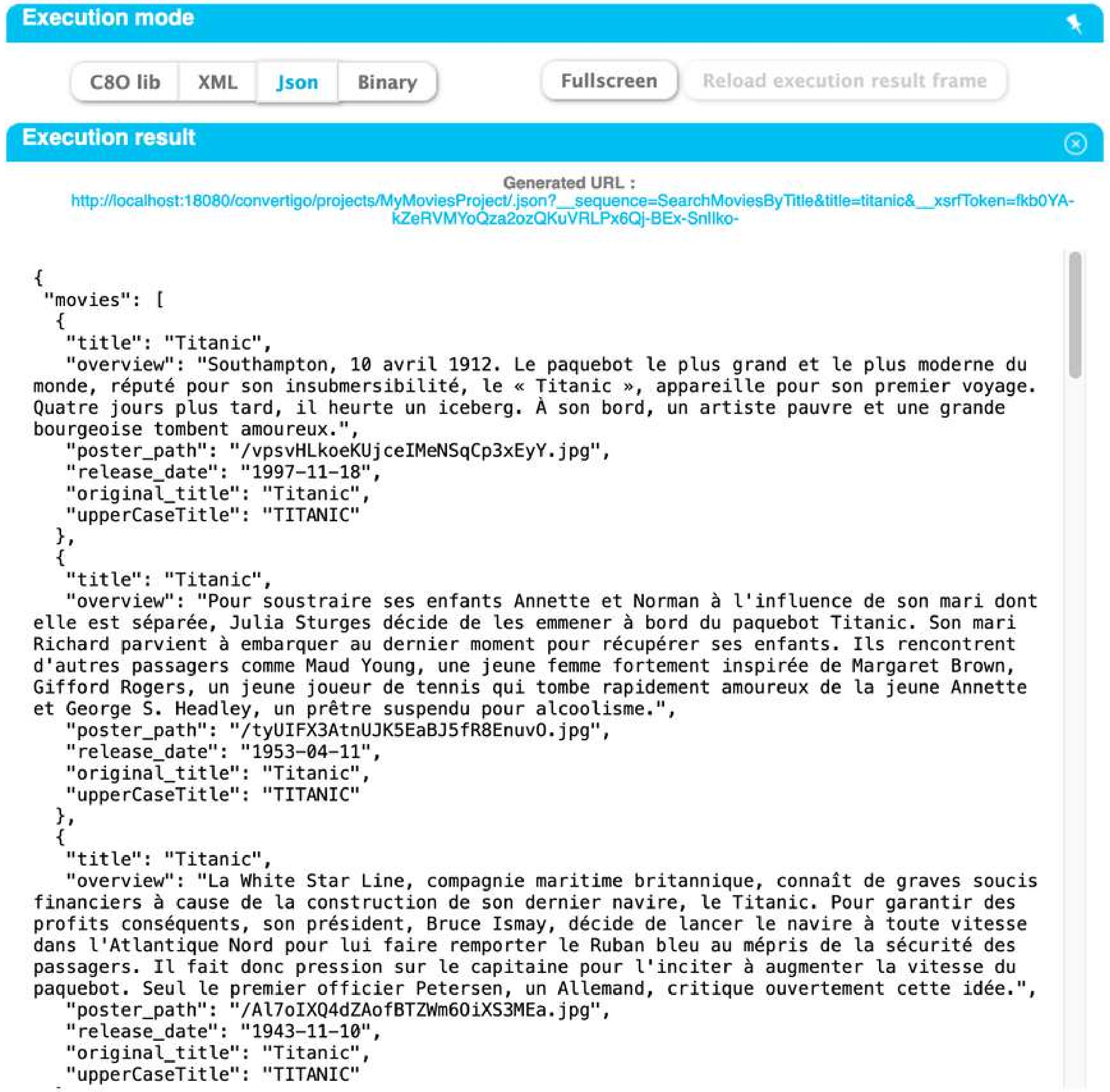


The **Execution mode is in Json**.



Click on **Execute** to run the test case.

The result is **displayed in JSON**.

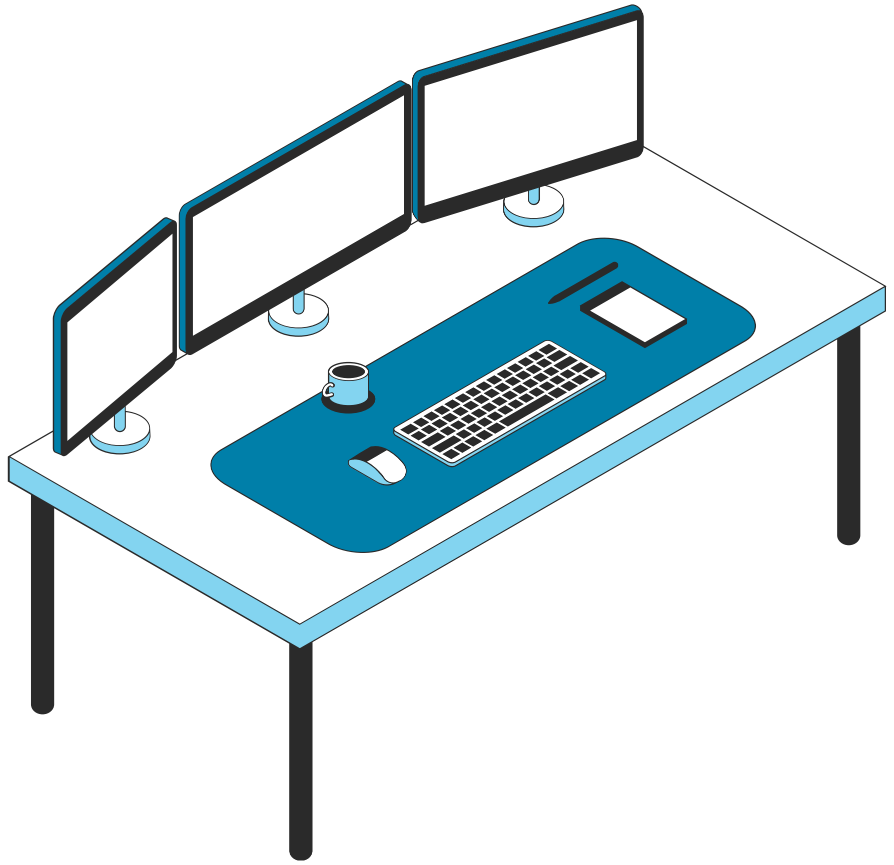


Generated URL :
http://localhost:18080/convertigo/projects/MyMoviesProject/.json?__sequence=SearchMoviesByTitle&title=titanic&__xsrfToken=fkb0YA-kZeRVMYoQza2ozQKuVRLPx6Qj-BEx-Snllko-

```
{
 "movies": [
  {
   "title": "Titanic",
   "overview": "Southampton, 10 avril 1912. Le paquebot le plus grand et le plus moderne du
monde, réputé pour son insubmersibilité, le « Titanic », appareille pour son premier voyage.
Quatre jours plus tard, il heurte un iceberg. À son bord, un artiste pauvre et une grande
bourgeoise tombent amoureux.",
   "poster_path": "/vpsvHLkoeKUjceIMeNSqCp3xEyY.jpg",
   "release_date": "1997-11-18",
   "original_title": "Titanic",
   "upperCaseTitle": "TITANIC"
  },
  {
   "title": "Titanic",
   "overview": "Pour soustraire ses enfants Annette et Norman à l'influence de son mari dont
elle est séparée, Julia Sturges décide de les emmener à bord du paquebot Titanic. Son mari
Richard parvient à embarquer au dernier moment pour récupérer ses enfants. Ils rencontrent
d'autres passagers comme Maud Young, une jeune femme fortement inspirée de Margaret Brown,
Gifford Rogers, un jeune joueur de tennis qui tombe rapidement amoureux de la jeune Annette
et George S. Headley, un prêtre suspendu pour alcoolisme.",
   "poster_path": "/tyUIFX3AtnUJK5EaBJ5fR8EnuvO.jpg",
   "release_date": "1953-04-11",
   "original_title": "Titanic",
   "upperCaseTitle": "TITANIC"
  },
  {
   "title": "Titanic",
   "overview": "La White Star Line, compagnie maritime britannique, connaît de graves soucis
financiers à cause de la construction de son dernier navire, le Titanic. Pour garantir des
profits conséquents, son président, Bruce Ismay, décide de lancer le navire à toute vitesse
dans l'Atlantique Nord pour lui faire remporter le Ruban bleu au mépris de la sécurité des
passagers. Il fait donc pression sur le capitaine pour l'inciter à augmenter la vitesse du
paquebot. Seul le premier officier Petersen, un Allemand, critique ouvertement cette idée.",
   "poster_path": "/Al7oIXQ4dZAofBTZWm6OiXS3MEa.jpg",
   "release_date": "1943-11-10",
   "original_title": "Titanic",
   "upperCaseTitle": "TITANIC"
```

9 – URL mapper

How to expose an API REST.



