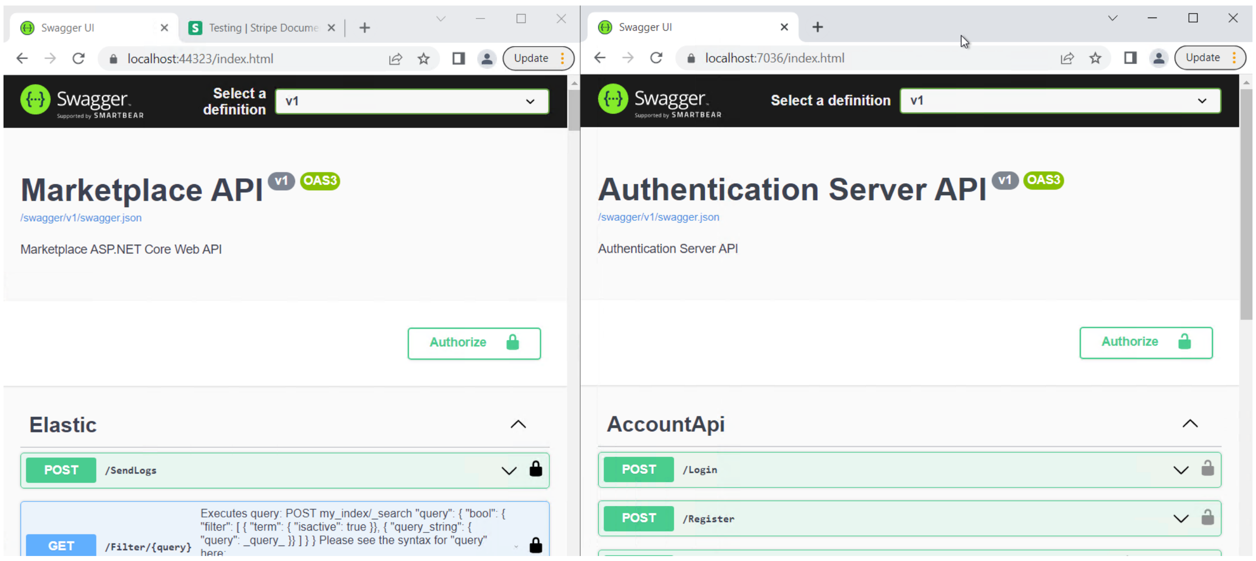
Before you start

First, you should enable https endpoint: https://docs.microsoft.com/en-us/azure/service-fabric/service-fabric-tutorial-dotnet-app-enable-https-endpoint.

To make Identity Server work, we need to add self-signed certificate. From PowerShell, run command:

```
C:\program files\microsoft sdks\service fabric\clustersetup\secure> .\CertSetup.ps1 -Install -CertSubjectName CN=mytestcert
```

There are 2 web applications in the solution: Marketplace.Auth and the Marketplace.API (pic.1). We will show their work on the local machine. You can also test the API with a postman, but for convenience, we connected the swagger to the API of both web applications. This document will show how authorization works.

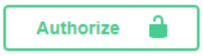


Executes query: POST my_index/_search "query": { "bool": { "filter": [{ "term": { "isactive": true }}, { "query_string": { "query": _query_ }}] } } Please see the syntax for "query" here:

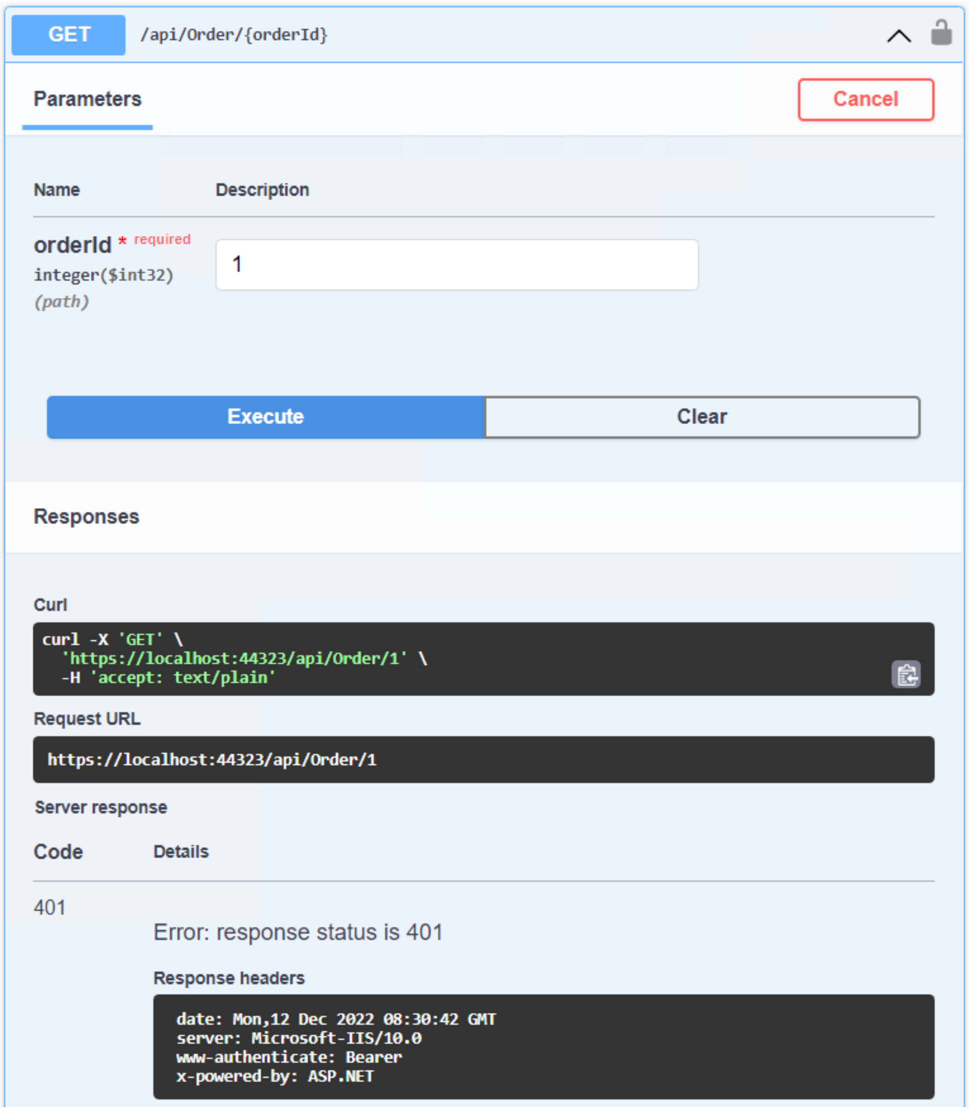
Pic.1 Web applications

Authorization

If no users registered, a new one should be created using the POST /Register API.

If we try to access data from API -> Orders -> GET /api/Order/{orderId} being not authorized, we will receive a response with a 401-status code (pic.2). To proceed, we should click 

button. In opened dialog, click on the Authorize button again. You will be redirected to the identity server web application. You can see that the port in the URL has changed. The login page will open (pic.3).



```
curl -X 'GET' \
  'https://localhost:44323/api/Order/1' \
  -H 'accept: text/plain'
```

