Authentication
Getting started

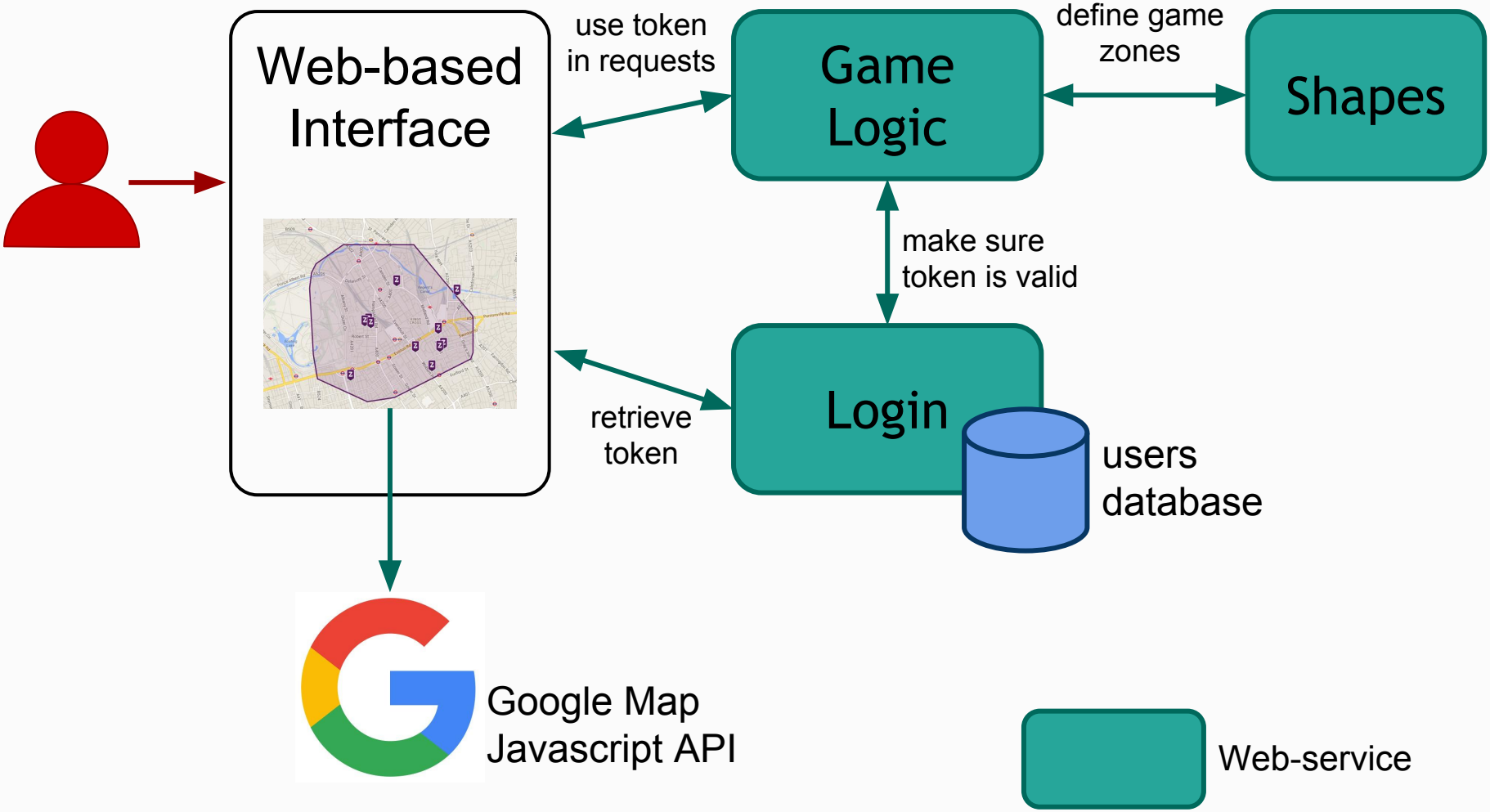
18/10/2016

Cécile Camillieri

iut Nice Côte d'Azur

Université Nice Sophia Antipolis

Reminder: Proposed Architecture



Token-based Authentication

1- User provides their username and password

2- User Interface queries **authentication Web-Service**

3- **Web-Service** checks the credentials and generates a unique session token

4- Every time a request that requires authentication is made to **your services**, the token is passed as an url parameter

5- **The given web service** must check with the **authentication service** that the token is valid.

