

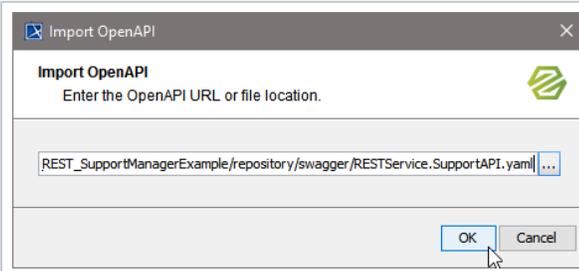
# Importing OpenAPI Files REST

You can import an OpenAPI file describing a REST interface directly from a file or via a URL. The importer will generate all necessary interface elements to your UML model to use this REST service with the REST Adapter.

At the moment, the import of **OpenAPI 3** specifications is not supported. You can use a tool to convert OpenAPI 3 into OpenAPI 2 and then import the result. A useful tool for this is <https://www.npmjs.com/package/api-spec-converter>. You can find more tools at <https://openapi.tools/>.

The OpenAPI import rules are described in detail in [REST Import Rules](#).

To import an OpenAPI for REST, select **Import > OpenAPI** from the Model Compiler menu.



Enter a file location or an URL. By clicking  you can use the **File Chooser** to lookup the file.

Note the following for **Bridge REST services**: You can get the URL to the OpenAPI file from the Bridge as described on [x UML Service Details](#) (see **REST service Interface**). Copy the link location or download the file.

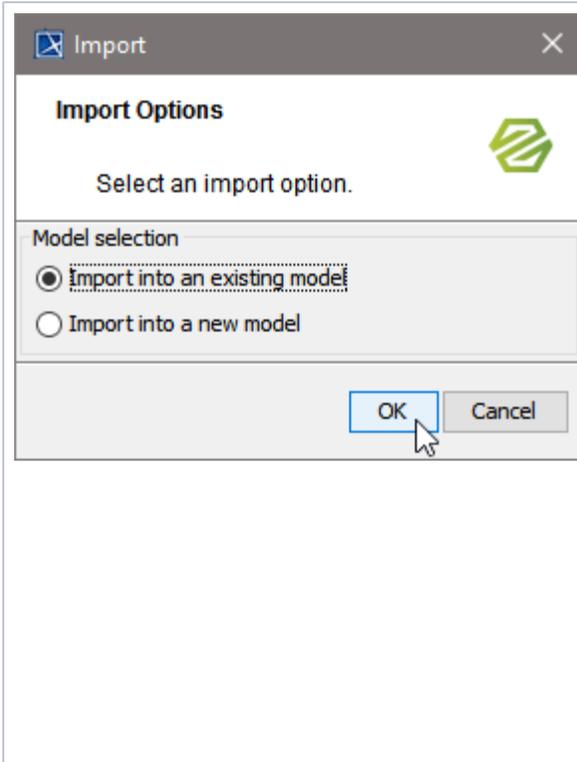
Click **OK**.

## On this Page:

- [Import Problems](#)
  - [The REST Service Only Provides a JSON Definition File](#)
  - [The YAML File Does Not Set All Necessary Options](#)
  - [The Import of the YAML File Fails Or The REST Service Provides No Description File At All](#)

## Related Pages:

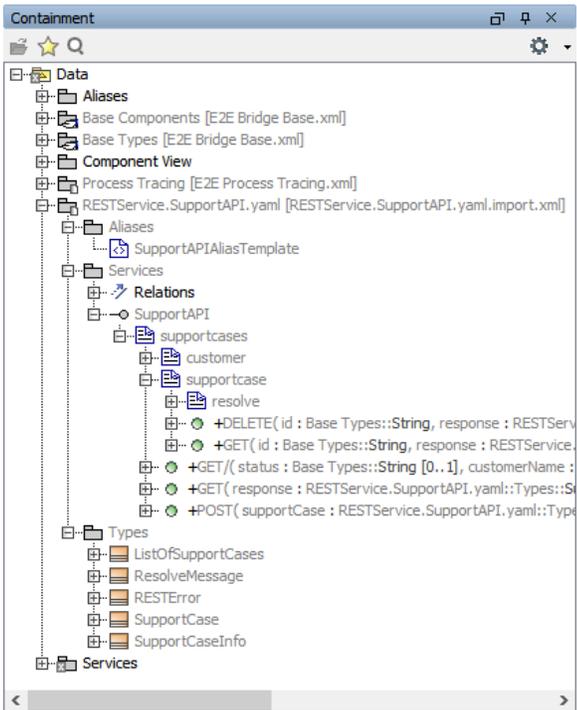
- [REST Adapter](#)
- [REST Import Rules](#)



The selected OpenAPI file can be imported into either an **existing** or a **new** UML model. If you choose to import to a new model, the REST Importer will import the OpenAPI file to a generated simple test model you can easily test all imported REST resources with.

Click **OK** to import the definitions. You can still cancel the import by clicking **Cancel**.

The importer generates the interface definitions to a new repository package having the name of the import file.



**Creation of aliases and alias templates:** During the first import into an existing model, aliases and alias templates are created. When a file is re-imported, the alias templates are changed and missing aliases are recreated. Existing aliases remain unchanged in this case. While importing into a new model, aliases and alias templates are created.

## Import Problems

## The REST Service Only Provides a JSON Definition File

The xUML REST Importer can import YAML files only. If the service you want to call provides an OpenAPI file in JSON only, you can convert the JSON file to YAML before importing it. You can find many JSON to YAML converters on the internet.

## The YAML File Does Not Set All Necessary Options

After having imported a YAML file, it may be that some special options (like e.g. **isVerbatimPath** or **external Name**) that are not part of the official Open API description, are not set, but may be necessary. Also, blob parameters, **Blob Body Content Type/Reject Other Response Content Type** and **Accepted Request Content Type/Reject Other Request Content Types** as described on [Handling Blobs in the REST Interface](#) are not supported by the importer.

In these cases, you will have to edit the import file manually to add blob parameters and set these options.

## The Import of the YAML File Fails Or The REST Service Provides No Description File At All

It may be that the import of the YAML file fails or the REST service provides no description file at all. In this case, you can

- draw the definitions manually by following the descriptions at [Manually Providing the REST Interface ...](#)
- try an API design tool, the [API Design Tool of Swagger](#). You can either check and rectify the YAML file here or write a new YAML file matching the service interface.