

TESTS AUTOMATISÉS



25/06/2018

SOMMAIRE

- Mise en situation
- Qu'est-ce que le BDD?
- Outils utilisés
- Mise en place du BDD
- Structuration du projet
- Scénarios
- Exécution
- Rapport

OBJECTIF



But du projet : Réalisation de tests automatisés sur l'application web EDC



QU'EST-CE QUE LE BDD ?

Qu'est-ce que le BDD?

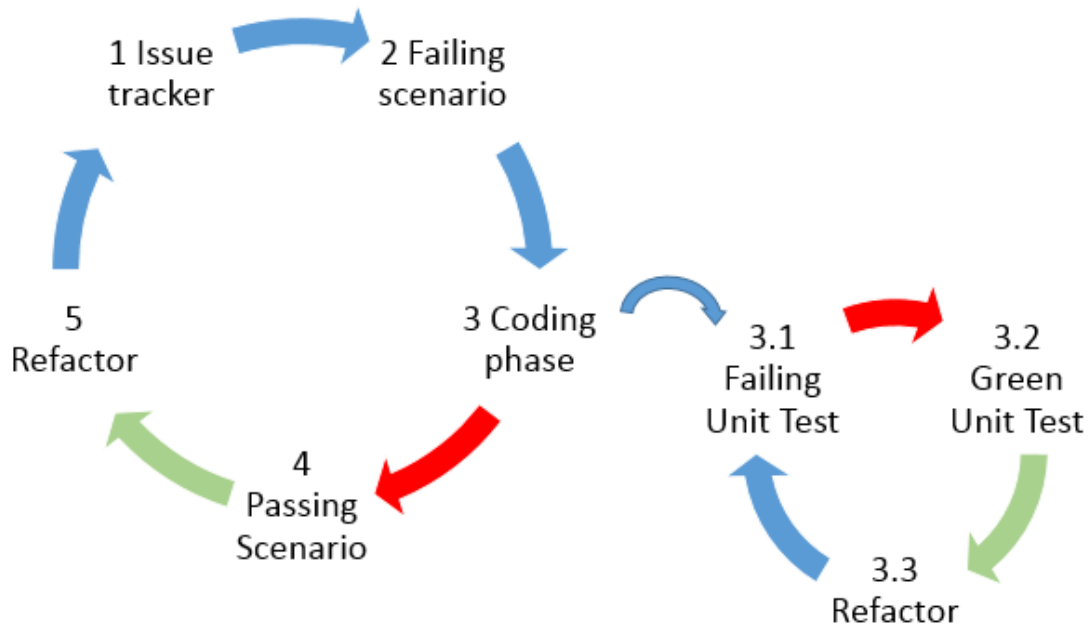
- D'après Wikipédia:

Le **behavior-driven development** (ou **BDD**) est une [méthode agile](#) qui encourage la collaboration entre les développeurs, les responsables qualités, les intervenants non-techniques et les entreprises participant à un projet de logiciel. Il a été conçu en 2003 par Dan North comme une réponse au [Test Driven Development](#).

Le processus BDD met en avant le langage naturel et les interactions dans le processus de développement logiciel. Les développeurs utilisant le BDD utilisent un langage naturel en combinaison avec le langage du domaine [domain driven design](#) pour décrire l'objectif et le bénéfice de leur code. Cela permet aux développeurs de se concentrer sur les raisons pour lesquelles le code doit être créé, plutôt que les détails techniques, et minimise la traduction entre le langage technique dans lequel le code est écrit et le domaine de la langue parlée par les entreprises, les utilisateurs, les intervenants, la gestion de projet...

Qu'est-ce que le BDD?