9.1 What is the URL mapper ?

9.2 URL mapper steps

9.3 Create an URL mapper for a transaction

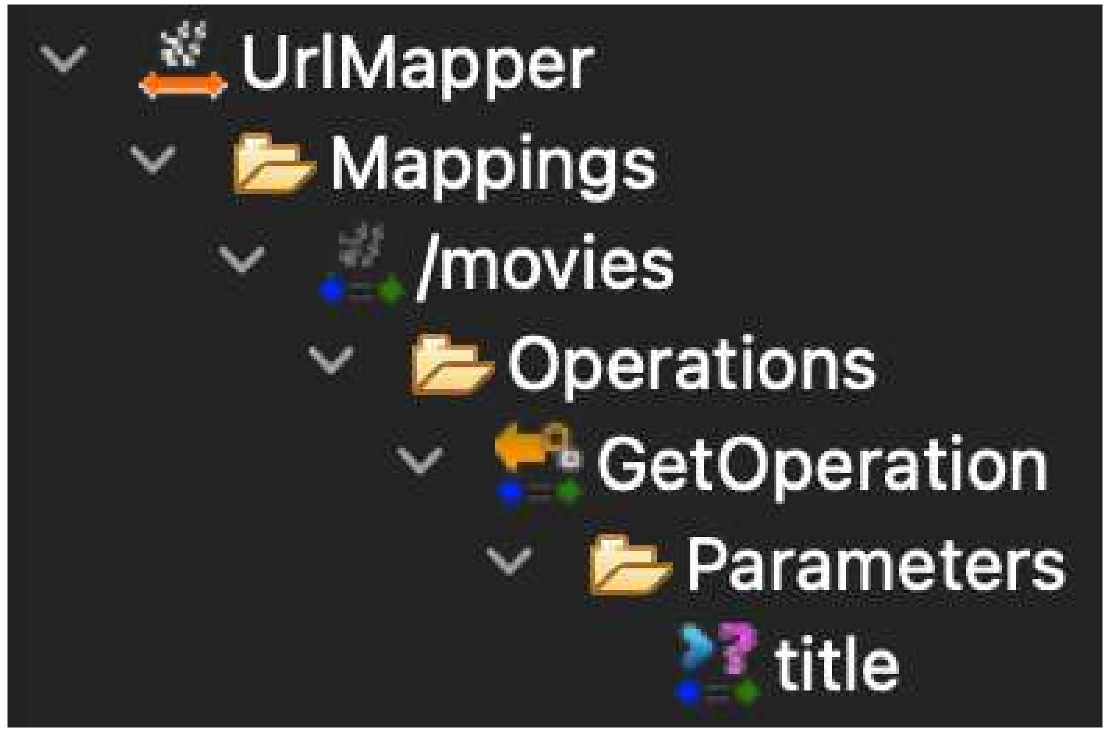
9.4 Test the URL mapper on Swagger

9.1 What is the URL mapper ?

The **URL mapper** is able to map **RESTful urls** to **Convertigo requestables** such as **Sequences and Transactions**.
This way Convertigo **can expose RESTful APIs** to the outside world.

You can have **only one URLMapper per project**,
but an URLmapper **can map URLs to any otherproject deployed on the server**.

Example of URL mapper structure in a Convertigo project



```
UrlMapper
  Mappings
    /movies
      Operations
        GetOperation
          Parameters
            title
```

9.2 URL mapper steps

Convertigo provides steps to create the URL mapper.

UrlMapper

UrlMapper

This step defines the **URL mapper** to use in the project.

PathMapping

PathMapping – Mapping step

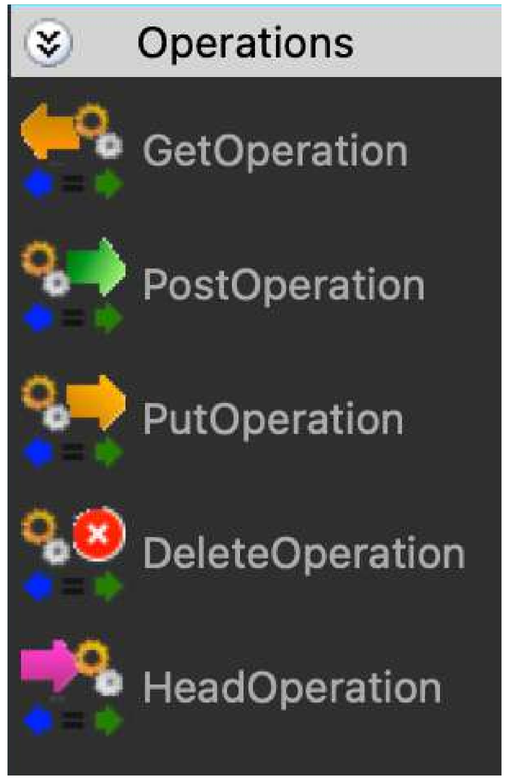
This step defines a **mapping path associated with the mapper**, the **base URL structure** an API user will have to use **to access this API Service**.
For example: /accounts/{accountid}.

9.2 URL mapper Objects

Operations Steps

These steps define the **HTTP operations associated with the mapping**.

For a **given operation on a given mapping**,
you define here **what should be the Requestable (Sequence or Transaction) to be executed**,
and **how will the variables for this requestable will be mapped**.



=> HTTP **GET** operation

=> HTTP **POST**operation

=> HTTP **PUT** operation

=> HTTP **DELETE** operation

=> HTTP **HEAD** operation

9.2 URL mapper Objects

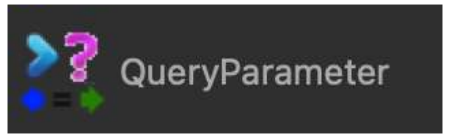
Parameters Steps

Convertigo provides steps to define parameters associated with the operation.

PathParameter

PathParameter – Parameters step

This step defines a path parameter by extracting the variable value from a segment of the URL path between {}. ex: /accounts/{accountid}



QueryParameter

QueryParameter – Parameters step

This step defines a query parameter by extracting the variable value from the query string.
ex: /accounts?verbose=1

HeaderParameter

HeaderParameter – Parameters step

This step defines a header parameter by extracting the variable value from the HTTP Header
of this parameter name.

9.2 URL mapper Objects

Responses Step

OperationResponse

OperationResponse – Responses step

This step defines an **HTTP response associated with the operation**.

When a service is invoked, it **responds with a HTTP status code**.

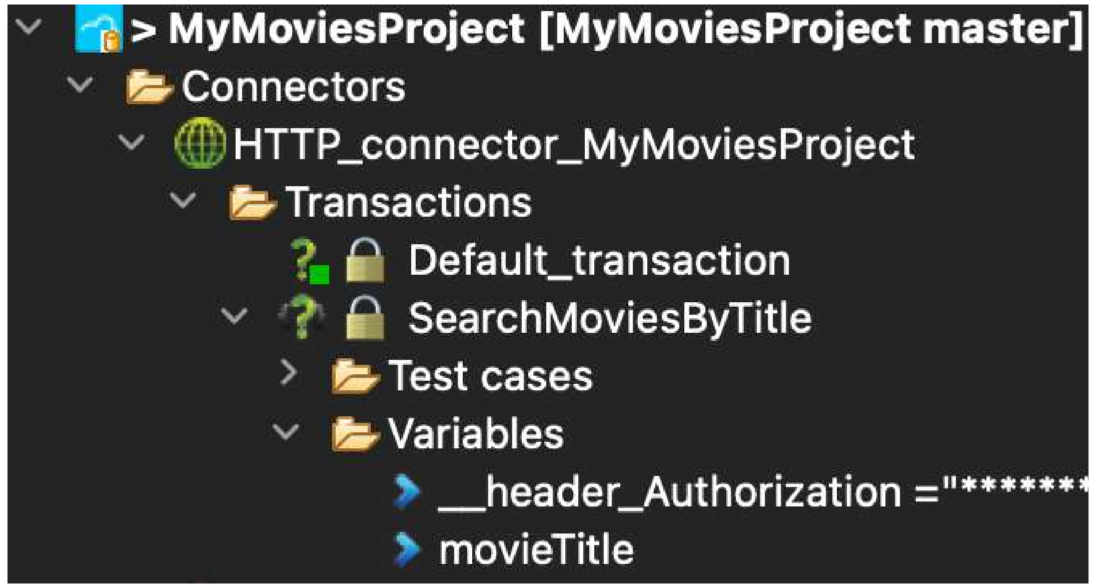
This mapping object will help you **define status codes** such as 200, 401 or any other **according to XPaths resolution** done on a **Convertigo Sequence response**.

The Sequence response will be **scanned by all the UrlMappingResponse objects** defined for a given operation. The **first one having its XPath matching** will **generate the corresponding status code**.

