Exercise - REST Connector

In this exercise, you will create a new *Entertain Me* process which contains a single *Generate Joke* task.

The *Generate Joke* task will use the **REST Connector** to obtain a joke from a publicly available Joke API.

Create Process

Create a new **Process Definition** by following the steps below:

1. Launch the **Web Modeler**
2. Create a new **BPMN Diagram**
3. This will open a new **BPMN Diagram** in the **Modeler**. In the properties panel on the right-hand side of the screen, set the **Name** and **ID** as follows:

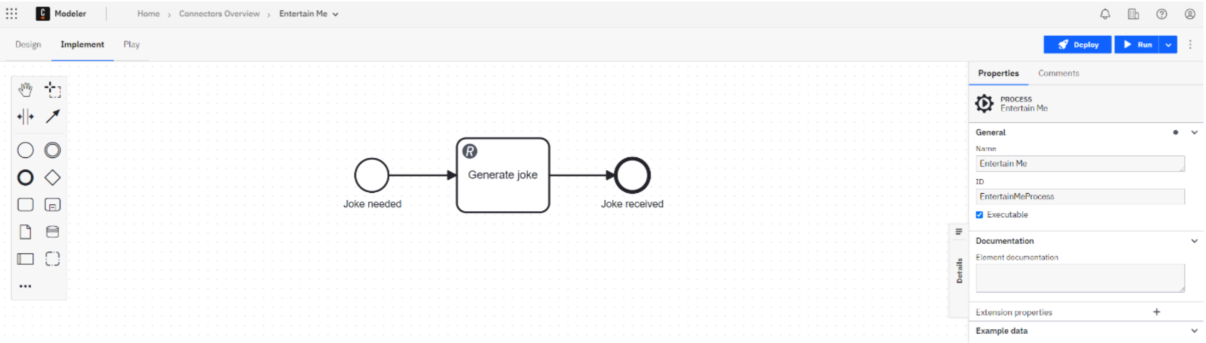
Name	Value
Name	Entertain Me
ID	EntertainMeProcess

   !!! info "Implement" Mode The Web Modeler offers different modes for easier collaboration between business and IT. To access the properties panel and later to deploy and start your process, make sure to stay in **Implement** mode for this exercise.

4. Model a **Start Event**, **Task** and **End Event** with the following properties:

Element	Name	ID
Start Event	Joke needed	StartEvent_JokeNeeded
Task	Generate joke	Task_GenerateJoke
End Event	Joke received	EndEvent_JokeReceived

5. Your process should look like the diagram below once you have connected the sequence flows:



PROCESS
Entertain Me

Name: Entertain Me

ID: EntertainMeProcess

☑ Executable

Create Joke Process

Configure the Generate Joke Task

You will now need to configure the *Generate Joke* Task to use the **REST Connector**.

1. Select the *Generate Joke* task, click on the **+Select** button in the **Template** section of the properties panel and choose **REST Connector** from the menu that appears.
2. Configure the **Authentication** properties as follows:

Name	**Value**
Type	None

Authentication

In this example, we are using an open API which does not require authentication.

3. Configure the **HTTP Endpoint** properties as follows:

Name	**Value**
Method	GET
URL	https://official-joke-api.appspot.com/jokes/programming/random

API Endpoint

In this example, we will send a GET request to the **Official Joke API**. We will use the programming category to request a random joke.

4. Configure the **Connection** properties as follows:

Name	Value
Connection Timeout	20

Connection Timeout

The default value for the request timeout is 20 seconds; an incident will be created in **Operate** if the request takes longer than this.