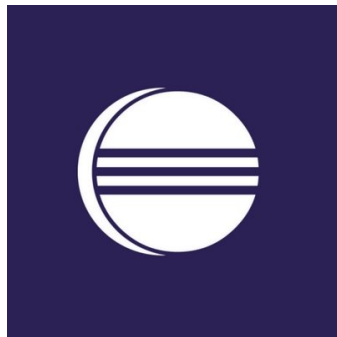
The BDD Process



OUTILS UTILISÉS



Maven



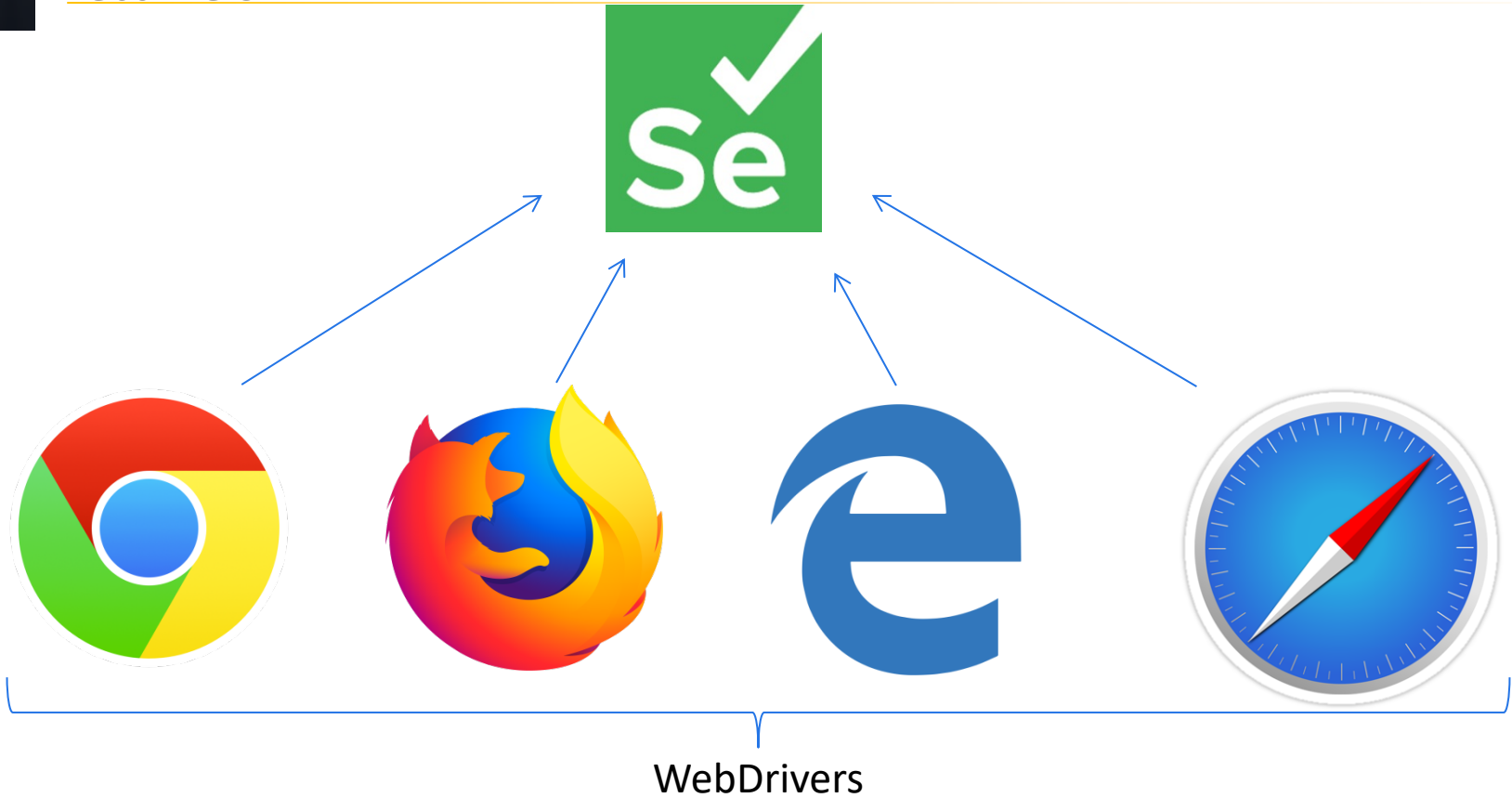
Eclipse



Selenium



Cucumber



Utilisation de AutoITX : librairie externe



Automatisation pour le système



Automatisation pour les interfaces
Web

MISE EN PLACE DU BDD



Ecriture d'un scénario

```
@tag
Feature: Test Login EDC
    The user can log in on edc software.
```

```
Scenario: Test login
    Given I am on the login page
    When I enter the username "stmgr1" and the password "stmgr1"
    And I click on login
    And I wait the url page change
    Then Homepage should be displayed
    And Finish
```

