

Geo Catching Sprint #3 Kick-off

03/01/2017

Cécile Camillieri/Clément Duffau



GeoCatching sprint #1

Drawing of zones on a map

User login and joining of a game

Browser-based geolocation of the player

Color differently the zone the player is in

GeoCatching sprint #2

[M1] Advanced administration of players and avatars.
Players ranking.

[M2] Team administration. Player can join a team and catch zones for this one.

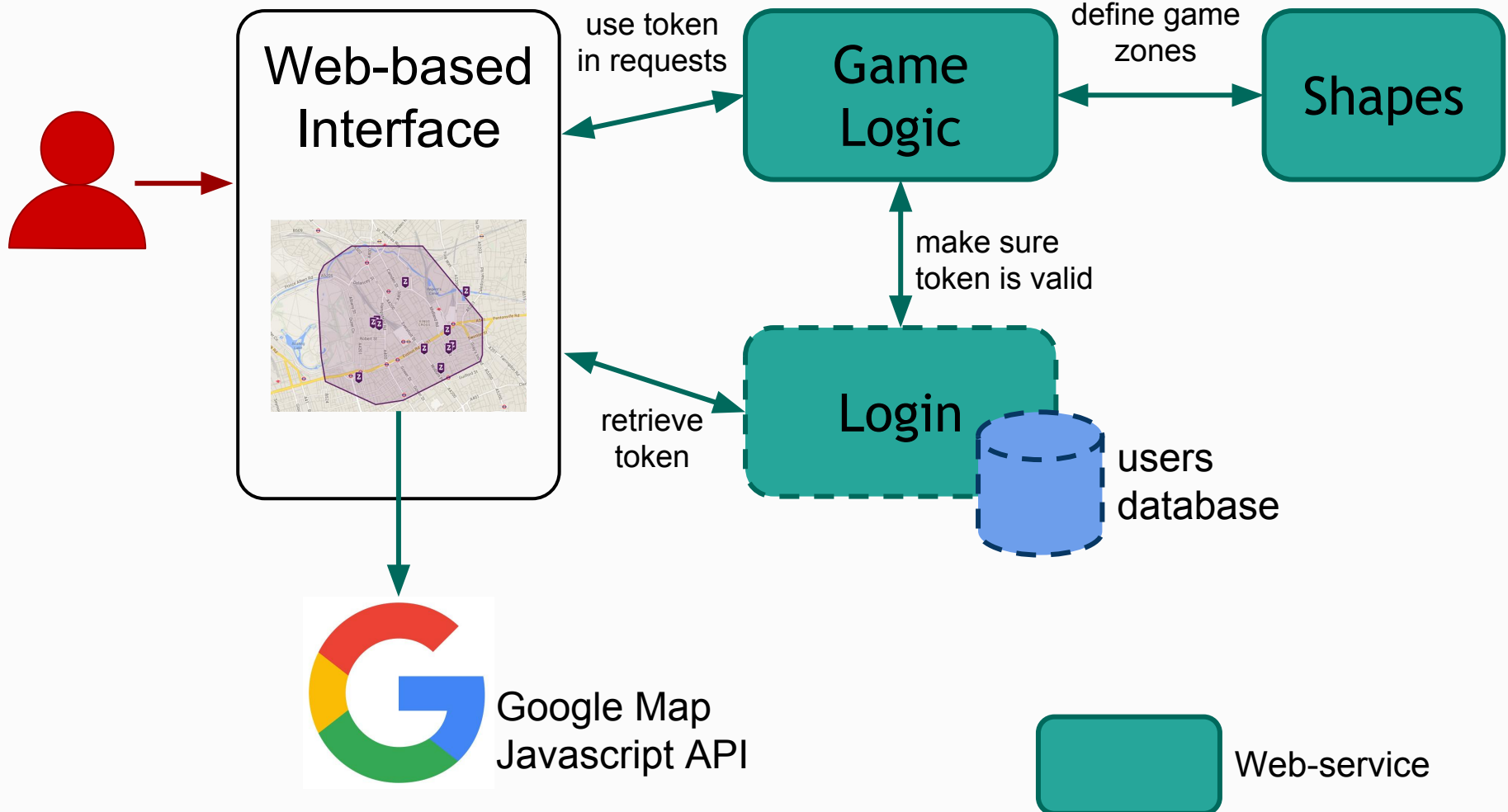
[M3] Define zones from shapes with the Shape WS