6- If so, the request can continue as normal.

Authentication web service

First step

- Provided for the first sprint:
  - Log user and generate token
  - Checks that a token is valid

- Each of you has an account.
  - login and password: your student id

- In the future:
  - we give you the source code
  - user account creation
  - password update
  - ...

Authenticating

Operations: login

- http://iut-outils-gl.i3s.unice.fr/jetty/**authentication**/

- Method: **POST**
- Content-Type: **application/x-www-form-urlencoded**

- Parameters (in request body):
  - username
  - password, hashed with MD5 algorithm

- Response:
  - success: code 200 (ok) + token
  - failure: code 401 (unauthorized)

Operations: login

- Calling a service in JS, some possibilities:
  - Plain JS with XMLHttpRequest

http://www.w3schools.com/xml/dom_http.asp. Example in the file 'script.js' provided with the shapes web service

  - AngularJS

https://docs.angularjs.org/api/ng/service/$http. Example in the provided Angular project.

- Security: hashing passwords with md5
  - Don't send passwords over the network like this
  - Hash = unilateral transformation
  - Careful! md5 is *not* secure enough for real-life applications...

- MD5 in JS, some possibilities:
  - Plain JS:

```
<script src="http://crypto-js.googlecode.com/svn/tags/3.0.2/build/rollups/md5.js"></script>
<script> var passhash = CryptoJS.MD5(password);</script>
```

  - AngularJS (install with bower):

https://www.npmjs.com/package/angular-md5

Verifying the token

Operations: token validation

- http://iut-outils-gl.i3s.unice.fr/jetty/**authentication**/**session**

- Method: **POST**
- Content-Type: **application/x-www-form-urlencoded**

- Parameters (in request body):
  - token

- Response:
  - success: code 204 (no content)
  - failure: code 401 (unauthorized)

Operations: token validation

- HTTP requests in Java:
  http://download.java.net/jdk7/archive/b123/docs/api/java/net/HttpURLConnection.html

  - Example with POST:

```
String url = "http://example.com";
String charset = "UTF-8"; // Or in Java 7+: java.nio.charset.StandardCharsets.UTF_8.name()
String param1 = "value1";
String param2 = "value2";
String query = String.format("param1=%s&param2=%s",
      URLEncoder.encode(param1, charset), URLEncoder.encode(param2, charset));

HTTPURLConnection connection = (HTTPURLConnection) new URL(url).openConnection();
connection.setDoOutput(true); // Triggers POST.
connection.setRequestProperty("Accept-Charset", charset);
connection.setRequestProperty("Content-Type", "application/x-www-form-urlencoded;charset=" + charset);
try (OutputStream output = connection.getOutputStream()) {
      output.write(query.getBytes(charset));
}

InputStream response = connection.getInputStream();
[...]
int status = connection.getResponseCode();
connection.disconnect();
```

Maven modules are back

On course website

- Zip file containing the skeleton of a maven project for your game
- Pom.xml (parent)
  - shapes-ws/pom.xml (shapes ws module)
  - game-ws/pom.xml (game logic ws module)
  - authentication (module for checking token validity)
- Your modules can have a dependency to the authentication module
- All modules can be compiled in one command: mvn clean package
- All web-services can be deployed in one command: mvn jetty:run

For October 23th 23:59

Everyone in the team joins Github Classroom.