Génération du code

```

@Given("^I am on the login page")
public void i_am_on_the_login_page() {
    // Write code here that turns the phrase above into concrete actions
    throw new PendingException();
}

@When("^I enter the username \"([^\"]*)\" and the password \"([^\"]*)\"")
public void iEnterTheUsernameAndThePassword(String username, String password) {
}

@When("^I click on login")
public void i_click_on_login() {
    // Write code here that turns the phrase above into concrete actions
    throw new PendingException();
}

@Then("Homepage should be displayed")
public void homepage_should_be_displayed() {
    // Write code here that turns the phrase above into concrete actions
    throw new PendingException();
}

@Then("^LoginPage should be displayed$")
public void loginpageShouldBeDisplayed() {
    // Write code here that turns the phrase above into concrete actions
    throw new PendingException();
}

@When("I log with the username \"([^\"]*)\" and the password \"([^\"]*)\"")
public void i_log_with_the_username_and_the_password(String strUser, String strPwd) throws InterruptedException {
    // Write code here that turns the phrase above into concrete actions
    throw new PendingException();
}
    
```

Écriture : Optimisation scénarios

```
@Given("^I am on the login page")
public void i_am_on_the_login_page() {
    // Write code here that turns the phrase above into concrete actions
    // throw new PendingException();
    webdriver = WebDriverFactory.getDriver();
    System.out.println(webdriver);
    WebDriverFactory.Setup(webdriver);
    loginTest = new LoginPage(webdriver);
}

@When("^I enter the username \"([^\"]*)\" and the password \"([^\"]*)\"")
public void iEnterTheUsernameAndThePassword(String username, String password) {
    loginTest.setUsername(username).setPassword(password);
}

@When("^I click on login")
public void i_click_on_login() {
    // Write Code here that turns the phrase above into concrete actions
    // throw new PendingException();
    loginTest.ClickLogin();
}

@Then("Homepage should be displayed")
public void homepage_should_be_displayed() {
    // Write code here that turns the phrase above into concrete actions
    // throw new PendingException();
    Assert.assertFalse(loginTest.isLoginPage());
}

@Then("^LoginPage should be displayed$")
public void loginpageShouldBeDisplayed() {
    Assert.assertTrue(loginTest.isLoginPage());
}

@When("I log with the username \"([^\"]*)\" and the password \"([^\"]*)\"")
public void i_log_with_the_username_and_the_password(String strUser, String strPwd) throws InterruptedException {
    // Write Code here that turns the phrase above into concrete actions
    // throw new PendingException();
    TimeUnit.SECONDS.sleep(2);
    iEnterTheUsernameAndThePassword(strUser, strPwd);
    loginTest.ClickLogin();
}
```

```
@tag
Feature: Test Login EDC
    The user can log in on edc software.
```

```
Scenario: Test login
    Given I am on the login page
    When I enter the username "stmgr1" and the password "stmgr1"
    And I click on login
    And I wait the url page change
    Then Homepage should be displayed
    And Finish
```

```
@tag
Feature: Create a Brick
    Roles can or not create bricks
```

```
@tag1
Scenario: As a contributor, i can not create a brick
    Given I am on the login page
    When I log with the username "cont1" and the password "dreyfuss"
    And I wait the url page change
    Then Homepage should be displayed
    When I click on Documentation
    And select the product "EDC" and the version "Version 1"
    And can not create a new brick
```

L'optimisation de scénarios par regroupement des phrases permet d'en réduire la quantité et donc les répétitions.

Écriture : Scénarios pour chaque rôle

Base commune pour les scénarios, adaptations en fonction des rôles

Contributor

@tag

Feature: Create a Brick

Roles can or not create bricks

@tag1

Scenario: As a contributor, i can not create a brick

```
Given I am on the login page
When I log with the username "cont1" and the password "dreyfuss"
And I wait the url page change
Then Homepage should be displayed
When I click on Documentation
And select the product "EDC" and the version "Version 1"
And can not create a new brick
```

StoreHouseManager

Feature: Create a Brick

Roles can or not create bricks

@tag1

Scenario: As a storehouse Manager, i can create a brick

```
Given I am on the login page
When I log with the username "stmgr1" and the password "stmgr1"
And I wait the url page change
Then Homepage should be displayed
When I click on Documentation
And select the product "EDC" and the version "Version 1"
And can create a new brick
And enter the title "AZERT" and choose the type "Window"
```


STRUCTURATION DU PROJET

Séparation de chaque rôle :

- ▲ src/test/resources
 - ▲ scenario
 - ▲ contributor
 - editor
 - ▲ storeHouseManager
 - config
 - editor
 - publishing
 - login.feature

