Rest API

Overview

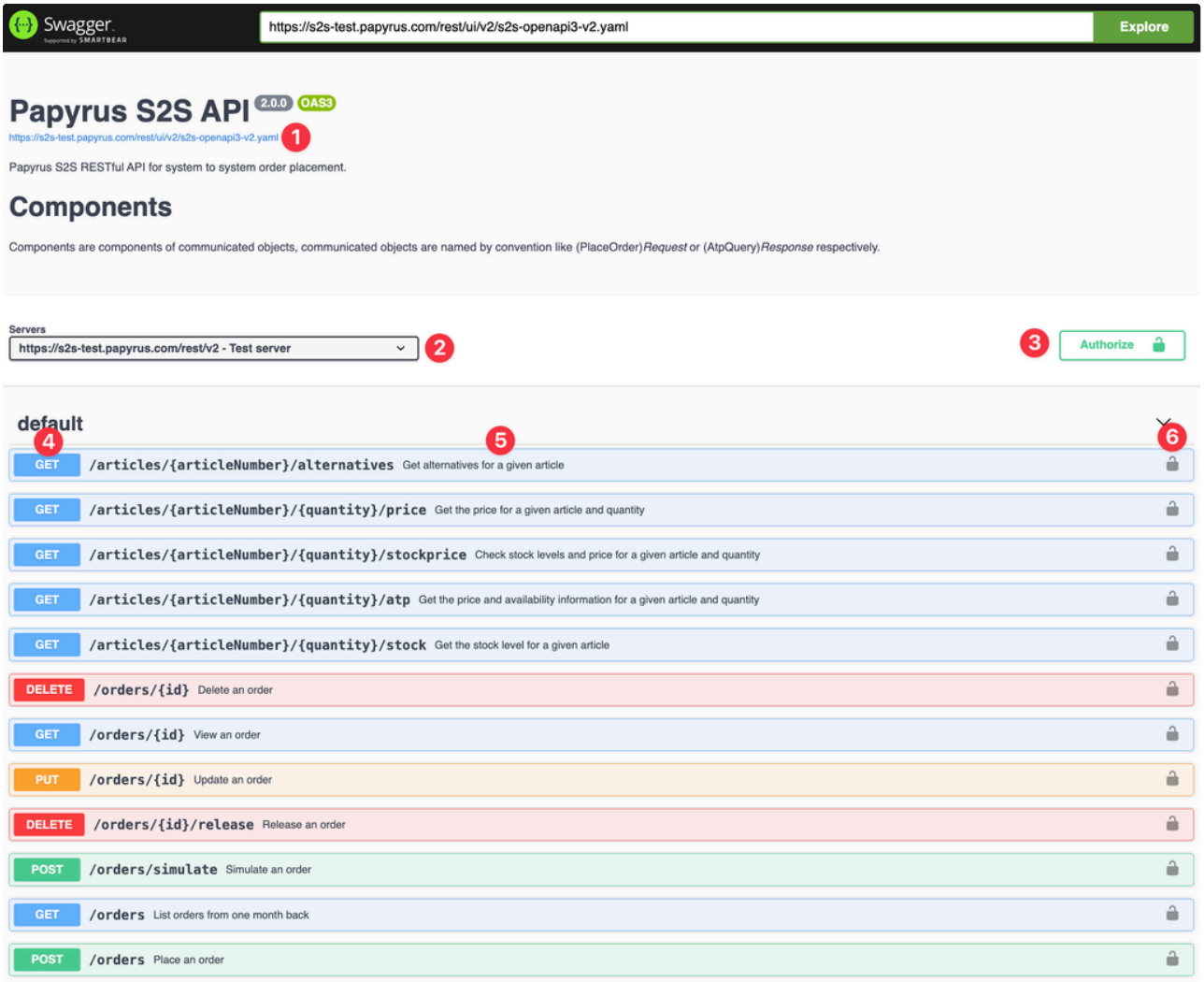
Papyrus REST API is built using the specification for The specification can be reached at OpenApi 3.0.