[M4] Handle catching a free zone via mini-game

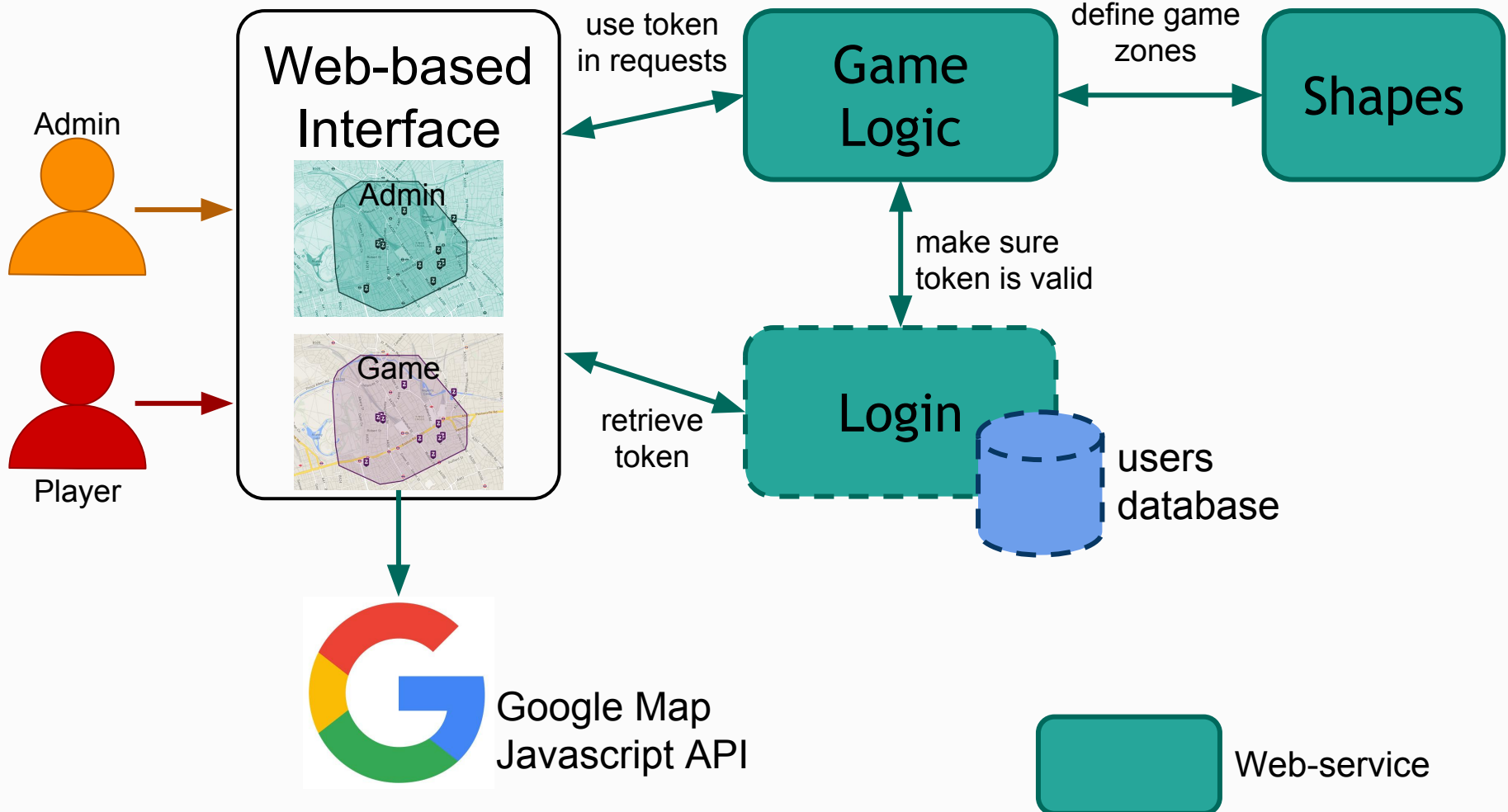
[M5] Handle zone reinforcement (zone already caught)

[M6] Handle catching a non-free zone via mini-game

Current Architecture



Current Architecture (for some)



