

Research document

Thomas van der Molen

IPS3-DB03

Project Information	
Project members	Thomas van der Molen (4168003)
Project name	Text Adventure
Version	1.1

Table of Contents

Version History.....	2
Introduction	3
ASP.NET Web API	3
RESTful API	3
JWT tokens	3
Google OAuth.....	4
Resources	5

Version History

Version	Date	Change
1.0	01-09-2021	Created file, and added ASP.NET Web API, JWT Tokens, Google OAuth and Resources.
1.1	24-09-2021	Added resources for signal R and JWT tokens

Introduction

In this document I will go over all functionalities / systems that I have done extensive research on, this can include but is not limited to; a testing project, reading a research paper or exploring different options within a technology.

ASP.NET Web API

For my project I have chosen to use a C# REST API backend, this is because I have previous experience in C# and the same technologies will possibly be used in my group project (this is also based on an ASP.NET backend).

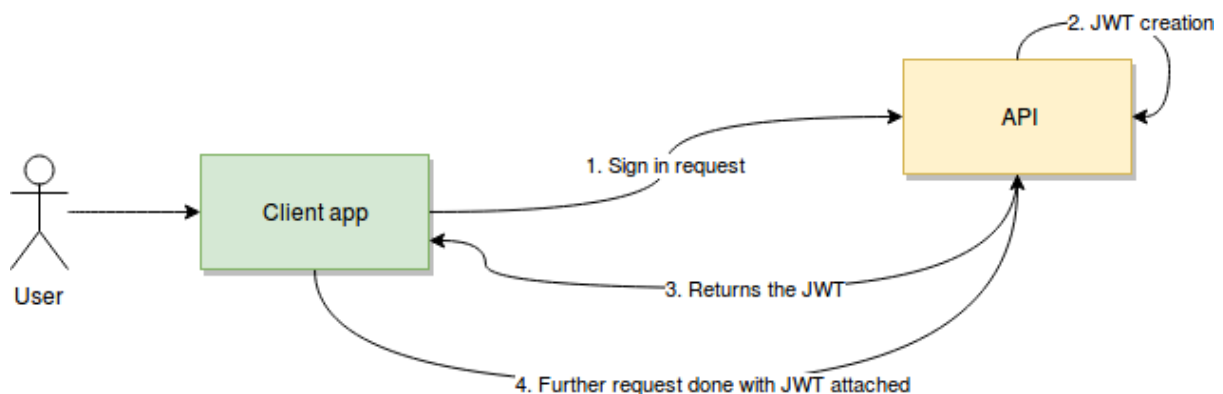
RESTful API

I have chosen to use a REST (RESTful) API for my backend communication. REST stands for representational state transfer, this means the connection will be based on clients, servers and resources. One of the main criteria of a RESTful API is that no user data will be stored in between requests (a stateless client-server communication), this means that every API request should be considered as an individual request and handle as so.

Another API technique that could be considered is SOAP (Simple Object Access Protocol), this means that SOAP uses service interfaces to expose its functionality to client applications. SOAP can also only use XML formats, because of this the bandwidth needed for SOAP will be larger than REST.

JWT tokens

JWT tokens are used to identify and authenticate an API request, it can be seen as a security token that has to be shown to the API to send/receive data.



JWT tokens stand for JSON Web Tokens, the token itself contains of three parts separated by “.”. The first part of the JWT token is the header, the header can contain many things but will normally contain the type and the encoding algorithm. A Visualization of the header can be seen below.

```
{
  "alg": "HS256",
  "typ": "JWT"
}
```

The next part of the token is the payload, the payload will contain the data that you want to send to the API, this can contain things like the username or the expiration date of the token. A Visualization of the payload can be seen below.

```
{  
  "Iss": "Thomas"  
  "Exp": 1550946689,  
  "Admin": false  
}
```