9.3 Create an URL mapper for a transaction

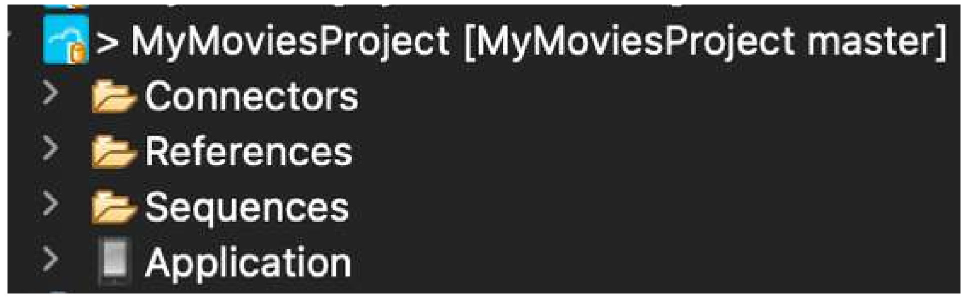
In our project, we have a **SearchMoviesByTitle transaction**, with a **variable named movieTitle**.

Let’s create an URL mapper for this transaction.

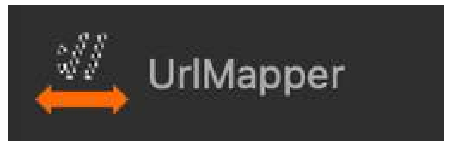


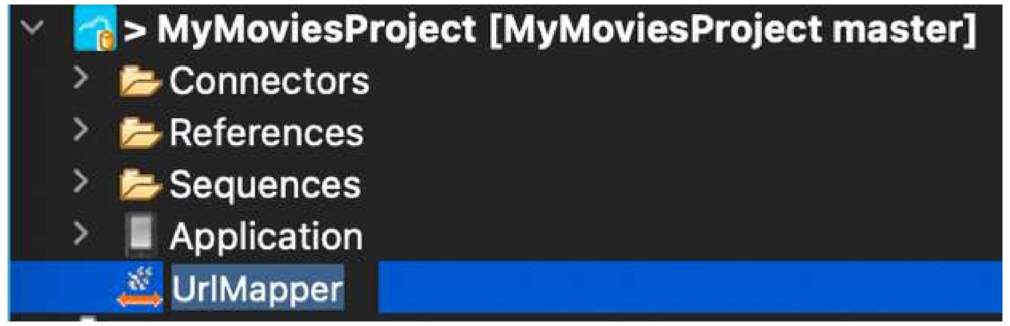
```
> MyMoviesProject [MyMoviesProject master]
  Connectors
    HTTP_connector_MyMoviesProject
      Transactions
        Default_transaction
        SearchMoviesByTitle
          Test cases
          Variables
            __header_Authorization ="*******
            movieTitle
```

Drag and drop the **UrlMapper step** from the palette in the project.



```
> MyMoviesProject [MyMoviesProject master]
  Connectors
  References
  Sequences
  Application
```

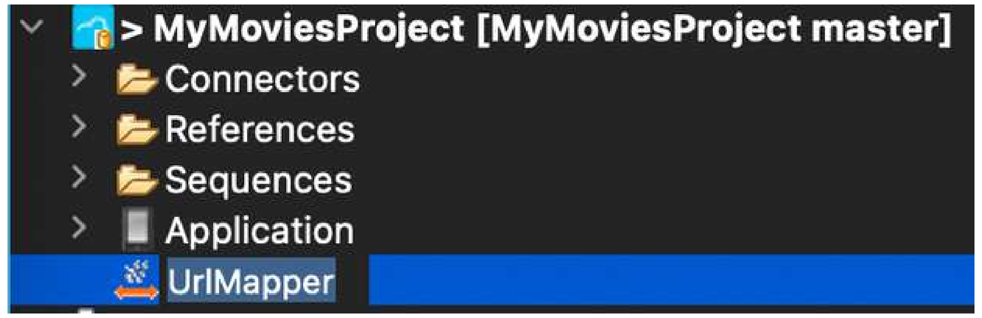




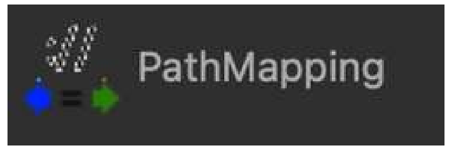
```
> MyMoviesProject [MyMoviesProject master]
  Connectors
  References
  Sequences
  Application
  UrlMapper
```

9.3 Create an URL mapper for a transaction.

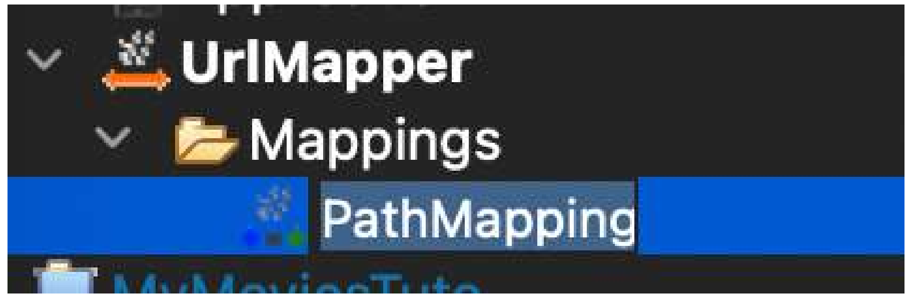
Drag and drop a **PathMapping step** from the palette in the **UrlMapper step**.



> MyMoviesProject [MyMoviesProject master]
Connectors
References
Sequences
Application
UrlMapper



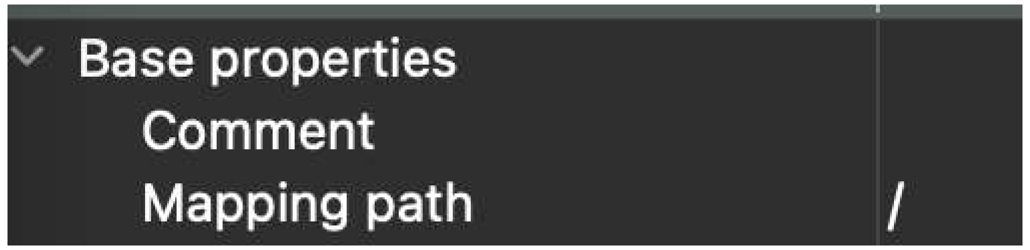
PathMapping



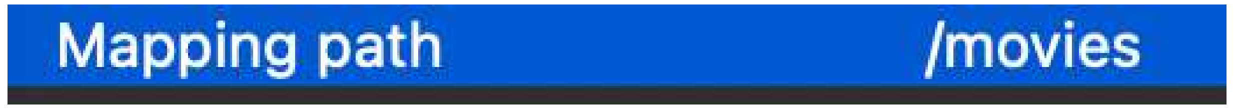
UrlMapper
Mappings
PathMapping

In the properties,

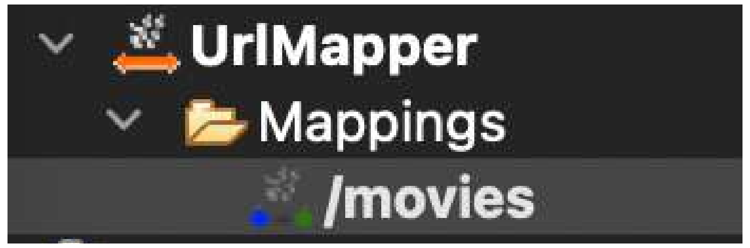
rename the Mapping path as /movies.



Base properties
Comment
Mapping path /



Mapping path /movies

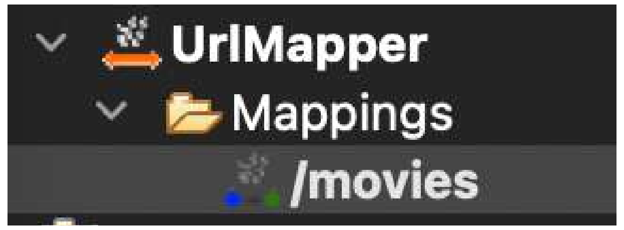


UrlMapper
Mappings
/movies

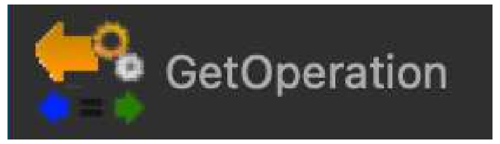
The path will appear as **/movies in the url.**

9.3 Create an URL mapper for a transaction.

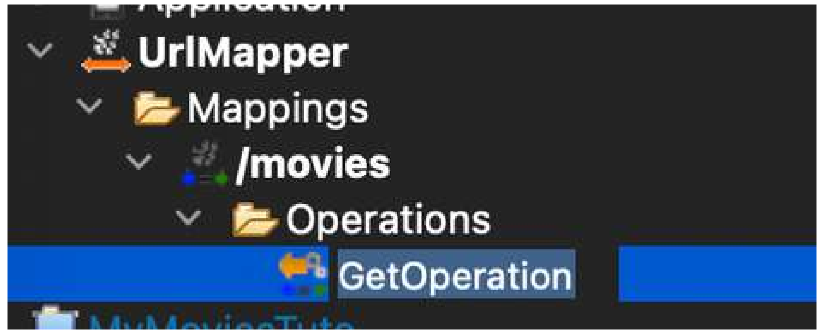
Drag and drop a **GetOperation step** from the palette in the **PathMapping /movies step**.