5. The snippet below contains an example response from a request to the Joke API. We will extract the relevant elements from this response in the next step.

```
{
 "status":200,
 "headers":
 {
  "date":"Thu, 01 Jan 1970 00:00:00 GMT",
  "access-control-allow-origin":"*",
  "server":"Google Frontend",
  "transfer-encoding":"chunked",
  "vary":"Accept-Encoding",
  "x-cloud-trace-context":"qwertzui1234567;o=1",
  "x-powered-by":"Express",
  "content-type":"application/json; charset=utf-8",
  "etag":"W/\"a1-qwert123456\"",
  "cache-control":"private",
  "alt-svc":"h3=\":443\"; ma=2592000,h3-29=\":443\"; ma=2592000,h3-Q050=\":443\";"
 },
 "body":
 [
  {
   "type":"programming",
   "setup":"Why do C# and Java developers keep breaking their keyboards?",

   "punchline":"Because they use a strongly typed language.",

   "id":74

  }

 ]

}
```

6. Configure the **Result Expression** of the **Response Mapping** property with the following JSON:

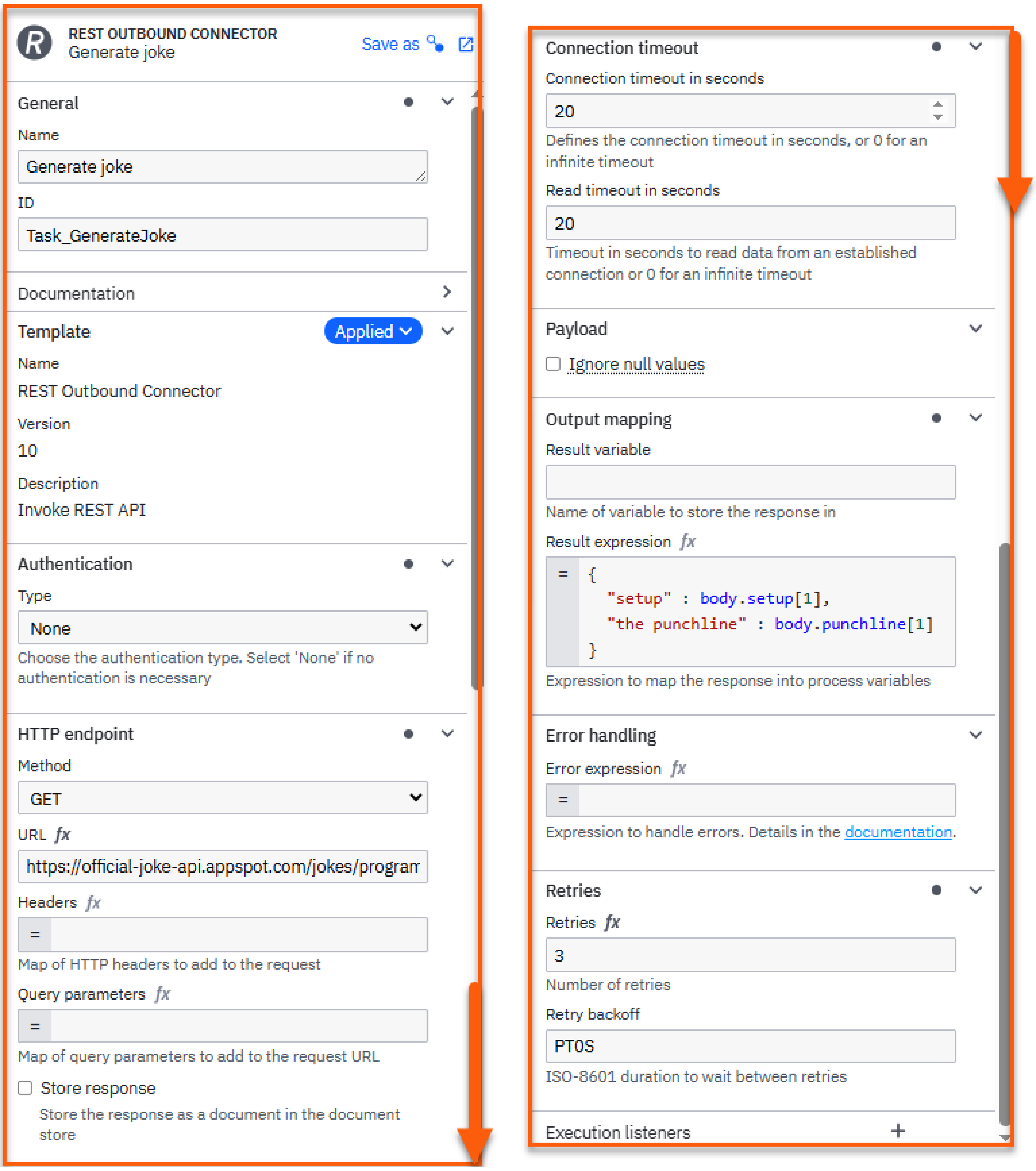
```
{

 "setup" : body.setup[1],

 "the punchline" : body.punchline[1]

}
```