Request URL

```
https://localhost:44323/api/Order/1
```

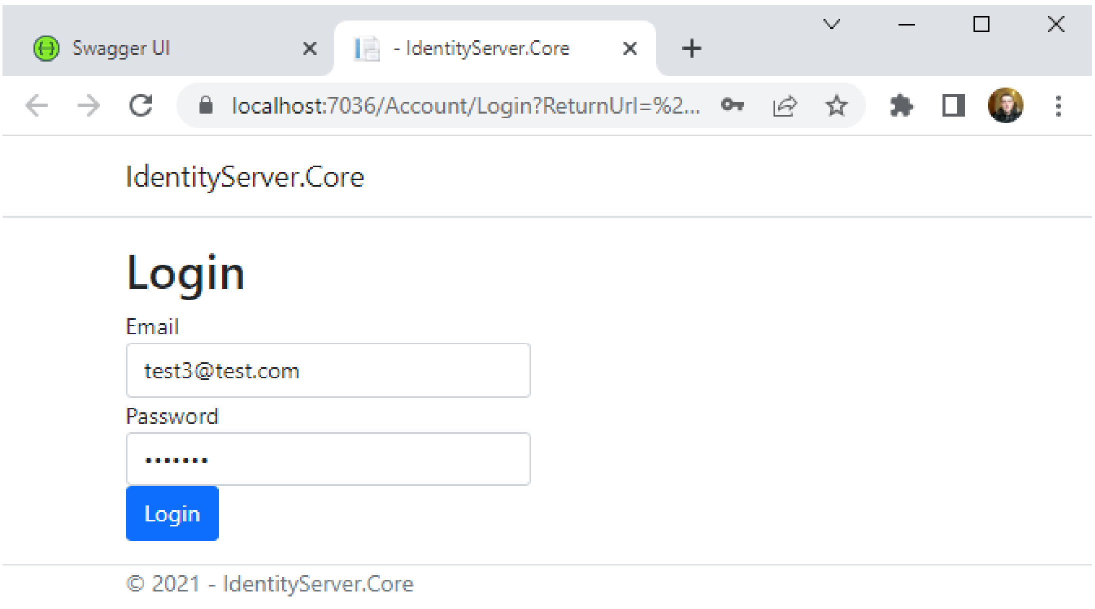
Server response

Code	Details
401	Error: response status is 401

Response headers

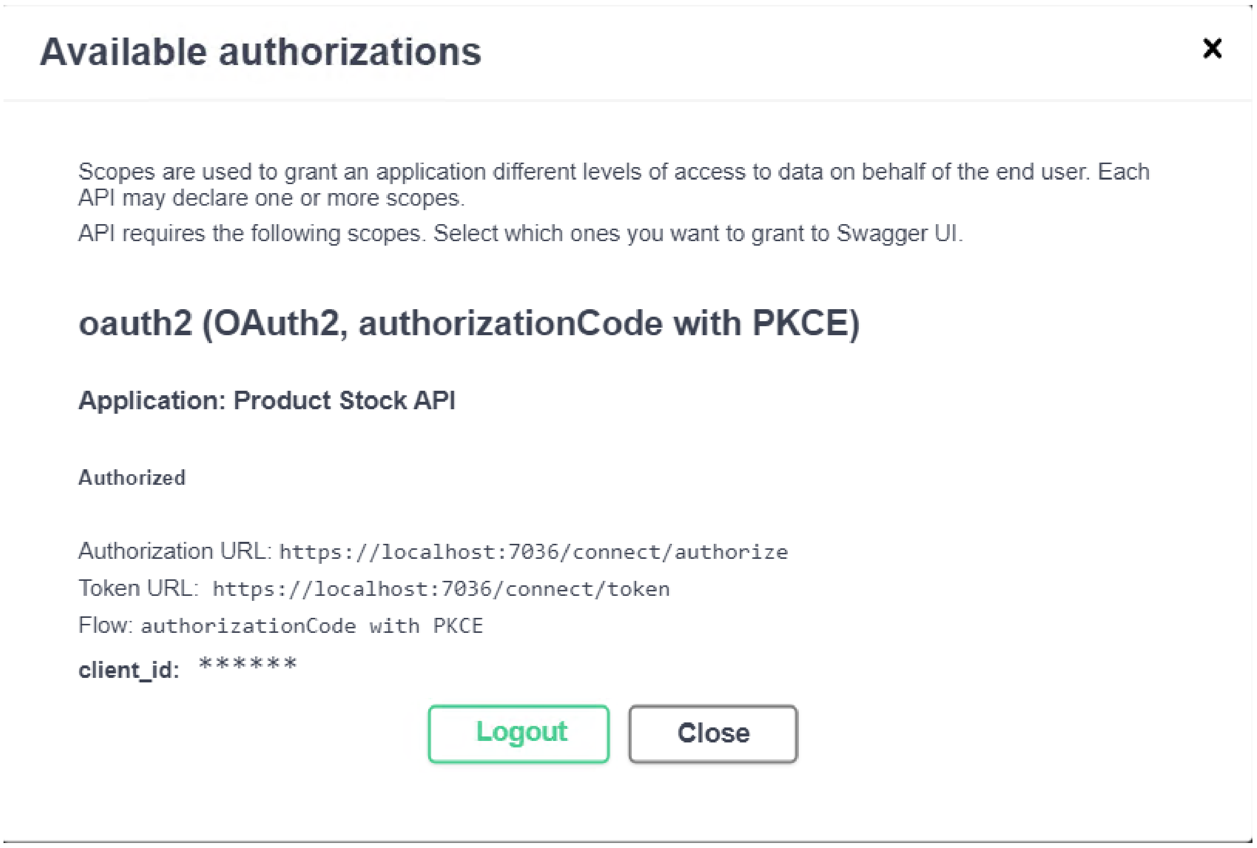
```
date: Mon,12 Dec 2022 08:30:42 GMT
server: Microsoft-IIS/10.0
www-authenticate: Bearer
x-powered-by: ASP.NET
```

Pic. 2 401-status code



Pic.3 Login page

After entering the login and password and clicking Login, we will be authorized and redirected back to the swagger page (pic.4):


Available authorizations

Scopes are used to grant an application different levels of access to data on behalf of the end user. Each API may declare one or more scopes.
API requires the following scopes. Select which ones you want to grant to Swagger UI.

oauth2 (OAuth2, authorizationCode with PKCE)