UrlMapper
Mappings
/movies

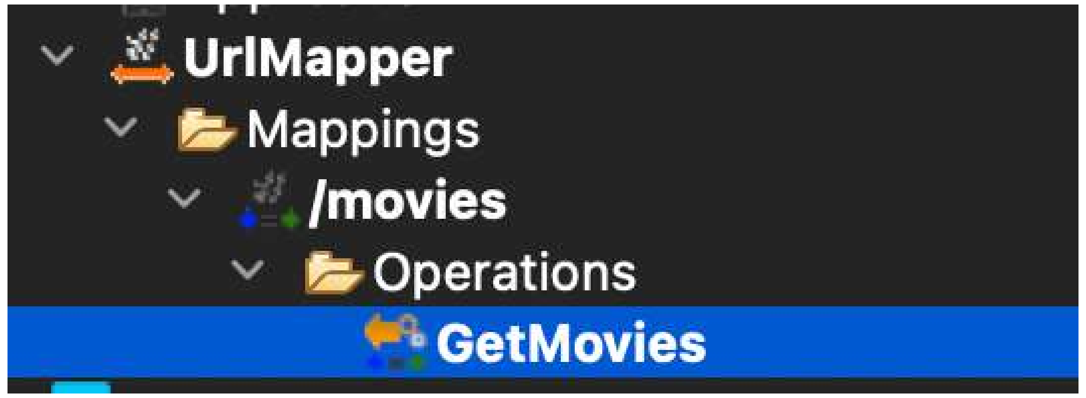


GetOperation



UrlMapper
Mappings
/movies
Operations
GetOperation

Rename the GetOperation step as **GetMovies**.



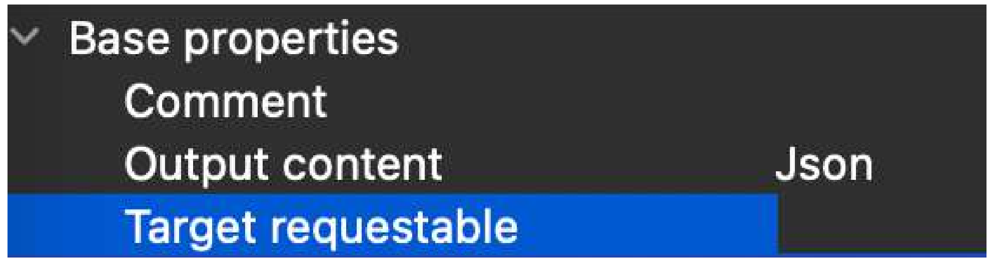
UrlMapper
Mappings
/movies
Operations
GetMovies

9.3 Create an URL mapper for a transaction.

Now, let's select which transaction or sequence we are going to map.

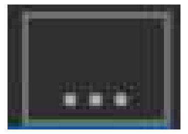
In the properties of GetMovies

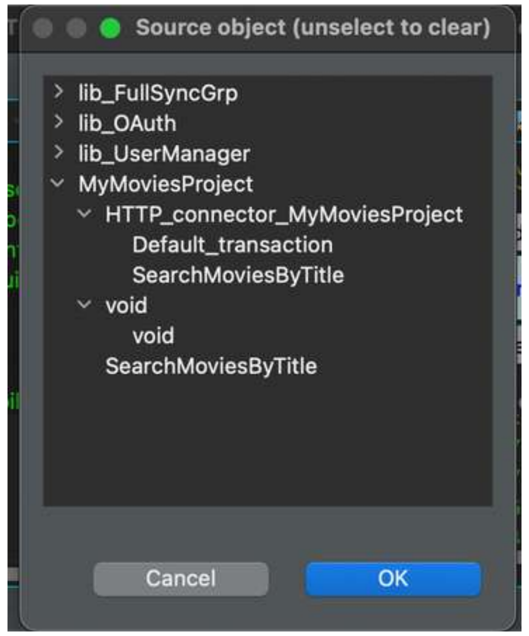
GetMovies



Base properties	
Comment	
Output content	Json
Target requestable	

At the end of the line of the **Target requestable property**, click on this icon.

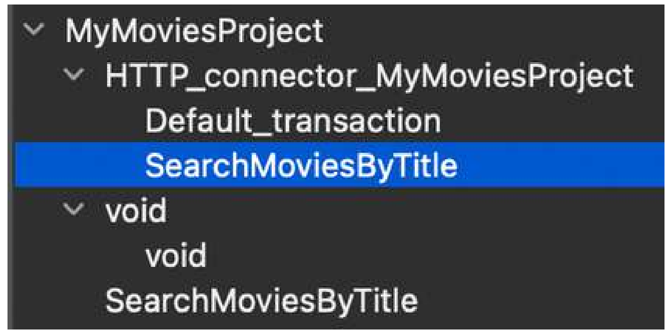



Source object (unselect to clear)

```
> lib_FullSyncGrp
> lib_OAuth
> lib_UserManager
v MyMoviesProject
  v HTTP_connector_MyMoviesProject
      Default_transaction
      SearchMoviesByTitle
  v void
      void
    SearchMoviesByTitle
```

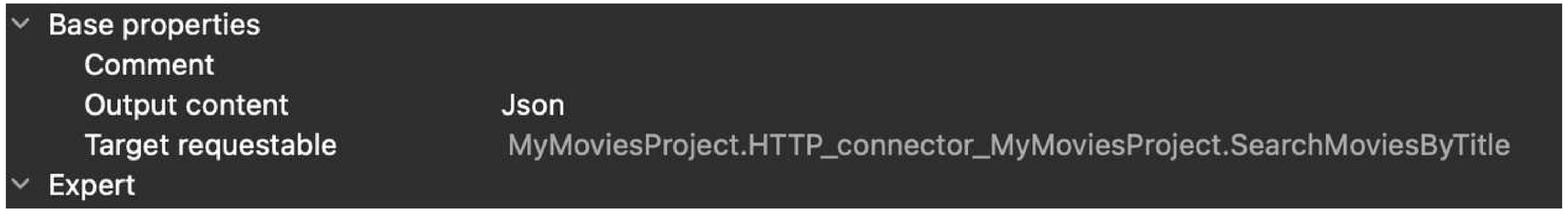
The **Source object window** appears.

Select the **SearchMoviesByTitle transaction**.



```
v MyMoviesProject
  v HTTP_connector_MyMoviesProject
      Default_transaction
      SearchMoviesByTitle
  v void
      void
    SearchMoviesByTitle
```

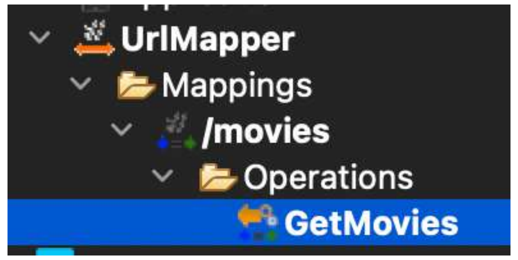
The SearchMoviesByTitle transaction appears as value in the Target requestable property of GetMovies.



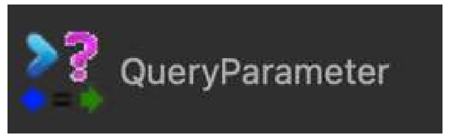
Base properties	
Comment	
Output content	Json
Target requestable	MyMoviesProject.HTTP_connector_MyMoviesProject.SearchMoviesByTitle
Expert	

9.3 Create an URL mapper for a transaction.

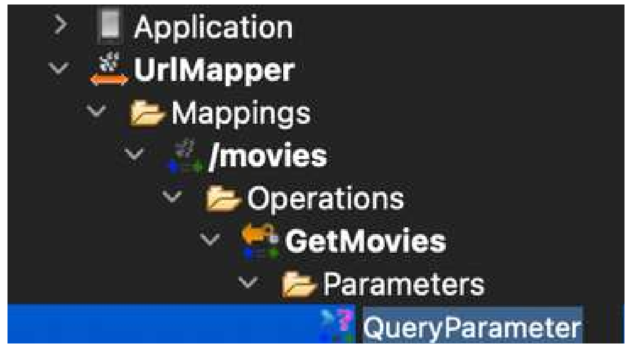
Drag and drop a **QueryParameter step** from the palette in the **GetMovies step**.