Swagger UI

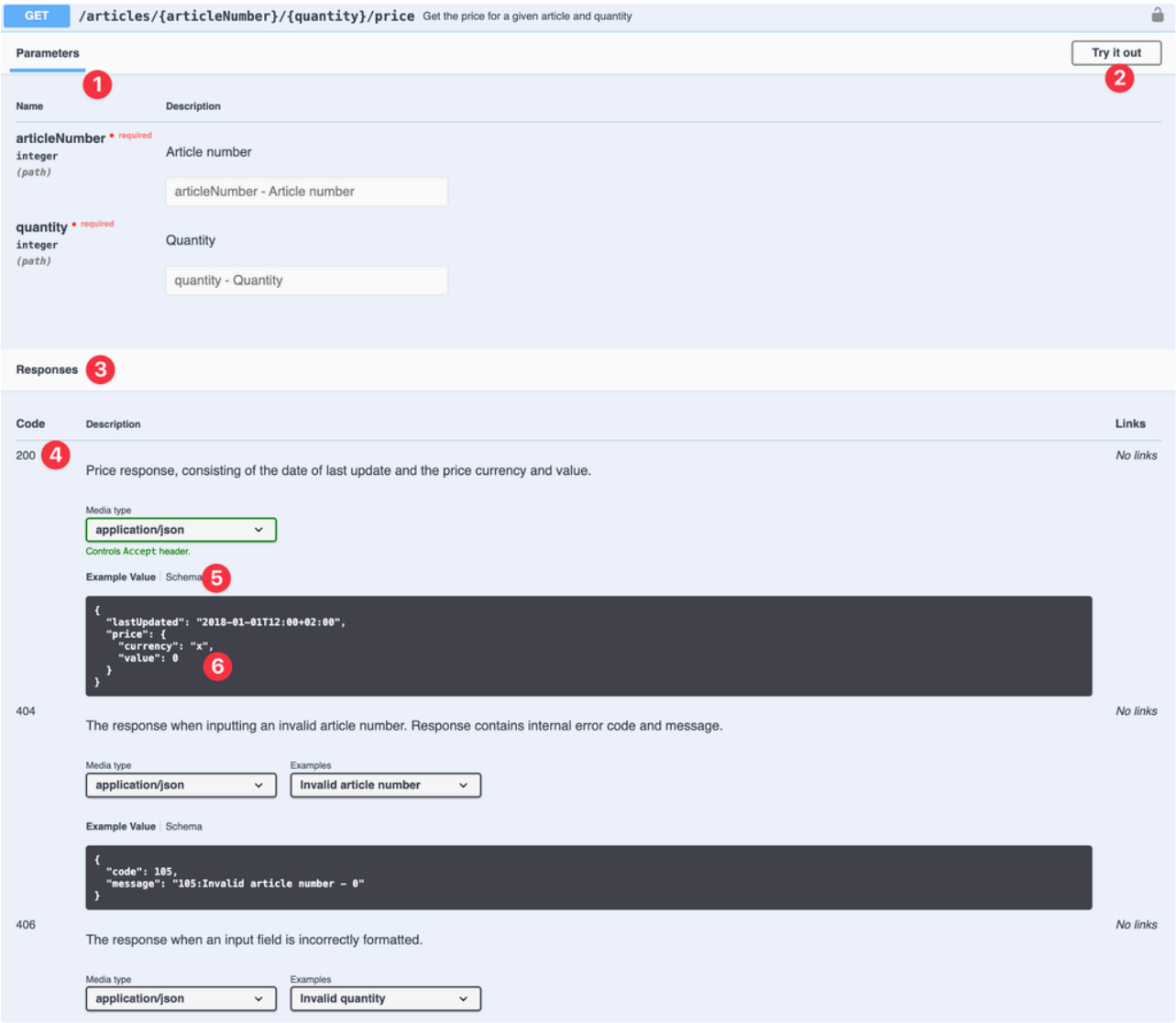
Quick start

To test out the functionality, you can access the interactive documentation directly. Access to the specification documentation can be found at