Result Expression

If we use a **Result Variable**, the Connector will simply store the entire response for the request in that variable. It is usually better to extract only the parts of the result which are relevant for the process. This is possible using the **Result Expression**.

In the **Result Expression**, we have defined two new variables (setup and punchline). The value of these variables is initialized from the first element of the JSON array returned in the body of the response.

7. The properties of the **Create Joke** task should look like this now:



REST OUTBOUND CONNECTOR
Generate joke

General

Name: Generate joke

ID: Task_GenerateJoke

Documentation

Template

Name: REST Outbound Connector

Version: 10

Description: Invoke REST API

Authentication

Type: None

Choose the authentication type. Select 'None' if no authentication is necessary

HTTP endpoint

Method: GET

URL *fx*: https://official-joke-api.appspot.com/jokes/prograr

Headers *fx*: =

Map of HTTP headers to add to the request

Query parameters *fx*: =

Map of query parameters to add to the request URL

□ Store response

Store the response as a document in the document store

Connection timeout

Connection timeout in seconds: 20

Defines the connection timeout in seconds, or 0 for an infinite timeout

Read timeout in seconds: 20

Timeout in seconds to read data from an established connection or 0 for an infinite timeout

Payload

□ Ignore null values

Output mapping

Result variable:

Name of variable to store the response in

Result expression *fx*

```
= {
    "setup" : body.setup[1],
    "the punchline" : body.punchline[1]
  }
```

Expression to map the response into process variables

Error handling

Error expression *fx*: =

Expression to handle errors. Details in the documentation.

Retries

Retries *fx*: 3

Number of retries

Retry backoff: PT0S

ISO-8601 duration to wait between retries

Execution listeners +

REST Connector properties

Test the Process

You can now test the **REST Connector** by deploying your process and starting an instance.