UrlMapper
Mappings
/movies
Operations
GetMovies

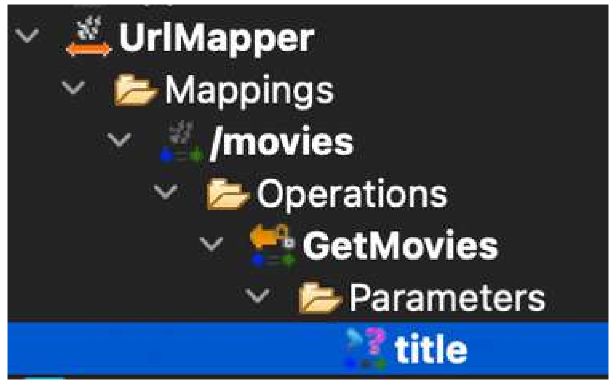


QueryParameter



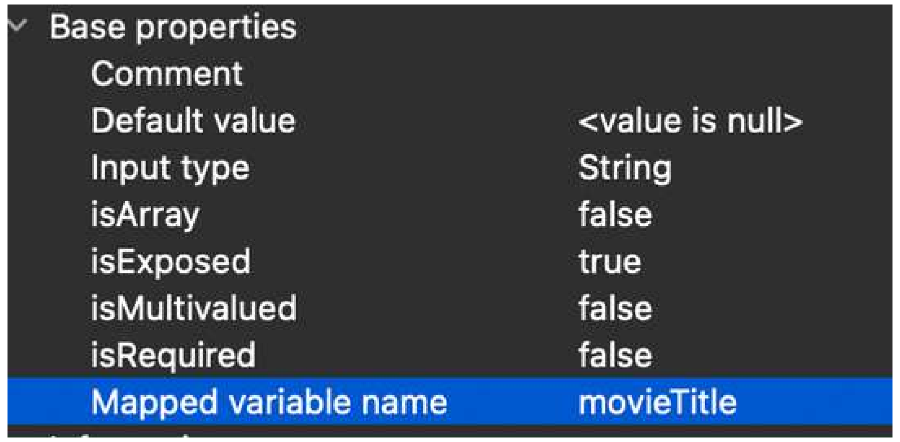
Application
UrlMapper
Mappings
/movies
Operations
GetMovies
Parameters
QueryParameter

Rename the QueryParameter step as **title**.



UrlMapper
Mappings
/movies
Operations
GetMovies
Parameters
title

In the **properties of the QueryParameter**, enter **movieTitle** (transaction variable name) as value of **Mapped variable name**.



Base properties	
Comment	
Default value	<value is null>
Input type	String
isArray	false
isExposed	true
isMultivalued	false
isRequired	false
Mapped variable name	movieTitle

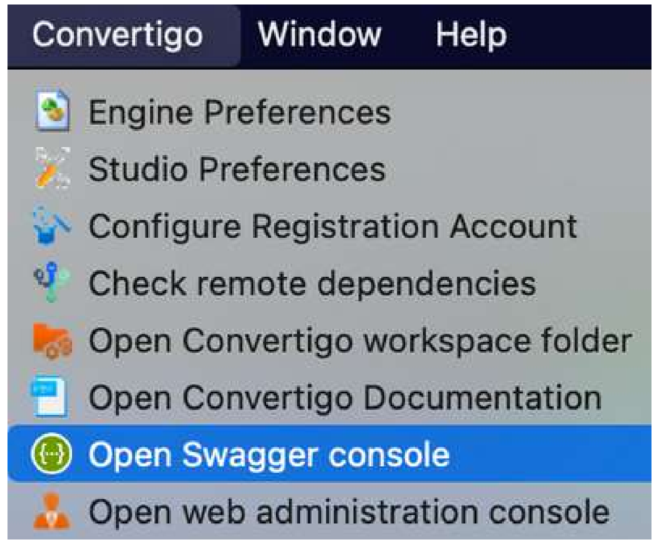
9.4 Test the URL mapper on Swagger

Now, let’s test our URL mapper on Swagger.

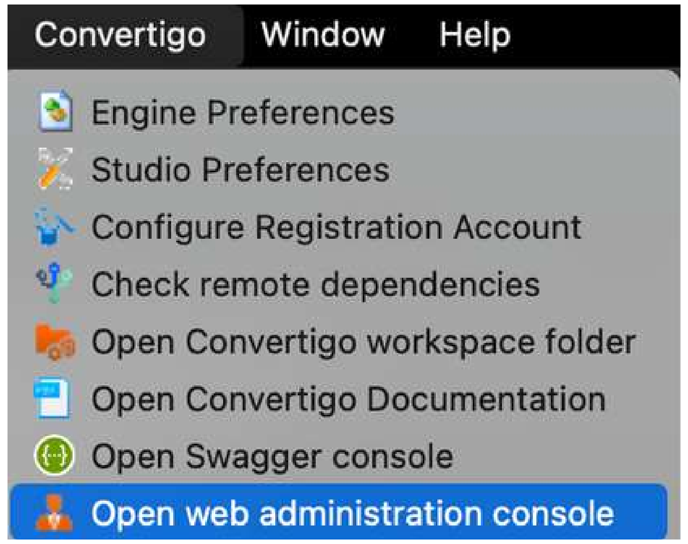
To open the Swagger console in your browser.

Click on

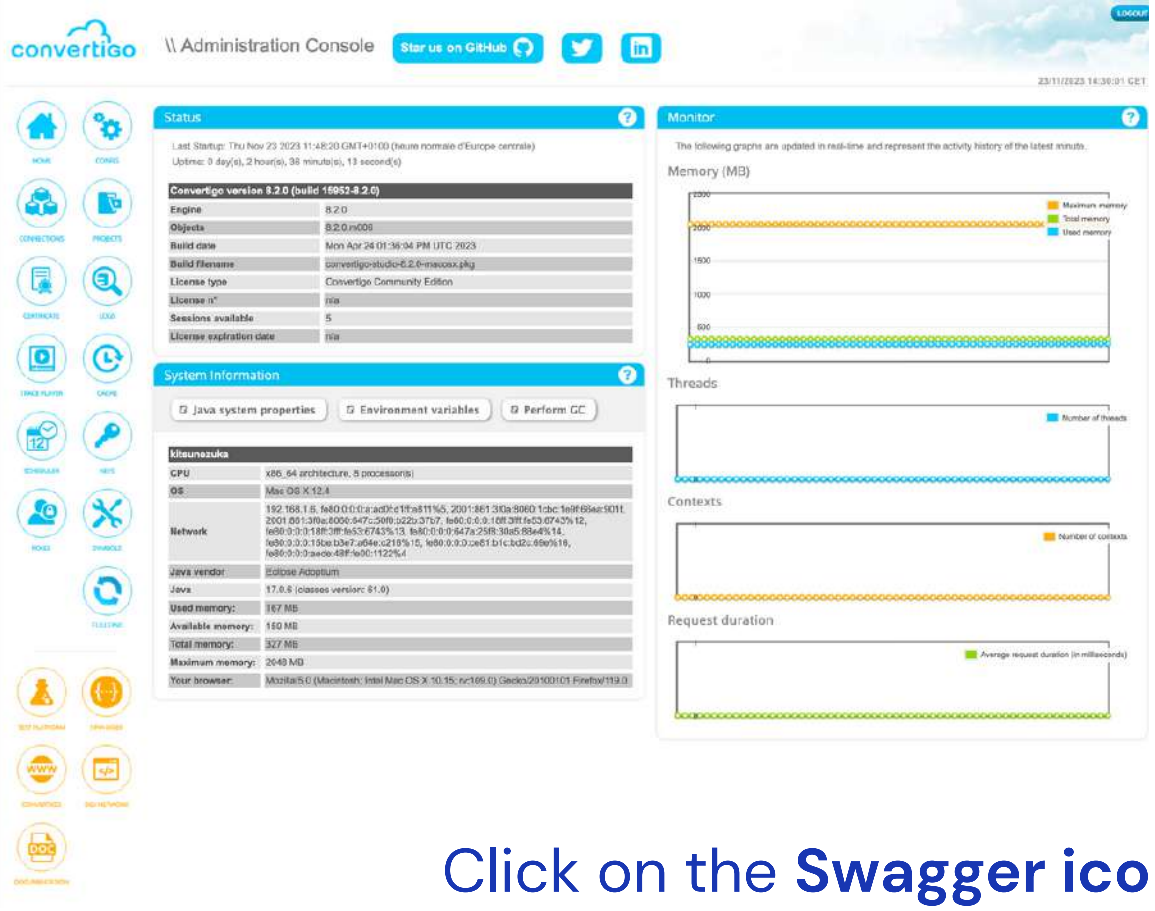
Open Swagger console.


Convertigo
Window
Help
Engine Preferences
Studio Preferences
Configure Registration Account
Check remote dependencies
Open Convertigo workspace folder
Open Convertigo Documentation
Open Swagger console
Open web administration console

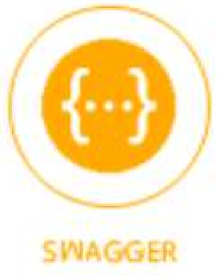
Or open the **web administration console**.