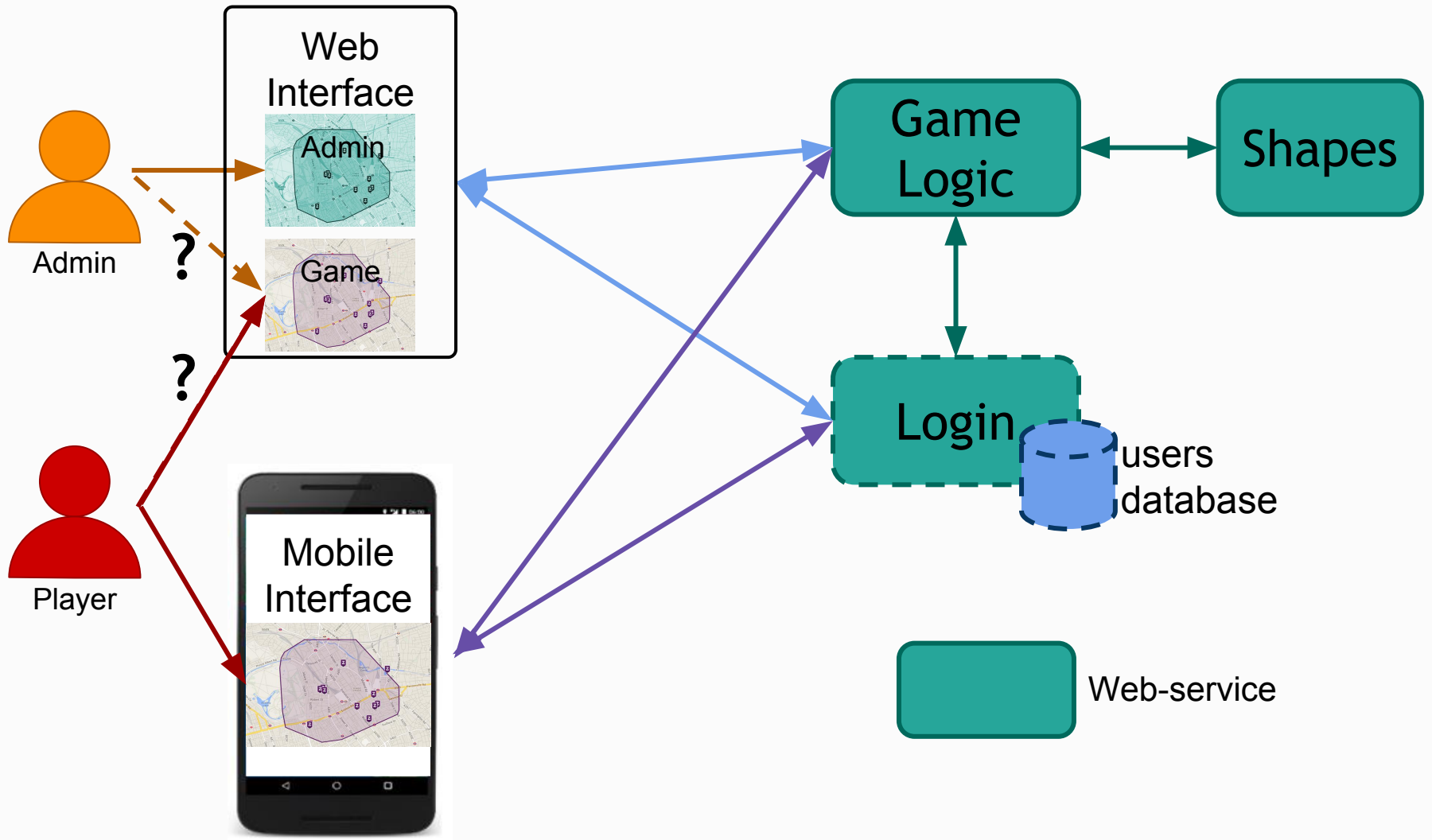
GeoCatching sprint #3 to 5

Integration of all sprints #1 and #2 features for your team

Development of a **mobile** interface to play

Creation of a new, **unique feature** per group

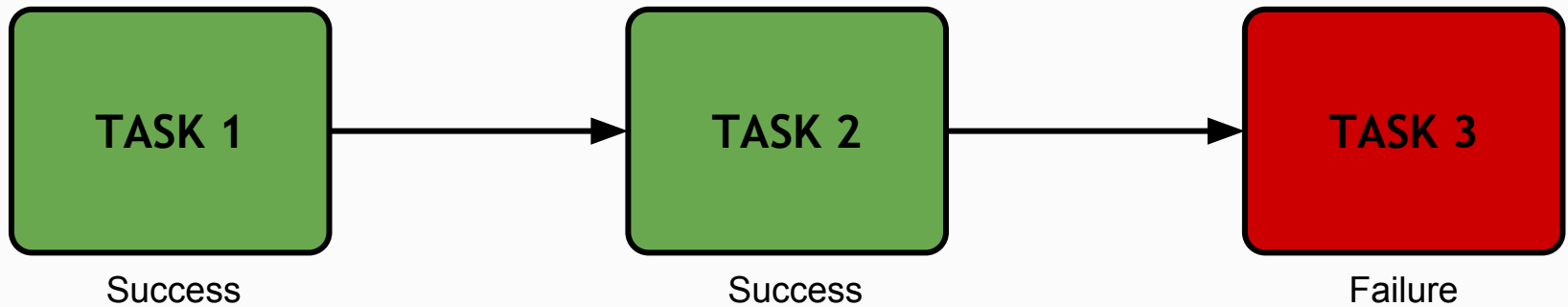
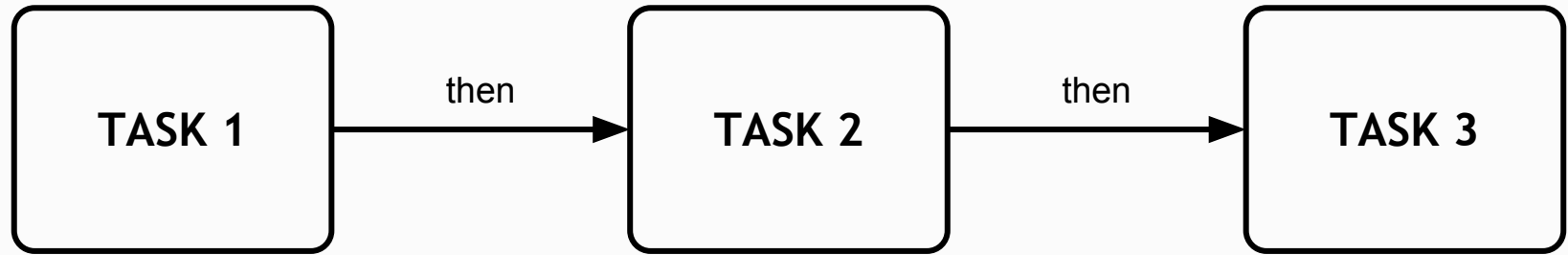
Current Architecture



How to...

Do more with Jenkins

Jenkins Pipelines



Pipelines with Jenkinsfile

```
node ('master') {  
    stage ('back-maven') {  
        checkout scm  
        sh 'mvn -f mavenfolder/pom.xml clean package'  
    }  
    stage ('front-angular') {  
        checkout scm  
        sh 'cd folder; npm install; bower install; grunt'  
    }  
    stage ('front-android') {  
        checkout scm  
        sh 'cd androidfolder; ./gradlew assemble'  
    }  
}
```

Jenkins & Android

- Need to update .gitignore file `.jar` → `^(*gradle-wrapper)*.jar`
- Generate a `gradle wrapper` (or use the one from Android Studio) and `add it to git`
 - Gradle wrapper = All the directory `gradle/` + `gradlew` at android project root
- `git update-index --chmod=+x gradlew`
- Commit and push to your repo

How to...

Deploy your web services

Back to: Plugins for custom goals

mvn jetty:run

run a jetty (server) instance based on the configuration

```
<plugins><plugin>....  
  <configuration>  
    <scanIntervalSeconds>10</scanIntervalSeconds>  
    <webApp>  
      <contextPath>/path</contextPath>  
      <descriptor> [...] web.xml</descriptor>  
    </webApp>  
  </configuration>  
</plugin></plugins>
```



Contains configuration for the web-service

Back to: Plugins for custom goals

mvn jetty:run

run a jetty (server) instance based on the configuration

=> Deploy the war on a jetty instance

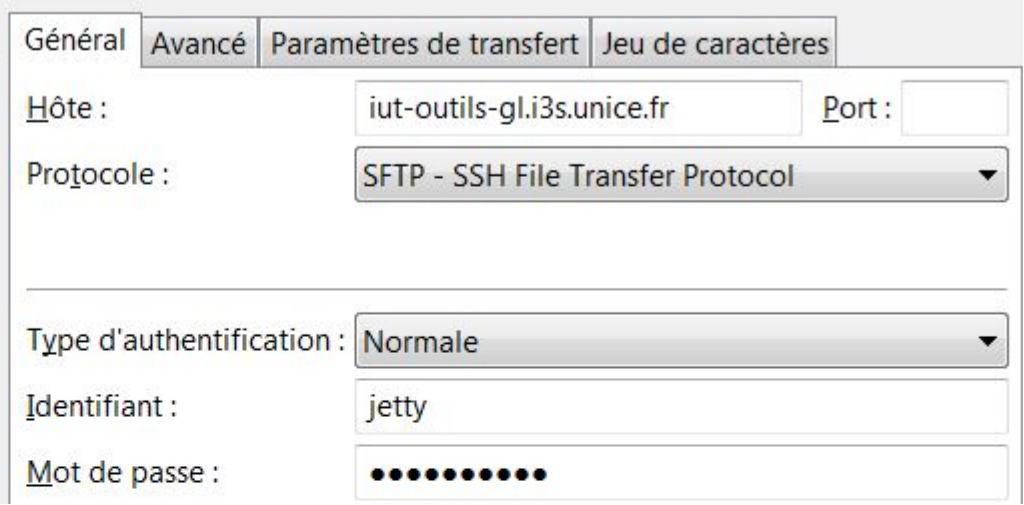
=> Available at localhost:8080/path

Deploying the war on a server

- Jetty server available at <http://iut-outils-gl.i3s.unice.fr/jetty/>
Username: 'jetty' Password: 'deploy2016'