Swagger UI



1. General documentation. Since this is a live document, improvements made to the documentation, for example after feedback from customers, will update immediately and so this might look different when viewed directly.
2. S2S servers list
    a. https://s2s-test.papyrus.com - should be used for testing purposes,
    b. https://s2s.papyrus.com - should be used for production.
3. Authorize
    a. All requests must be authorized, and will not work otherwise. Clicking this will authenticate your browser session so you can actually try out requests toward the live system in the browser.
    b. Details on username and password will be provided to you by your customer contact.
4. Request METHOD.
5. The request mapping structure, excluding query parameters. Query parameters or POST body content can be viewed if you click anywhere within the request's borders.
6. Secured request, we have no anonymous methods. In other words, authentication is required for all requests.



1. Parameter list, in this example
    a. **articleNumber**, integer (path) - the articleNumber placeholder in the path should be replaced with the value of this parameter. The value is an integer.
    b. **quantity**, integer (path) - same as above
    c. example URL would be: */articles/88010611/100/price*
2. Clicking here will display input boxes next to each of the defined parameters along with an Execute button
    a. Try it out!
    b. Trying out means actually making a real request and getting a response.
3. Responses. This section describes possible responses for a call.
4. Response code, HTTP status.
5. Example value/Schema toggle - click to toggle between displaying the model returned and an example JSON structure of a return value.
6. An example return value for, in this case, a price query.

Feel free to explore the interactive documentation further.

Quick start - your client

Usage of the system requires article lists in the form of "price files". See 2.4 Price files

To aid your development, you need the specification itself, which is downloaded from

- API version 1
  - https://s2s-test.papyrus.com/rest/ui/s2s-openapi3-v1.yaml
- API version 2
  - https://s2s-test.papyrus.com/rest/ui/s2s-openapi3-v2.yaml

> Shared components & paths definitions file is also required to download in case of creating local REST client:
>
> - https://s2s-test.papyrus.com/rest/ui/s2s-shared-api.yaml

The URL to the current specification can always be seen on

- Swagger UI

OpenAPI Generetor CLI can generate the code for your client based on the specification. It can be installed using NPM, Homebrew, Docker or other tools.

- CLI Installation | OpenAPI Generator
- Generators List | OpenAPI Generator

Java Client example

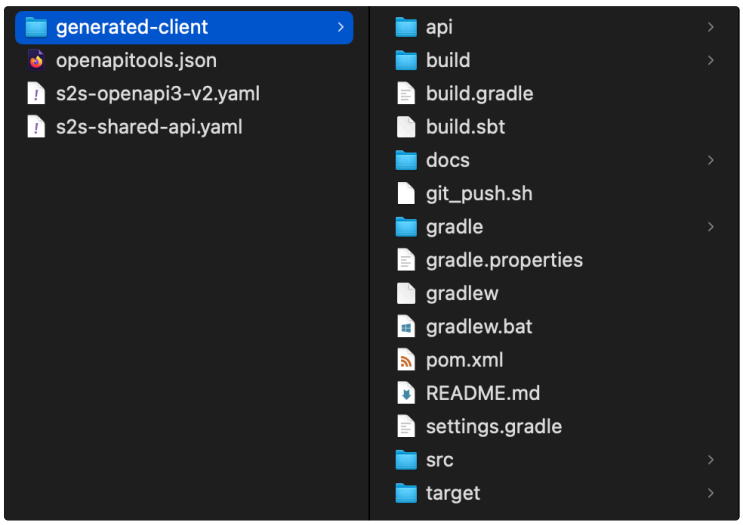
Ensure that mentioned yaml files were downloaded (assuming v2) and are in the same disc location:

- https://s2s-test.papyrus.com/rest/ui/s2s-shared-api.yaml
- https://s2s-test.papyrus.com/rest/ui/s2s-openapi3-v2.yaml

Used OpenAPI Generetor CLI to generate project:

```
openapi-generator-cli generate -i s2s-openapi3-v2.yaml  -g java -o ./generated-client --additional-properties=library=resttemplate
```

It will generate the Java project using RestTemplate with structure which can be imported using IDE (eg In)Java project using RestTemplate with structure which can be imported using IDE (eg IntelliJ Idea):



After setting up the project, it could be simply tested by the following code.