Le login est commun à tous

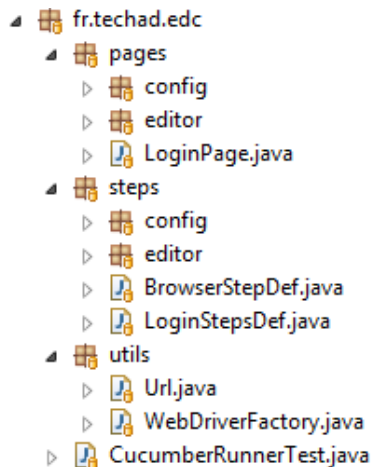
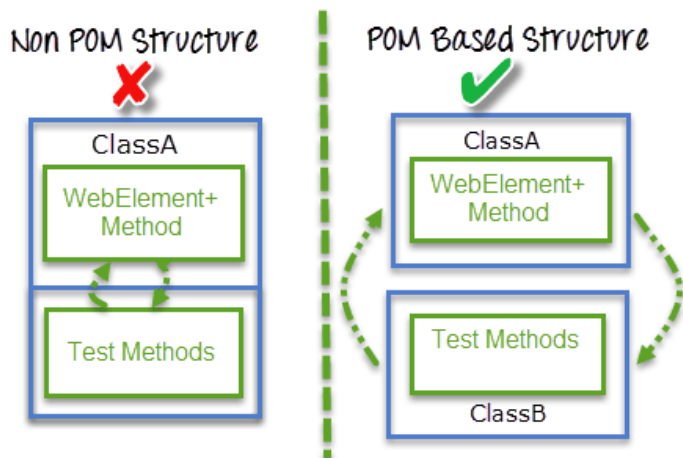
```
@tag
Feature: Test Login EDC
  The user can log in on edc software.

Scenario: Test login
  Given I am on the login page
  When I enter the username "stmgr1" and the password "stmgr1"
  And I click on login
  And I wait the url page change
  Then Homepage should be displayed
  And Finish
```



Meilleure approche

- La structure en Page Object Model (POM) est un paradigme ou design pattern devenu très populaire dans les tests d'interface car cela évitait la duplication de code
- Grâce à ce modèle les tests sont automatiquement impactés lors d'une modification de code



← Pages (code)

← Définitions des méthodes pour les tests

Évolution de la structure

- src/test/java
 - fr.techad.edc
 - pages
 - config
 - products
 - strategies
 - topologies
 - users
 - ConfigMenuComp.java
 - editor
 - contributors
 - documentationTypes
 - editorArea
 - ConfigEditorMenuComp.java
 - CreateBrickPage.java
 - UploadImg.java
 - VersionSelector.java
 - LoginPage.java
 - steps
 - config
 - CreateProductStepsDef.java
 - CreateUserSetpsDef.java
 - StrategiesSetpsDef.java
 - editor
 - ArticleImgStepsDef.java
 - ArticlesStepsDef.java
 - CreateBrickStepsDef.java
 - EditBrickStepsDef.java
 - ImgTestStepsDef.java
 - BrowserStepDef.java
 - LoginStepsDef.java
 - utils
 - Url.java
 - WebDriverFactory.java
 - CucumberRunnerTest.java

Pour chaque page, répartition en zones d'affichage : évite la duplication du code et permet une bonne lisibilité

The screenshot shows the EDC Software Compagny web application. The sidebar menu on the left has 'Product Management' selected, which is highlighted with a blue box and an arrow pointing from the 'ConfigMenuComp.java' file in the file structure. The top status bar displays: 'Used users : 19 / 40', 'Storage space used : 0.55%', and 'Used storehouses : 2 / 2'. The main content area is titled 'Products of TECH' and contains a table with the following data:

	Name ^	Version ^	Type ^	Vendor ^	Status ^
<input type="checkbox"/>	EDC	Version 2	WEB	EDC Software Compagny	✓
<input type="checkbox"/>	EDC	Version 2 CpTest	WEB	EDC Software Compagny	✓
<input type="checkbox"/>	EDC	Version 2 Cp 200618	WEB	EDC Software Compagny	✓
<input type="checkbox"/>	EDC	Version 1	WEB	TECH ADVANTAGE	✓
<input type="checkbox"/>	EDC IDE Eclipse	Version 2	RCP	EDC Software Compagny	✓

EXÉCUTION

■ Via Eclipse

The screenshot shows the Eclipse IDE interface. On the left, the Project Explorer displays a project structure with a 'Cucumber Feature' folder containing 'createproduct.feature' and 'imgTest.feature'. The 'imgTest.feature' file is selected. The main editor area shows the 'Cucumber Feature Runner' dialog for the file 'imgTest.feature'. The dialog has tabs for 'Arguments', 'JRE', 'Classpath', and 'Common'. The 'Project' field is set to 'scenarios'. The 'Feature Path' field is set to 'C:\Users\lejeune\git\edc-req\scenarios\target\test-classes\scenario\storeHouseManager\editor\editArticleImg.feature'. The 'Glue' field is set to 'classpath:'. The 'Run As' context menu is open, showing the option '1 Cucumber Feature'.