- Deployment:

- **Command line:** `scp /path/to/war jetty@iut-outils-gl.i3s.unice.fr:/opt/jetty/webapp/my.war`
- Using a tool like [FileZilla](#):



The image shows a screenshot of the FileZilla configuration dialog box, specifically the 'Général' (General) tab. The dialog is used to configure an SFTP connection. The fields are as follows:

- Hôte :** iut-outils-gl.i3s.unice.fr
- Port :** (empty)
- Protocole :** SFTP - SSH File Transfer Protocol
- Type d'authentification :** Normale
- Identifiant :** jetty
- Mot de passe :** (masked with dots)

Deploying the wars on a server

- Your wars must be directly in the folder `/opt/jetty/webapp/`
 - No sub-folder
- Only upload a single war for the same service
 - Always use the same names for your wars
- The name of your wars must follow this pattern
 - Start with 'idse-x' where x is the id of your team (1 to 4)
 - A file 'idse-6.war' will result in your web-service to be deployed at the url:

[http://iut-outils-gl.i3s.unice.fr/jetty/
idse-6/{whats/defined/in/your/war}](http://iut-outils-gl.i3s.unice.fr/jetty/idse-6/{whats/defined/in/your/war})

How to...

Use Maps API

for Android

Setting up with an existing project

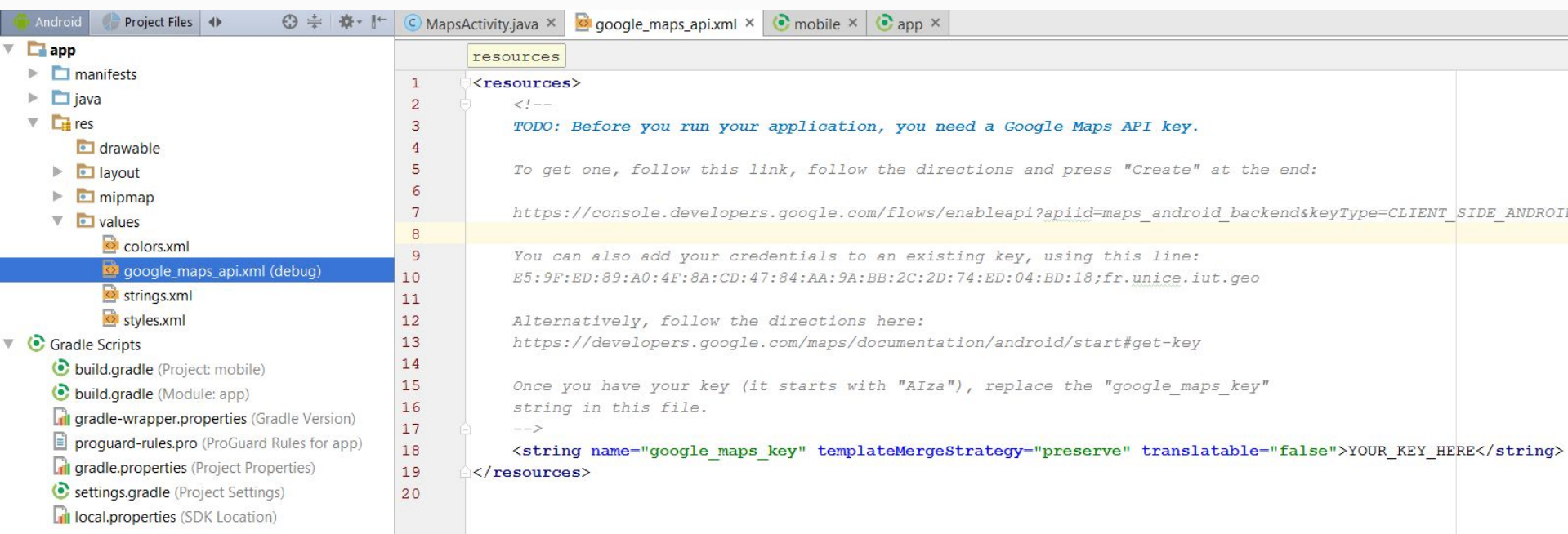
- Create a new Google Maps Activity

The image shows a screenshot of an IDE's 'New' menu. The 'Activity' option is highlighted. To the right, a list of activity templates is visible, with 'Gallery...' selected. Below the list, the following dependencies are shown:

```
implementation 'com.google.android.gms:play-services:10.0.1'
```