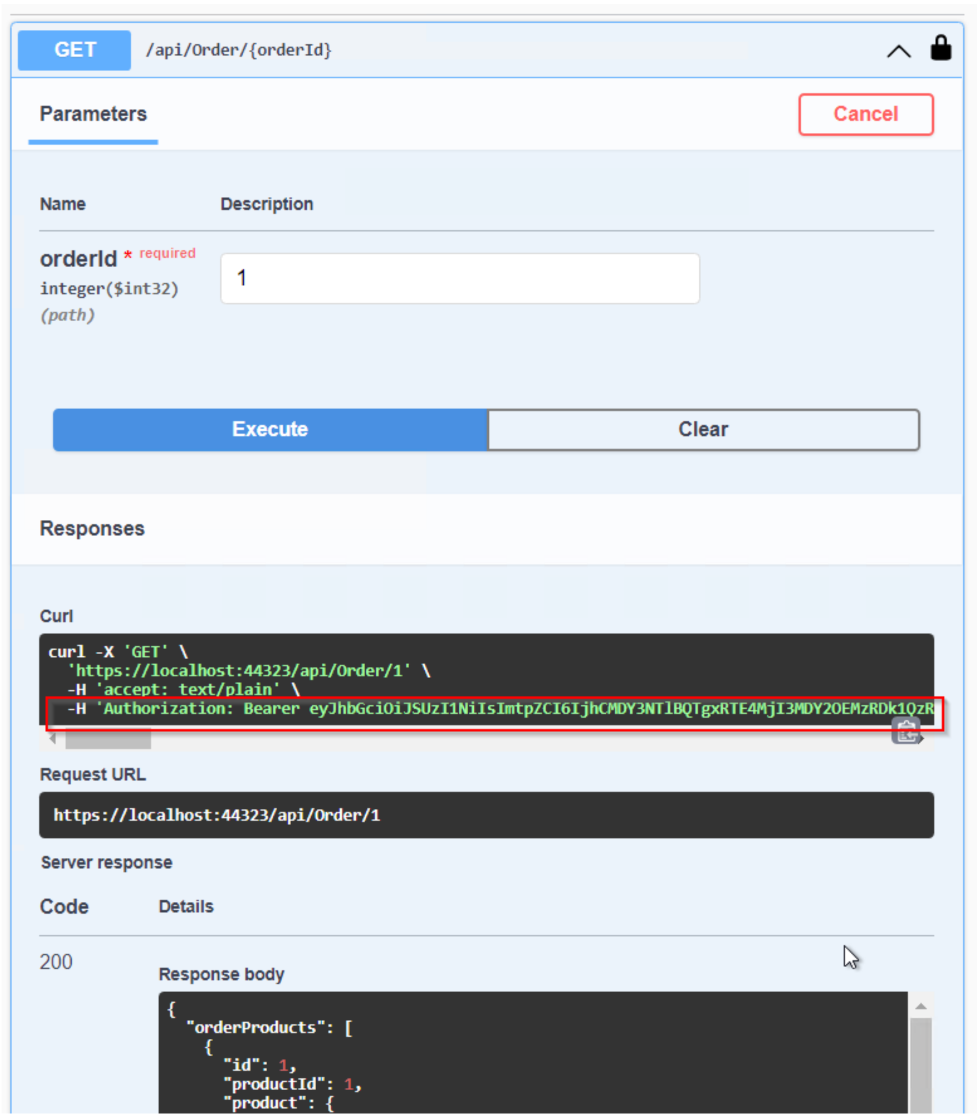
Application: Product Stock API

Authorized

Authorization URL: `https://localhost:7036/connect/authorize`
Token URL: `https://localhost:7036/connect/token`
Flow: `authorizationCode with PKCE`
client_id: ******

Pic. 4 Successful authorization

You can safely close this popup, since we are authorized and the swagger is already responsible for storing and using the access token. If you use Postman, then the flow will be slightly different, but even there the goal will be to get an access token. Now let's try to access the protected resource again. As we can see, the swagger took care of substituting the access token in the header and allowed us to access the API successfully (pic.5).



```
curl -X 'GET' \
  'https://localhost:44323/api/Order/1' \
  -H 'accept: text/plain' \
  -H 'Authorization: Bearer eyJhbGciOiJSUzI1NiIsImtpZCI6IjhCMDY3NTlBQTgxRTE4MjI3MDY2OEMzRDk1QzR
```

```
https://localhost:44323/api/Order/1
```

```
{
  "orderProducts": [
    {
      "id": 1,
      "productId": 1,
      "product": {
```