Convertigo
Window
Help
Engine Preferences
Studio Preferences
Configure Registration Account
Check remote dependencies
Open Convertigo workspace folder
Open Convertigo Documentation
Open Swagger console
Open web administration console

In the web administration console



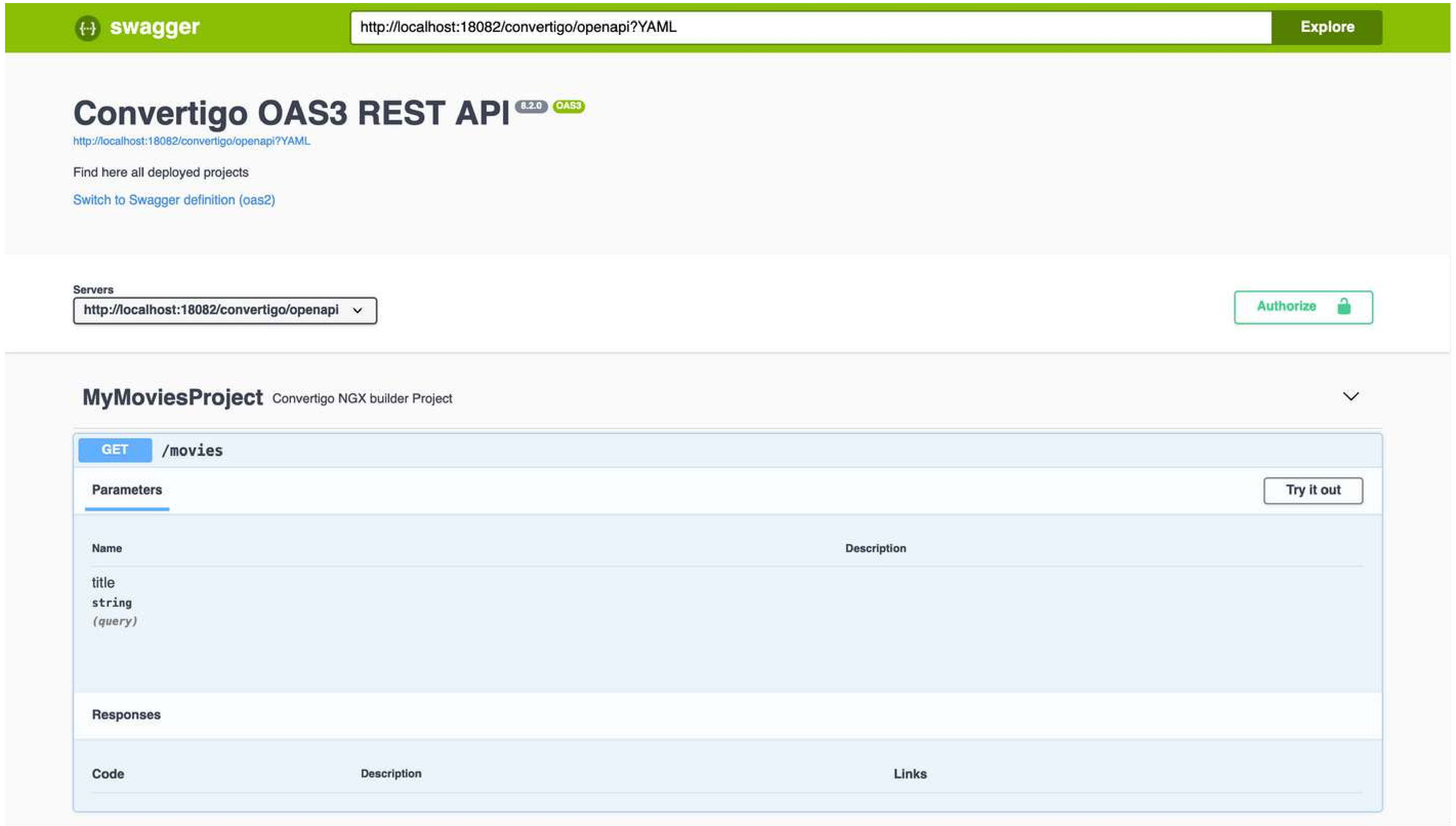
Click on the **Swagger icon**.



9.4 Test the URL mapper on Swagger

In the Swagger console of your browser,

we can see a GET /movies request with a title parameter.



Convertigo OAS3 REST API

http://localhost:18082/convertigo/openapi?YAML

Find here all deployed projects

Switch to Swagger definition (oas2)

Servers: http://localhost:18082/convertigo/openapi

MyMoviesProject Convertigo NGX builder Project

GET `/movies`

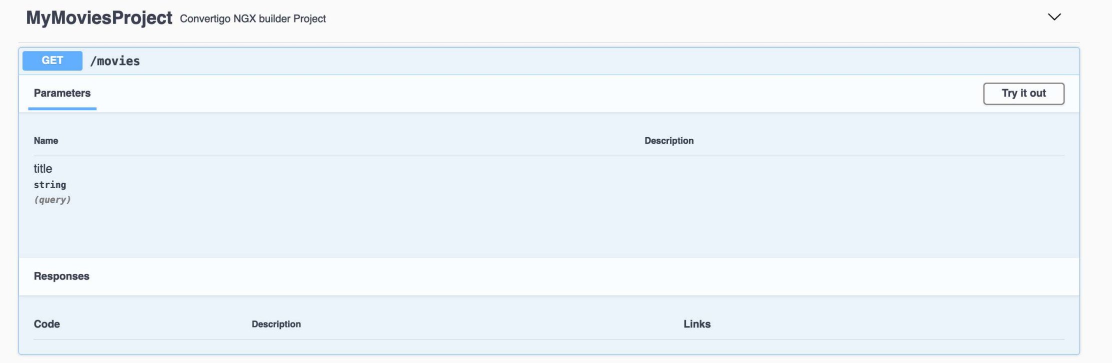
Name	Description
title string (query)	

Responses

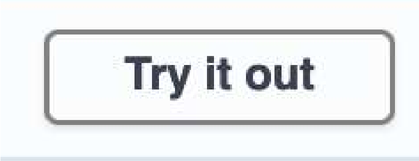
Code	Description	Links

9.4 Test the URL mapper on Swagger

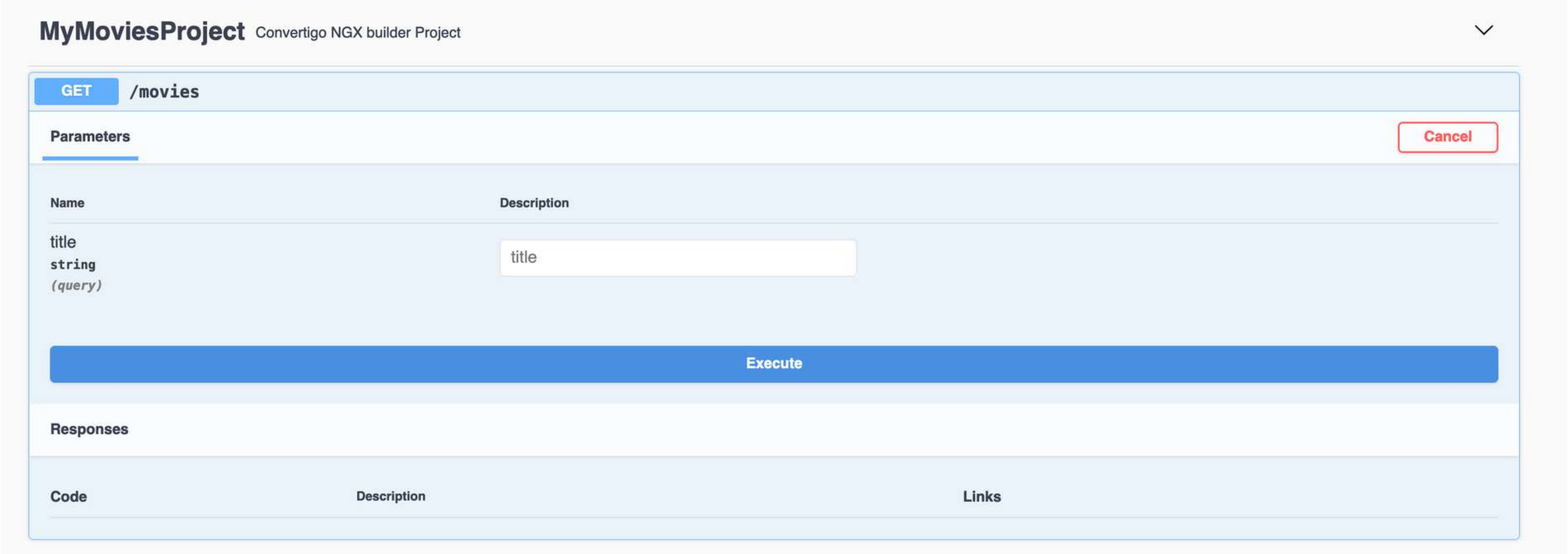
Let's test the GET /movies request with **Try it out**.