Setting up with an existing project

- A file 'google_maps_api.xml' will be added in your resources

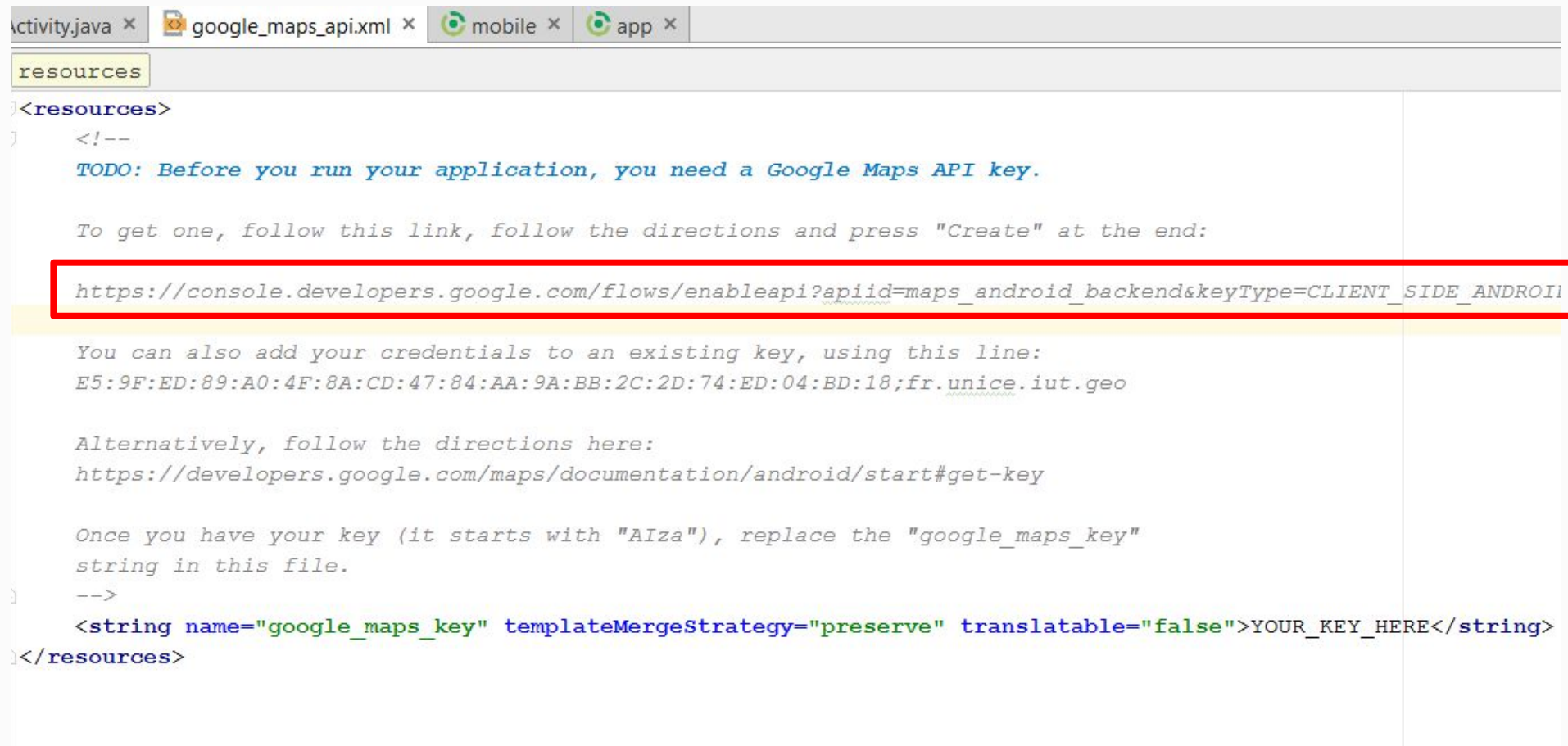


The screenshot shows an IDE interface with a project tree on the left and a code editor on the right. The project tree shows the 'app' directory with sub-directories 'manifests', 'java', and 'res'. The 'res' directory is expanded to show 'values', which contains 'colors.xml', 'google_maps_api.xml (debug)', 'strings.xml', and 'styles.xml'. The code editor shows the content of 'google_maps_api.xml' with the following text:

```
1 <resources>
2 <!--
3  TODO: Before you run your application, you need a Google Maps API key.
4
5  To get one, follow this link, follow the directions and press "Create" at the end:
6
7  https://console.developers.google.com/flows/enableapi?apiid=maps\_android\_backend&keyType=CLIENT\_SIDE\_ANDROID
8
9  You can also add your credentials to an existing key, using this line:
10 E5:9F:ED:89:A0:4F:8A:CD:47:84:AA:9A:BB:2C:2D:74:ED:04:BD:18;fr.unice.iut.geo
11
12 Alternatively, follow the directions here:
13 https://developers.google.com/maps/documentation/android/start#get-key
14
15 Once you have your key (it starts with "AIza"), replace the "google_maps_key"
16 string in this file.
17 -->
18 <string name="google_maps_key" templateMergeStrategy="preserve" translatable="false">YOUR_KEY_HERE</string>
19 </resources>
20
```

Setting up with an existing project

- Follow this link to get a key (it is different from the JS key)



```
activity.java x google_maps_api.xml x mobile x app x
resources
<resources>
  <!--
  TODO: Before you run your application, you need a Google Maps API key.

  To get one, follow this link, follow the directions and press "Create" at the end:

  https://console.developers.google.com/flows/enableapi?apiid=maps_android_backend&keyType=CLIENT_SIDE_ANDROID

  You can also add your credentials to an existing key, using this line:
  E5:9F:ED:89:A0:4F:8A:CD:47:84:AA:9A:BB:2C:2D:74:ED:04:BD:18;fr.unice.iut.geo

  Alternatively, follow the directions here:
  https://developers.google.com/maps/documentation/android/start#get-key

  Once you have your key (it starts with "AIza"), replace the "google_maps_key"
  string in this file.
  -->
  <string name="google_maps_key" templateMergeStrategy="preserve" translatable="false">YOUR_KEY_HERE</string>
</resources>
```

Setting up with an existing project

- Follow the instructions

☰ Google APIs



Enregistrer l'application pour Google Maps Android API dans la Console d'API Google

Console d'API Google vous permet de gérer votre application et de surveiller l'utilisation de l'API.

Vous n'avez aucun projet existant. Un projet nommé "My Project" va être créé.

Veillez m'envoyer par e-mail des informations concernant les nouvelles fonctionnalités annoncées, des suggestions pour améliorer les performances, des enquêtes de satisfaction et des offres spéciales.

Oui Non

Je reconnais que l'utilisation de tous les [services et API associées](#) est soumise aux [Conditions d'utilisation](#).

Oui Non

Accepter et continuer

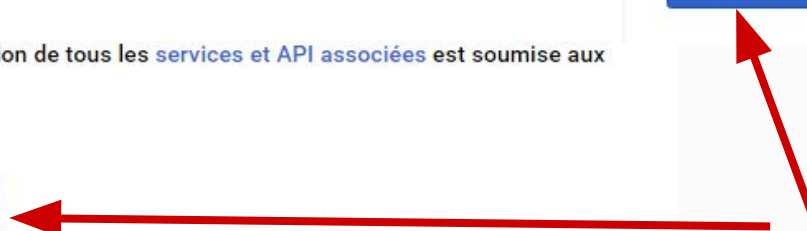
☰ Google APIs



Le projet a été créé et "Google Maps Android API" a été activé.