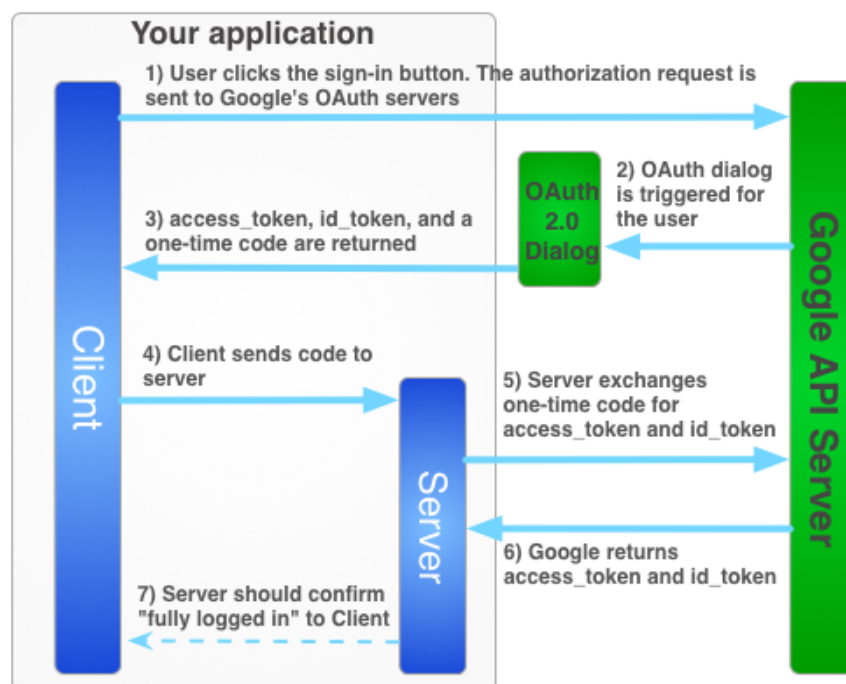
The last part of the token is the signature, the signature will combine the header and payload, it will also contain a secret token. This signature will verify that the data has not been edited. If the data would be edited the signature would not match up with the payload and header anymore.

JWT tokens should also contain an expiration date, after this date the token will not be accepted anymore, and the user will have to verify his identity to get a new one. In practice when a token gets sent in with a valid expiration date and the token gets accepted, the API will return a token with a refreshed expiration date.

Google OAuth

For my website it is important that users log in so I can save their data effectively, for this reason I think it should be important that it is as easy as possible to register/login. To help this I have looked registering on a website via google authentication.

This means that instead of registering an account on the web application itself, you will use your google account to login. After logging in, google will send a special ID to the client this ID can be sent to the backend. The backend will exchange this one-time special ID with google in exchange for user information such as email or username.



Resources

Here is a list of resources that can / have been used for and during the research of the items covered in this document.

ASP.NET Web API

[Testing Web API backend project](#)

<https://www.redhat.com/en/topics/api/what-is-a-rest-api>

<https://www.guru99.com/comparison-between-web-services.html>

[Project from tutorial](#)

<https://www.youtube.com/watch?v=nEQvA5HfEDE>

https://docs.microsoft.com/en-us/aspnet/core/signalr/introduction?WT.mc_id=dotnet-35129-website&view=aspnetcore-5.0

<https://stackoverflow.com/questions/22917723/signalr-security>

<https://docs.microsoft.com/en-us/aspnet/core/signalr/security?view=aspnetcore-5.0>

JWT tokens

[Testing backend JWT token generation](#)

[Testing frontend JWT handling token](#)

[Project from tutorial](#)

<https://jwt.io/>

<https://blog.logrocket.com/jwt-authentication-best-practices/>

<https://blog.logrocket.com/how-to-secure-a-rest-api-using-jwt-7efd83e71432/>

Google OAuth

<https://www.scienceabc.com/innovation/oauth-how-does-login-with-facebook-google-work.html>

<https://medium.com/mickeysden/react-and-google-oauth-with-net-core-backend-4faaba25ead0>

[Project from tutorial](#)

[Testing OAuth backend](#)

[Testing OAuth frontend](#)

<https://stackoverflow.com/questions/58251781/transferring-react-client-side-google-sign-in-to-asp-net-server-side>