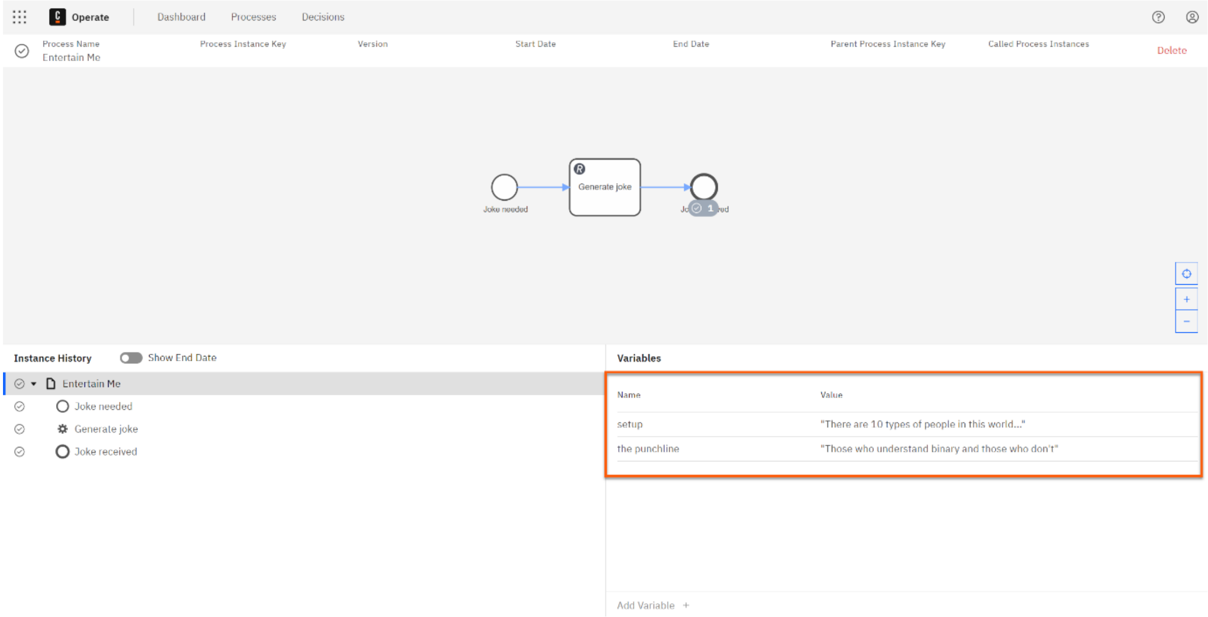
1. Deploy the process to your cluster by clicking the **Deploy Diagram** button
2. Click the **Run** button to start a new instance of the process
3. Launch **Operate** and select the *Entertain Me* process from **Process Instances by Name**

Operate

You can switch to **Operate** using the **App Switcher** in the top left corner of the screen. Alternatively, you can click on the **Process Started** pop-up at the bottom of the screen which is displayed for a few seconds after starting your instance.

4. Click on the **Process Instance Key** for the process that you have just started; it should have completed immediately

5. You should be able to see that your process instance has completed successfully and the setup and punchline variables have been populated as **Process Variables**.



Process Name	Process Instance Key	Version	Start Date	End Date	Parent Process Instance Key	Called Process Instances
Entertain Me						

Instance History

- Entertain Me
  - Joke needed
  - Generate joke
  - Joke received

Variables

Name	Value
setup	"There are 10 types of people in this world..."
the punchline	"Those who understand binary and those who don't"

Completed Joke Process

Configured the REST Connector

Congratulations! You have created a process which uses the **REST Connector** to obtain data from an external service.

Solution - REST Connector

If you encountered any problems during this course or were unable to complete the process configuration, you can compare your solution with our example solution below.

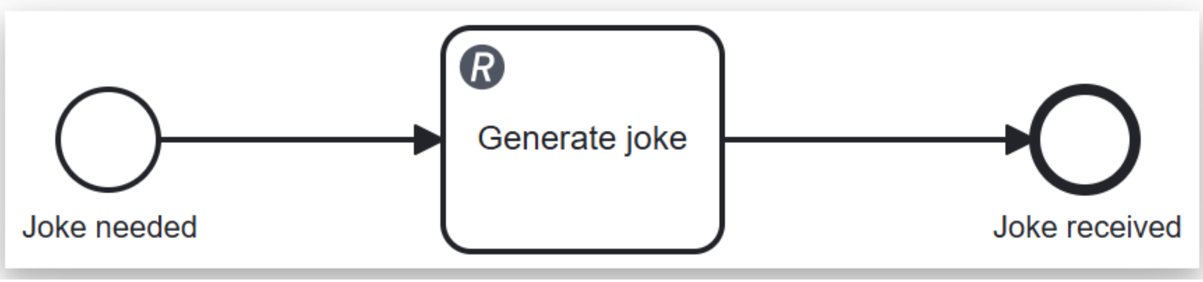
GitHub Repository

Our version of the **Process Definition** including the **REST Connector** configuration can be found in the GitHub Repository below:

- Solution - Entertain Me

Process Diagram

Our version of the **Process Diagram** can be found below:


Joke needed
R
Generate joke
Joke received