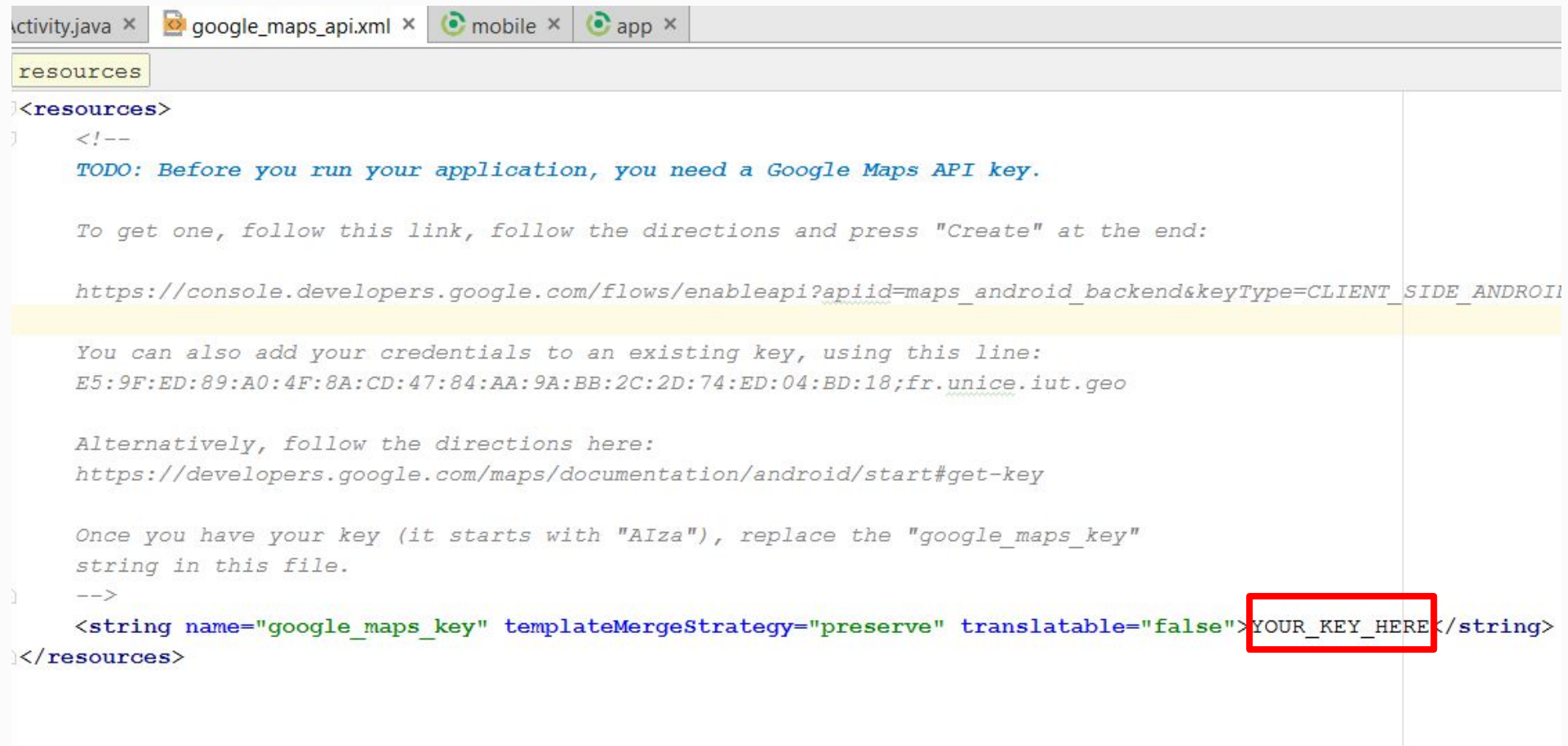
Ensuite, vous devez créer une clé API afin d'appeler l'API.

Créer la clé d'API



Setting up with an existing project

- Put your key in the file



```
activity.java x google_maps_api.xml x mobile x app x
resources
<resources>
  <!--
  TODO: Before you run your application, you need a Google Maps API key.

  To get one, follow this link, follow the directions and press "Create" at the end:

  https://console.developers.google.com/flows/enableapi?apiid=maps_android_backend&keyType=CLIENT_SIDE_ANDROID

  You can also add your credentials to an existing key, using this line:
  E5:9F:ED:89:A0:4F:8A:CD:47:84:AA:9A:BB:2C:2D:74:ED:04:BD:18;fr.unice.iut.geo

  Alternatively, follow the directions here:
  https://developers.google.com/maps/documentation/android/start#get-key

  Once you have your key (it starts with "AIza"), replace the "google_maps_key"
  string in this file.
  -->
  <string name="google_maps_key" templateMergeStrategy="preserve" translatable="false">YOUR_KEY_HERE</string>
</resources>
```

Project configuration

- **build.gradle** should have a google play services dependency

```
compile 'com.google.android.gms:play-services:10.0.1'
```

- **the manifest** should define the proper permissions

```
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
```

```
<uses-permission android:name="android.permission.INTERNET" />
```

- **the manifest** should define the key and play services version

```
<meta-data
```

```
    android:name="com.google.android.geo.API_KEY"
```

```
    android:value="@string/google_maps_key" />
```

```
<meta-data
```

```
    android:name="com.google.android.gms.version"
```

```
    android:value="@integer/google_play_services_version" />
```

Setting up without Android Studio

- Detailed instructions

- <https://developers.google.com/maps/documentation/android-api/config?hl=fr>
- https://developers.google.com/maps/documentation/android-api/signup?hl=fr#obtenir_une_cle_dapi_android

- For Xamarin

- https://developer.xamarin.com/guides/android/platform_features/maps_and_location/maps/part_2_-_maps_api/
- https://developer.xamarin.com/guides/android/platform_features/maps_and_location/maps/obtaining_a_google_maps_api_key/

- Documentation

- <https://developers.google.com/maps/documentation/android-api/intro?hl=fr>

How to...

Emulate

Android

With Genymotion

- Genymotion is a free, powerful Android emulator
 - <https://www.genymotion.com/fun-zone/>
 - Download & Install
 - Launch emulation for pre-defined devices
- Plugin for Android Studio
 - Use from Android Studio: <https://www.genymotion.com/plugins/>
- Enable use of Google Play Services (needed for maps)
 - <https://inthecheesefactory.com/blog/how-to-install-google-services-on-genymotion/en>

How to...

Write better tests

i've seen things...

terrible things...