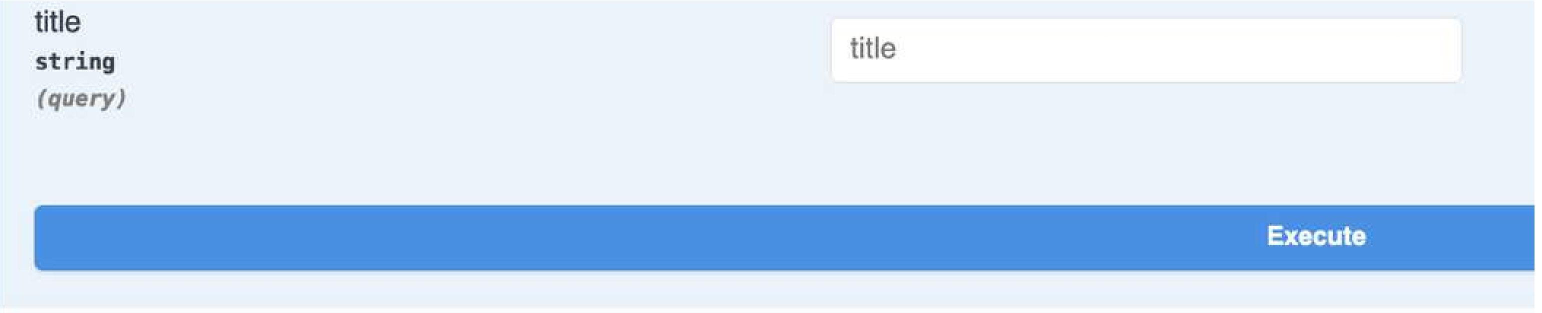


Click on the Try it out button.

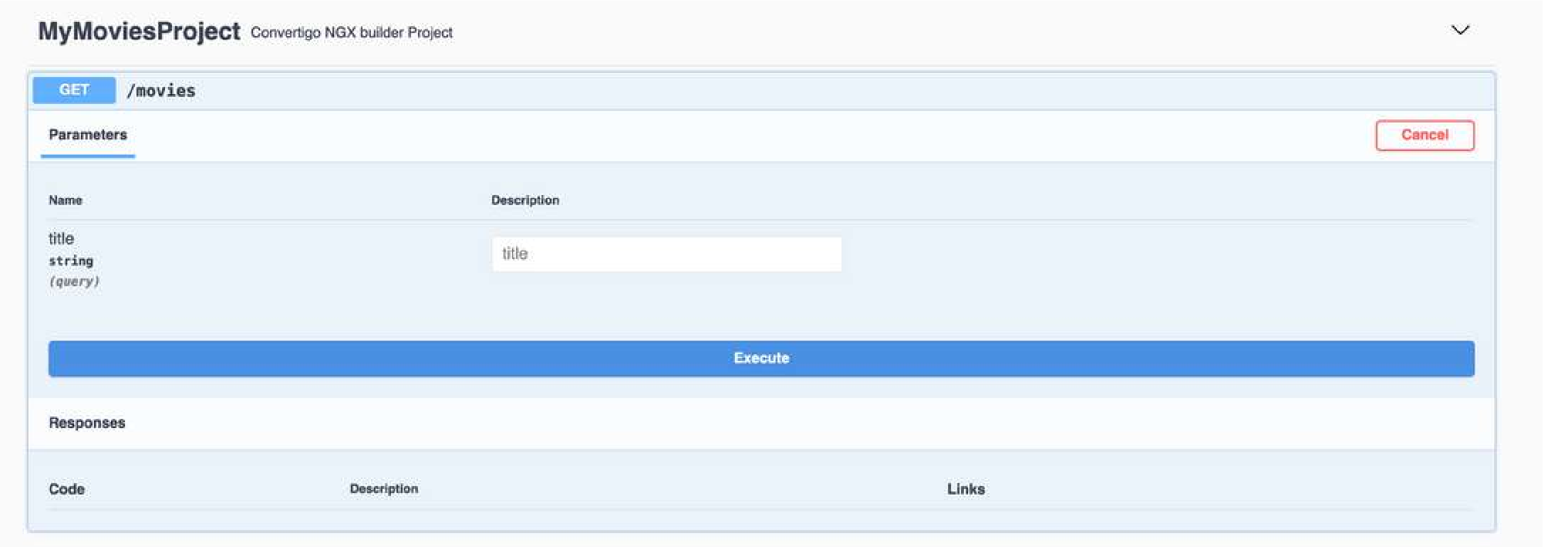


A **title field** and an **Execute button** appear.

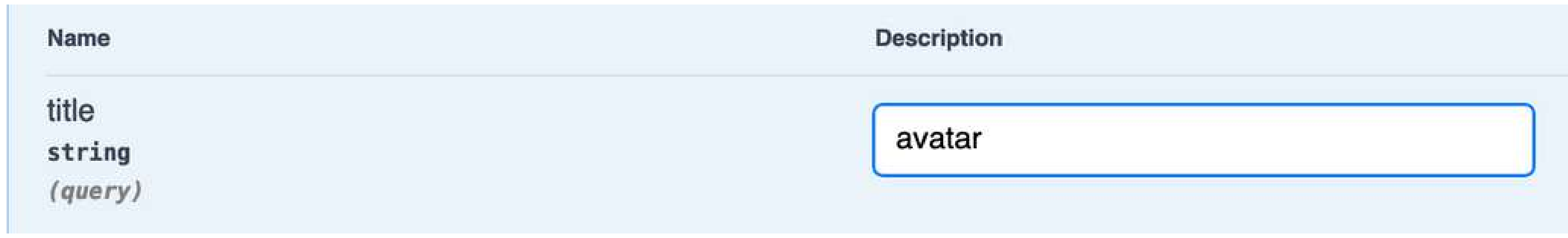




9.4 Test the URL mapper on Swagger



Enter a value in the title field (here “avatar”).