Username, password, articleNumber - should be changed to use the real credentials (provided by Your contact) and valid articleNumber.

```java
package org.openapitools.client;

import org.openapitools.client.api.DefaultApi;
import org.openapitools.client.model.AlternativeArticlesResponse;
import org.springframework.web.client.RestTemplate;

public class Main {

    public static void main(String[] args) {
        final ApiClient apiClient = new ApiClient(new RestTemplate());
        apiClient.setUsername("Aladdin");
        apiClient.setPassword("open sesame");

        final DefaultApi defaultApi = new DefaultApi(apiClient);

        final long articleNumber = 1L;
        final AlternativeArticlesResponse alternativeArticlesResponse =
defaultApi.alternativeArticles(articleNumber);

        System.out.println(alternativeArticlesResponse);
    }
}
```

Limitations

There might be limitations on functionality that can actually be used by your client based on the enabled functionality for the customer using the client.

Such limitations and what they are must be discussed with your customer representative.

Price files

In order to use the S2S Web Services interface, the external application will need information about Papyrus's assortment, e.g. article numbers. This is available in so called price files which are available in various machine-readable format. A price file contains at a minimum article names and numbers, but may (depending on the format) also contain additional product and category information. It may also contain customer unique prices.

Customers manually download their price files from Papyrus' e-shop and must be imported into their respective external application.

Authentication

In order to access the Web Services, the caller has to provide valid credentials. The authentication scheme is HTTP Basic Authentication (IETF RFC 2617 HTTP Unauthorized). If the credentials are not valid, the server replies with a (status code 401). In short, a header parameter containing must be supplied. Authorization "Basic base64encode(username:password)"

Example for user "Aladdin" with password "open sesame":

Header name	Value
Authorization	Basic QWxhZGRpbjpvcGVuIHNlc2FtZQ==

Fees

The table below lists the possible fees that can be returned from the *GetOrder PlaceOrder Service Interactions* and . Please note that the local market may decide to not show fees at all or only selected fees. Also, the fee text may be localized if decided so by the local market.

Fee id	Fee text
SMALL_ORDER_TRANSPORT_FEE	Small order transport fee
SMALL_ORDER_FEE	Small order fee
SERVICE_FEE	Service fee
FREIGHT1_FEE	Freight fee
FREIGHT2_FEE	Freight fee
RETURN_FEE	Return fee
CARRY_BEYOND_LOADING_FEE	Carry beyond loading bay fee
INTERNET_DISCOUNT_FEE	Internet discount
CUTTING1_FEE	Cutting fee
CUTTING2_FEE	Cutting fee
CUTTING_TOTAL	cutting fee
PACKAGING1_FEE	Packaging fee
PACKAGING2_FEE	Packaging fee
PACKAGING_TOTAL	packaging fee
EXPRESS_DELIVERY_FEE	Express delivery fee
ENVIRONMENTAL_FEE	Environmental fee
BROKEN_PACKAGE_FEE	Broken package fee
PICKUP_DISCOUNT_FEE	Pickup discount
TIME_WINDOW_FEE	Time window fee
DC_ENVIRONMENTAL_FEE	DC environmental fee
FIXED_SERVICE_FEE	Fixed service fee
TRANSPORT_FEE	Transport fee
MINIMUM_TRANSPORT_FEE	Minimum transport fee
TRANSPORT_TOTAL	transport fee

References

Swagger API Documentation & Design Tools for Teams | Swagger

OpenAPI OpenAPI Specification - Version 3.1.0 | Swagger