What I don't want to see anymore

- You moved your tests to proper folders...
 - But did not update the package declaration

```
1 package providers;
```

Package name 'providers' does not correspond to the file path 'fr.unice.iut.shapes.providers' [more...](#) (Ctrl+F1)

- `assertTrue(true)`
 - or `assertEquals(true, true)`
 - **=> A test that is not implemented should fail**
- Copy-paste of tests between groups
 - spoiler: it shows

**The assert*
family**

Most useful Assert* methods:

- **AssertEquals (expectedValue, testedValue);**
 - `assertEquals("star", shape.getId());`
 - `assertEquals(42.3, point.getX(), 0.01);`
- **AssertNull / AssertNotNull**
 - `assertNull(ShapesProvider.findById("unknown"));`
 - `assertNotNull(ShapesProvider.findById("star"));`
- **AssertTrue / AssertFalse**
 - `assertTrue(ShapesProvider.deleteShape("star"));`
 - `assertFalse(ShapesProvider.deleteShape("unknown"));`

Some more examples

USE `assertEquals(42.3, point.getX(), 0.01);`

NOT `assertTrue(point.getX() == 42.3);`

USE `assertNotEquals(ShapesProvider.findById("star"), shape);`

NOT `assertFalse(shape.equals(ShapesProvider.findById("star")));`

USE `assertNotNull(ShapesProvider.findById("star"));`

NOT `assertTrue(ShapesProvider.findById("star") != null);`

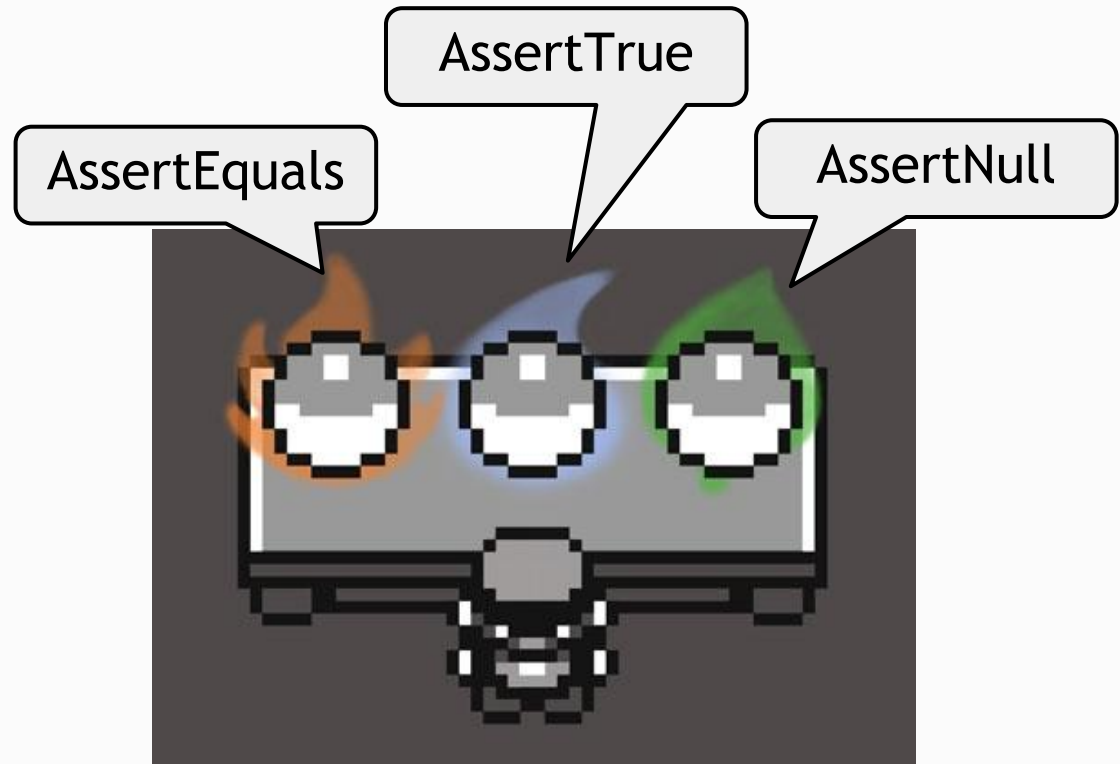
USE `assertNull(ShapesProvider.findById("unknown"));`

NOT `assertEquals(null, ShapesProvider.findById("unknown"));`

Chose wisely !

Think about:

- Correctness
- Readability
- Extensibility



**What are unit
tests?**

What should this methods test?

```
//Method test of createShape from the class ShapeProvider
@Test
public void testCreateShape()
{

}
}
```

```
//Method test of deleteShape from the class ShapeProvider
@Test
public void testDeleteShape()
{

}
}
```

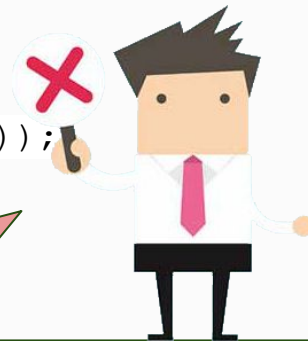


ShapesProvider.createShape(..) and ShapesProvider.deleteShape(..)

What is it ACTUALLY testing?

```
//Method test of createShape from the class ShapeProvider
@Test
public void testCreateShape()
{
    int nb = allShapes.size();
    allShapes.add(new Shape());
    assertEquals(nb+1, allShapes.size());
}
```

```
//Method test of deleteShape from the class ShapeProvider
@Test
public void testDeleteShape()
{
    int nb = allShapes.size();
    allShapes.remove(0);
    assertEquals(nb-1, allShapes.size());
}
```



ArrayList.add(..) and ArrayList.remove(..)

UNIT Test

- Test the behavior of a single element (method, class, ...)
- When you test a method, it is a black box
 - You know pre-conditions
 - You know parameters it takes as input
 - You know the exceptions it throws and when
 - You test what it returns, exceptions and post-conditions
- The implementation of the method should not change the code needed to test it
 - You could write tests BEFORE your write code
= Test Driven Development (TDD)

What is it ACTUALLY testing?

```
//Method test of createShape from the class ShapeProvider
```

```
@Test
```

```
public void testCreateShape()
```

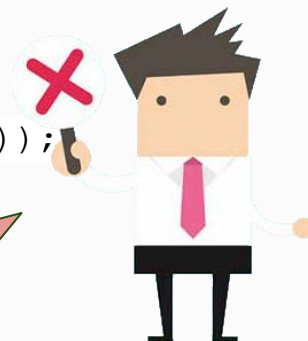
```
{  
    int nb = allShapes.size();  
    allShapes.add(new Shape());  
    assertEquals(nb+1, allShapes.size());  
}
```

```
//Method test of deleteShape from the class ShapeProvider
```

```
@Test
```

```
public void testDeleteShape()
```

```
{  
    int nb = allShapes.size();  
    allShapes.remove(0);  
    assertEquals(nb-1, allShapes.size());  
}
```



ArrayList.add(..) and ArrayList.remove(..)