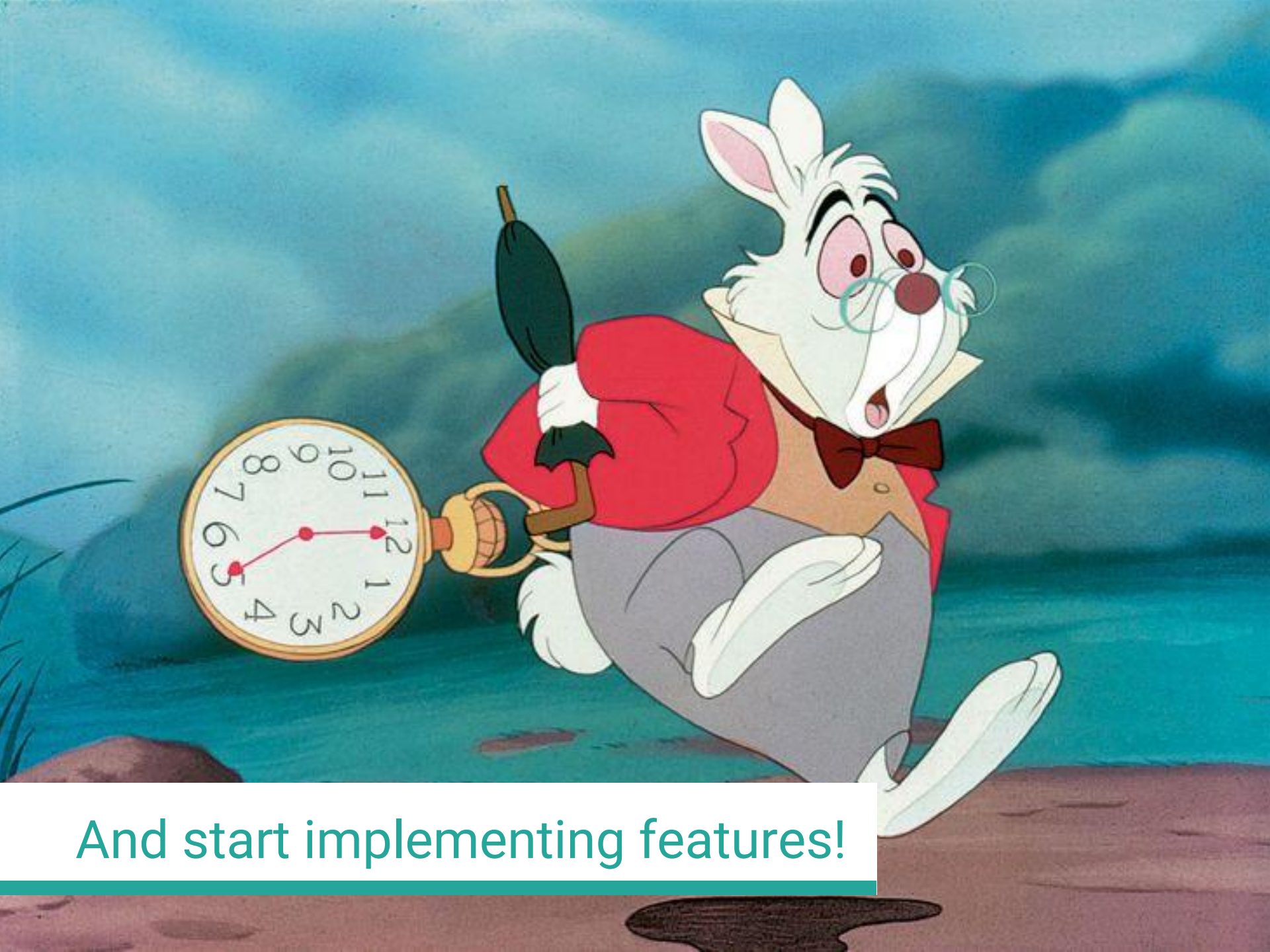
Install maven: http://www.mkyong.com/maven/how-to-install-maven-in-windows/

Setup your maven modules based on the zip in the website.

Don’t forget to update the pom for the shape ws as well.

Readme: commands needed to build and run the project.

Add proper Jenkinsfile in your repository.



And start implementing features!