API Authentication

For most API calls you will need to authenticate yourself. This is done by passing an OAuth `access_token` with every request. You can obtain an ac
`cess_token` using the OAuth2 login **or** you can ask the user to provide you with an *authorization code* and exchange that for an `access_token`.

> ⚠ Note that you are **not allowed** to ask a user for their username/password! This password should only be entered on the portal itself or on a website from the school. Please contact us if this is a problem for you,

Create an access_token in the portal

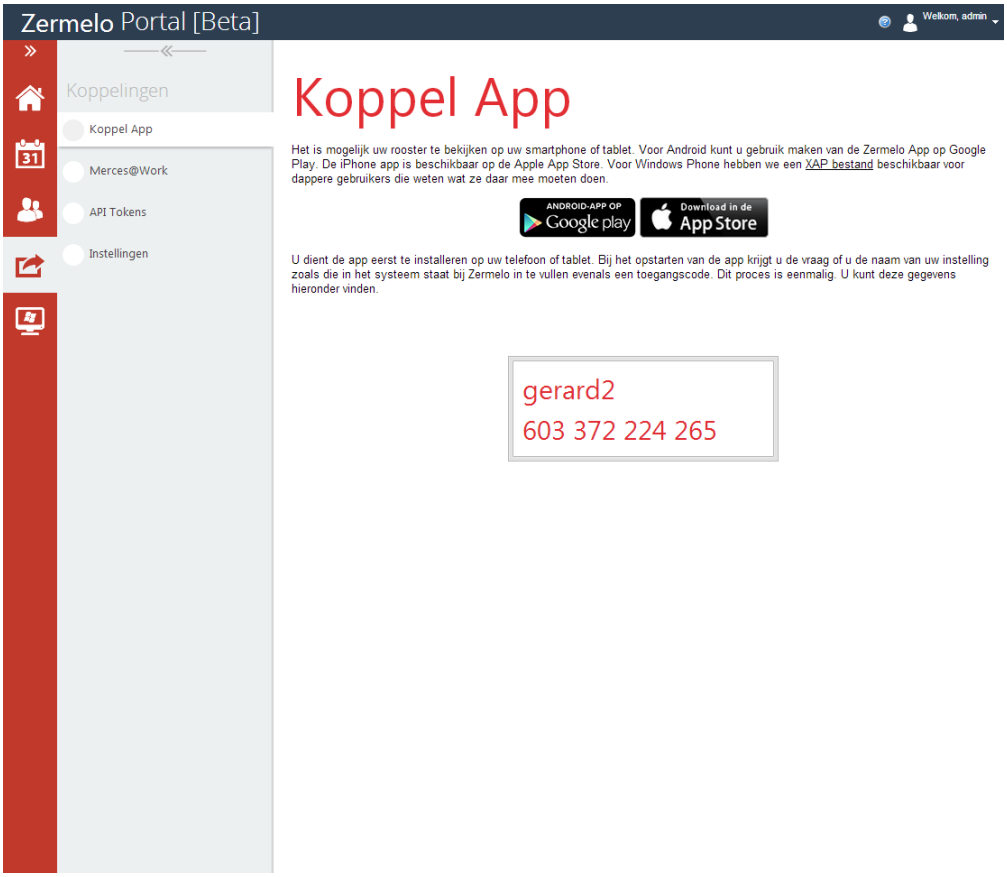
> ⓘ **Only as admin**
>
> Directly creating tokens is only possible if you're an *admin* user.

In the portal, navigate to **Beheer > Admin-paneel > API Tokens**. You can create a token using **Toevoegen**.

Obtain an authorization code from the user

If you want to link another application to the API but it is on another device (for example a phone), it may not be convenient to have the user type the entire access token. In that case you can use the auth code to retrieve the access token. Note that this only works once.

The user can generate a new authorization code with just the permissions to view the schedule using the "Koppel App" (Link App) screen:



Here "gerard2" is the address of the portal (full address would be gerard2.zportal.nl) and 603372224265 is the authorization code. The access token you can obtain using this code is valid for multiple years and has sufficient permissions to view the schedule.

Exchange the authorization code for an access token

When you have obtained an authorization code either from the user or from the OAuth login you can exchange this for an access token.

You will have to HTTP `POST` the authorization code to the API to retrieve the access token. For example: this is what you need to do using curl:

```
$ curl --data "grant_type=authorization_code&code=123456789012" https://schoolnaam.zportal.nl/api/v2/oauth
/token
```

You should replace *schoolnaam* and *1234567890123* with the values provided by the user or by the OAuth login step. The response will be 200 if everything went OK and 400 if something went wrong. This is a successful response:

```
{
    "access_token":"gmj5if8aqcute3n0evi6g3uaee",
    "token_type":"bearer",
    "expires_in":57600
}
```

Example of obtaining a token from a code using HTML

Paste this in a HTML file to exchange the code for a token using a form. This may be a good way to get a feel for it. See if this works in your browser!

```
<html>
 <body>
  <p>Please enter the code without any spaces.</p>
  <form action="https://schoolnaam.zportal.nl/api/v2/oauth/token" method="post">
   <input type="hidden" name="grant_type" value="authorization_code"/>
   <input type="text" name="code"/>
   <input type="submit" value="POST"/>
  </form>
 </body>
</html>
```

Using the access token

After you've received the token you should store it in your application and use it for future requests to the API. When performing a request there are two methods to provide the *access token*. The first is to pass it as a parameter in the request. Example: `GET /api/v2/students?access_token=ACCESS_TOKEN`. It is also possible to pass the token in an HTTP header: `Authorization: Bearer ACCESS_TOKEN`. Finally, it's possible to use HTTP Basic authentication with the username `Bearer` and the access token as password.

The access token has limited validity. It expires when the time *expires_in* has elapsed. An access token returned with `expires_in=3600` will expire in one hour.

If you want to invalidate a token, use it to perform an empty post to `/oauth/logout`.