Makes more sense, doesn't it?

```
//Method test of createShape from the class ShapeProvider
```

```
@Test
```

```
public void testCreateShape()
```

```
{
```

```
    int nb = ShapesProvider.getAllShapes().size();
```

```
    ShapesProvider.createShape(new Shape());
```

```
    assertEquals(nb+1, ShapesProvider.getAllShapes().size());
```

```
}
```

```
//Method test of deleteShape from the class ShapeProvider
```

```
@Test
```

```
public void testDeleteShape()
```

```
{
```

```
    int nb = ShapesProvider.getAllShapes().size();
```

```
    ShapesProvider.deleteShape(ShapesProvider.findShapeById("star"));
```

```
    assertEquals(nb-1, ShapesProvider.getAllShapes().size());
```

```
}
```

But it's not enough....

```
/**
 * Definit une nouvelle {@link Shape} dans le service.
 *
 * @param shape - L'objet {@link Shape} a ajouter
 * @throws ShapeAlreadyExistException si une forme avec le meme identifiant etait
 * deja presente.
 */
public static void createShape(Shape shape) throws ShapeAlreadyExistException {
    if (findShapeById(shape.getId()) != null)
        throw new ShapeAlreadyExistException();
    else
        allShapes.put(shape.getId(), shape);
}

// Our test
@Test
public void testCreateShape() {
    int nb = ShapesProvider.getAllShapes().size();
    ShapesProvider.createShape(new Shape());
    assertEquals(nb+1, ShapesProvider.getAllShapes().size());
}
```



Not good enough! But why?

Towards more complete tests



The Javadoc should be all we need to write the tests!

```
/**  
 * Definit une nouvelle {@link Shape} dans le service.  
 *  
 * @param shape - L'objet {@link Shape} a ajouter  
 * @throws ShapeAlreadyExistException si une forme avec le meme identifiant  
 * etait deja presente.  
 */
```

- Parameters: Test edge cases

- What happens if I give a 'null' Shape, or a Shape with only 1, 2, or no vertex? Should it be added?

=> Writing tests helps us realize that our documentation, specifications or implementation is not good enough

- Maybe an InvalidShapeException could be thrown, or a boolean returned as false if the shape was not added bc it was invalid

Towards more complete tests



The Javadoc should be all we need to write the tests!

```
/**  
 * Definit une nouvelle {@link Shape} dans le service.  
 *  
 * @param shape - L'objet {@link Shape} a ajouter  
 * @throws ShapeAlreadyExistException si une forme avec le meme identifiant  
 * etait deja presente.  
 */
```

- Exceptions: Make sure they occur when expected
 - A `ShapeAlreadyExistException` should be thrown if I add the same shape twice

Test exceptions - v0.1



We can create a separate test method for this case!

```
@Test
public void testCreateAlreadyExistingShape()
{
    // We know star is already in here
    Shape s = new Shape("star");

    try {
        ShapesProvider.createShape(s);
        fail("a ShapeAlreadyExistsException should have been thrown");
    } catch (ShapeAlreadyExistsException e) {
        assertTrue(true);
    }
}
```

NO assertTrue(true) !!!



Test exceptions - v0.5



We can create a separate test method for this case!

```
@Test
public void testCreateAlreadyExistingShape()
{
    // We know star is already in here
    Shape s = new Shape("star");

    try {
        ShapesProvider.createShape(s);
        fail("a ShapeAlreadyExistsException should have been thrown");
    } catch (ShapeAlreadyExistsException e) {
        // doing nothing has the same result
    }
}
```

Meh...



Test exceptions - v1



We can create a separate test method for this case!

```
@Test
public void testCreateAlreadyExistingShape()
{
    int nb = ShapesProvider.getAllShapes().size();
    // We know star is already in here
    Shape s = new Shape("star");

    try {
        ShapesProvider.createShape(s);
        fail("a ShapeAlreadyExistsException should have been thrown");
    } catch (ShapeAlreadyExistsException e) {
        assertEquals(nb+1, ShapesProvider.getAllShapes().size());
    }
}
```

That's more like it...



Test exceptions - Another way



We can create a separate test method for this case!

```
@Test(expected = ShapeAlreadyExistException.class)
public void testCreateAlreadyExistingShape()
{
    int nb = ShapesProvider.getAllShapes().size();
    // We know star is already in here
    Shape s = new Shape("star");
    ShapesProvider.createShape(s);
}
```



Here, the test will fail if the exception is not thrown, and pass otherwise.

- The `@Test(expected ...)` shows directly what the test is for
- It's cleaner and easier to read

Semester Timeline

Timeline

3 Sprints:

- **19/02:** Release (sprint-3) + Technical Report
- **26/03:** Release (sprint-4) + Defense (code review)
- **14/05:** Final release (sprint-5) + Demo to the client

Organize your sprints as you want

don't forget about client value though !

Reminders - tools

- Code hosted on Github Classroom

<https://classroom.github.com/group-assignment-invitations/79979cffe409d0d68c89b9e72d6599c6>

- A JenkinsFile for continuous integration

<http://iut-outils-gl.i3s.unice.fr/jenkins/job/lpidse16-17/job/lpidse16-17/>

- A new Jira project
- Reference tasks in commit messages
- Releases on Github & Jira: sprint-X

Sprint 3

How to start

- Put all your legacy code on the new github repo
 - Describe your code to your teammates
 - Describe your web-services (urls, inputs, outputs)
 - Discuss together what to keep, and how to start
 - Validate this with your coach
- Take the backlog of all your previous projects
 - Recreate all open tasks on the new board
 - Go from it, sort and redefine them with your team

Report - what's expected

February 19 before 23:59 **Late** → **0**

Location

- file report.pdf in a docs/ folder on the repo + tag 'sprint-3'.

Content

- Team organization
- How did you integrate your code? (code review, starting from the database, starting from the desktop before going mobile, integrate one feature at a time at the same time as the mobile interface, etc?)
- Did your initial plan work out and why? Was it predictable?
- Difficulties you faced



GO!!