Pic.5 Authorized action

Marketplace.Auth methods description

AccountAPI has such methods (pic.6)

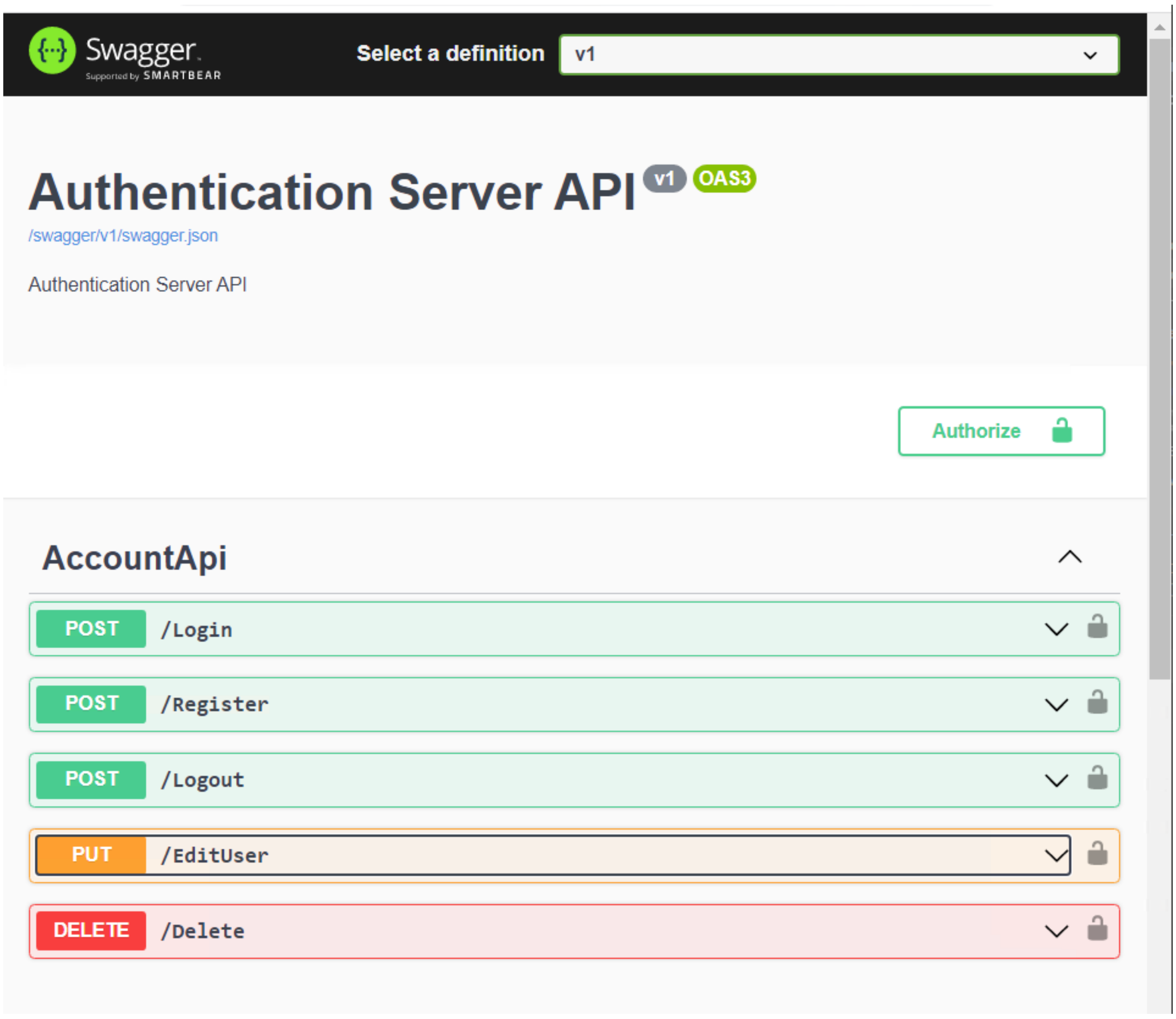
/Login - user authorization. Just an additional authorization method. This is not the same with the authorization page where we entered the username and password.

/Register - registration of a new user. On an empty database, you will need to add a new user.

/Logout - logout the current user. Protected resource, requires the user to be authorized.

/EditUser - editing the current user. A protected resource that requires the user to be authorized.

/Delete - delete the current user. A protected resource that requires the user to be authorized.



Authentication Server API v1 OAS3

/swagger/v1/swagger.json

Authentication Server API

AccountApi

POST /Login

POST /Register

POST /Logout

PUT /EditUser

DELETE /Delete

Pic.6 AccountApi methods