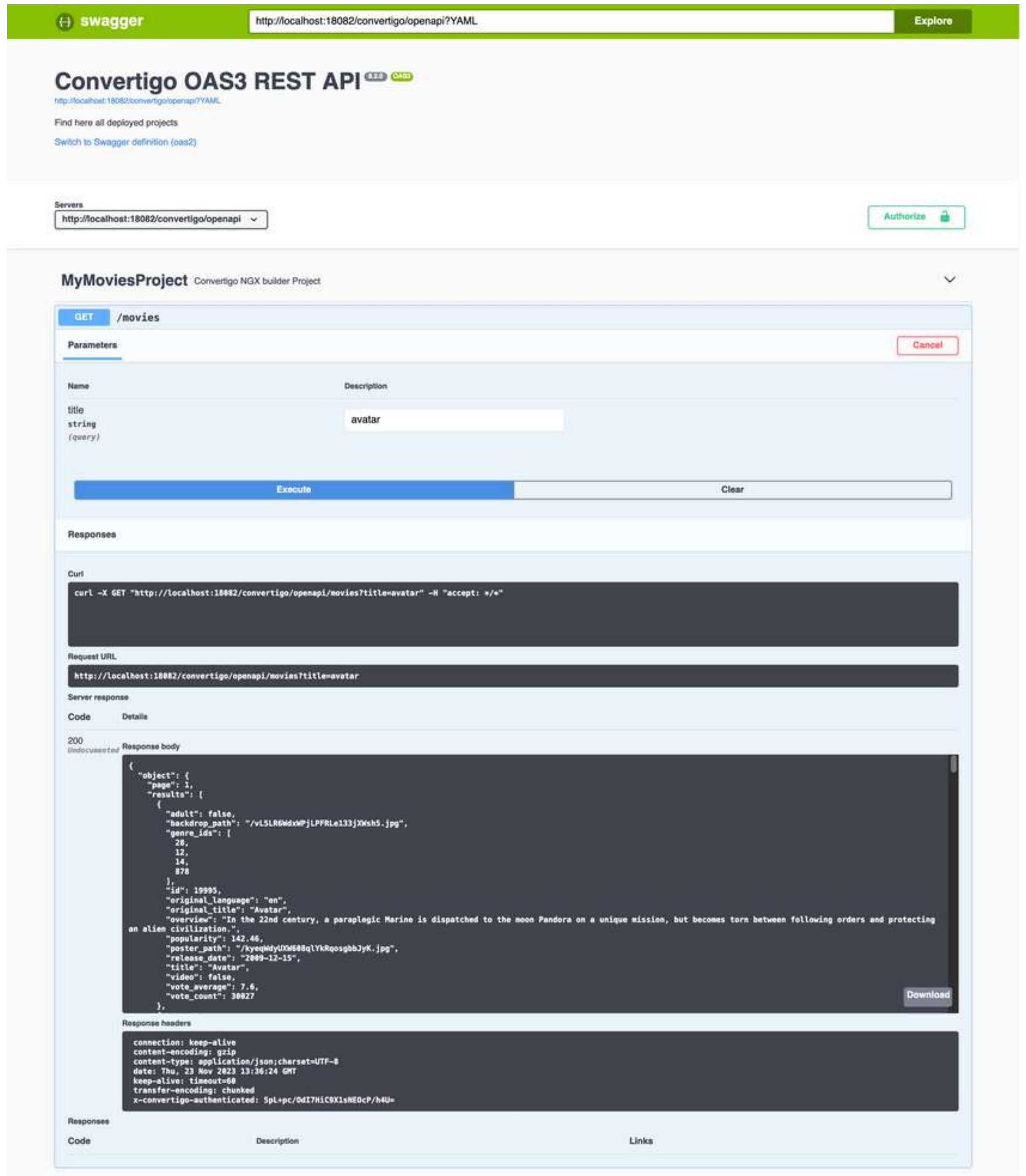


Name	Description
title string *(query)*	avatar

Click on Execute

9.4 Test the URL mapper on Swagger

A response result of the GET /movies request appears in the Swagger



swagger http://localhost:18082/convertigo/openapi?YAML Explore

Convertigo OAS3 REST API

http://localhost:18082/convertigo/openapi?YAML

Find here all deployed projects

Switch to Swagger definition (oas2)

Servers
http://localhost:18082/convertigo/openapi

Authorize

MyMoviesProject Convertigo NGX builder Project

GET /movies

Parameters Cancel

Name	Description
title string *(query)*	avatar

Execute Clear

Responses

Curl

```
curl -X GET "http://localhost:18082/convertigo/openapi/movies?title=avatar" -H "accept: */*"
```

Request URL

```
http://localhost:18082/convertigo/openapi/movies?title=avatar
```

Server response

Code Details

200 *Undocumented* Response body

```
{
  "object": {
    "page": 1,
    "results": [
      {
        "adult": false,
        "backdrop_path": "/vL5LR6WdxWPjLPFRLe133jXWsh5.jpg",
        "genre_ids": [
          28,
          12,
          14,
          878
        ],
        "id": 19995,
        "original_language": "en",
        "original_title": "Avatar",
        "overview": "In the 22nd century, a paraplegic Marine is dispatched to the moon Pandora on a unique mission, but becomes torn between following orders and protecting
an alien civilization.",
        "popularity": 142.46,
        "poster_path": "/kyeqWdyUXW608qlYkRqosgbbJyK.jpg",
        "release_date": "2009-12-15",
        "title": "Avatar",
        "video": false,
        "vote_average": 7.6,
        "vote_count": 30027
      },
```

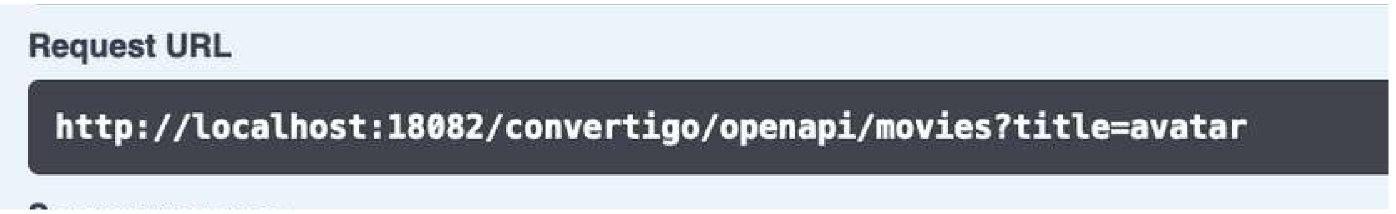
Download

Response headers

```
connection: keep-alive
content-encoding: gzip
content-type: application/json;charset=UTF-8
date: Thu, 23 Nov 2023 13:36:24 GMT
keep-alive: timeout=60
transfer-encoding: chunked
x-convertigo-authenticated: 5pL+pc/OdI7HiC9X1sNEOcP/h4U=
```

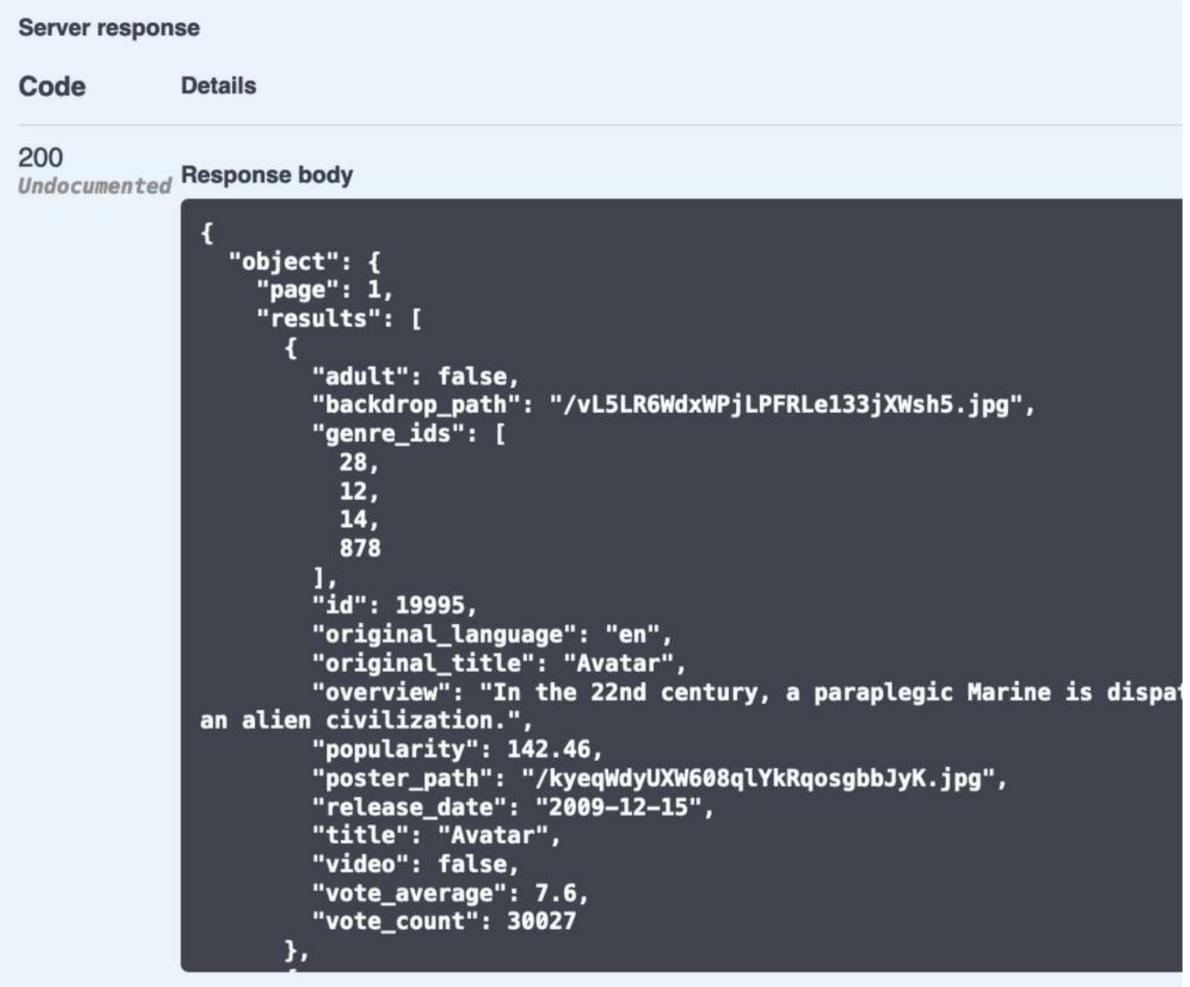
Responses

Code Description Links



Request URL

```
http://localhost:18082/convertigo/openapi/movies?title=avatar
```



Server response

Code Details

200 *Undocumented* Response body

```
{
  "object": {
    "page": 1,
    "results": [
      {
        "adult": false,
        "backdrop_path": "/vL5LR6WdxWPjLPFRLe133jXWsh5.jpg",
        "genre_ids": [
          28,
          12,
          14,
          878
        ],
        "id": 19995,
        "original_language": "en",
        "original_title": "Avatar",
        "overview": "In the 22nd century, a paraplegic Marine is dispat
an alien civilization.",
        "popularity": 142.46,
        "poster_path": "/kyeqWdyUXW608qlYkRqosgbbJyK.jpg",
        "release_date": "2009-12-15",
        "title": "Avatar",
        "video": false,
        "vote_average": 7.6,
        "vote_count": 30027
      },
```