Exécution des Features via les configurations de lancements

```
package fr.techad.edc;  
  
import org.junit.runner.RunWith;  
  
@RunWith(Cucumber.class)  
@CucumberOptions(monochrome = true, features = { "classpath:scenario" }, glue = {  
    "fr/techad/edc/steps" }, plugin = { "pretty", "html:target/cucumber" })  
public class CucumberRunnerTest {  
}
```

Grâce aux options de lancement de Cucumber, il est possible de lancer l'exécution des tests les uns à la suite des autres en fonction des rôles.

```
[INFO] Scanning for projects...
[INFO]
[INFO] < fr.techad.edc-scenarios-bdd >-----
[INFO] Building edc-scenarios-bdd 0.0.1-SNAPSHOT
[INFO] [1 jar]
[WARNING] The POM for net.java.dev.jna:jna:jar:4.1.0 is invalid, transitive dependencies (if any) will not be available, enable debug logging for more details
[WARNING] The POM for net.sourceforge.cssparser:cssparser:jar:0.9.22 is invalid, transitive dependencies (if any) will not be available, enable debug logging for more details
[WARNING] The POM for net.sourceforge.htmlunit:neko-htmlunit:jar:2.25 is invalid, transitive dependencies (if any) will not be available, enable debug logging for more details
[WARNING] The POM for org.eclipse.jetty.websocket:websocket-common:jar:9.4.3.v20170317 is invalid, transitive dependencies (if any) will not be available, enable debug logging for more details
[WARNING] The POM for org.eclipse.jetty:jetty-io:jar:9.4.1.v20170120 is invalid, transitive dependencies (if any) will not be available, enable debug logging for more details
[WARNING] The POM for org.eclipse.jetty:jetty-util:jar:9.4.1.v20170120 is invalid, transitive dependencies (if any) will not be available, enable debug logging for more details
Downloading from central: https://repo.maven.apache.org/maven2/org/seleniumhq/selenium/selenium-api/maven-metadata.xml
Downloaded from central: https://repo.maven.apache.org/maven2/org/seleniumhq/selenium/selenium-api/maven-metadata.xml (3.4 kB at 3.9 KB/s)
[WARNING] The POM for org.seleniumhq.selenium:selenium-api:jar:3.11.0 is invalid, transitive dependencies (if any) will not be available, enable debug logging for more details
Downloading from central: https://repo.maven.apache.org/maven2/org/seleniumhq/selenium/selenium-support/maven-metadata.xml
Downloaded from central: https://repo.maven.apache.org/maven2/org/seleniumhq/selenium/selenium-support/maven-metadata.xml (3.6 kB at 100 KB/s)
[WARNING] The POM for org.seleniumhq.selenium:selenium-remote-driver:jar:3.7.1 is invalid, transitive dependencies (if any) will not be available, enable debug logging for more details
[WARNING] The POM for org.seleniumhq.selenium:selenium-remote-driver:jar:3.10.0 is invalid, transitive dependencies (if any) will not be available, enable debug logging for more details
[INFO]
[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ edc-scenarios-bdd ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] Copying 1 resource
[INFO]
[INFO] --- maven-compiler-plugin:3.7.0:compile (default-compile) @ edc-scenarios-bdd ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] --- maven-resources-plugin:2.6:testResources (default-testResources) @ edc-scenarios-bdd ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] Copying 14 resources
[INFO]
[INFO] --- maven-compiler-plugin:3.7.0:testCompile (default-testCompile) @ edc-scenarios-bdd ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] --- maven-surefire-plugin:2.12.4:test (default-test) @ edc-scenarios-bdd ---
[INFO] Surefire report directory: C:\Users\lejeune\git\edc-req\scenarios\target\surefire-reports

T E S T S
Running fr.techad.edc.CucumberRunnerTest
@tag
Feature: Create a Brick
  Roles can or not create bricks
    @tag @tag1
    Scenario: As a contributor, i can not create a brick # scenario
    /contributor/editor/createBrick.feature:6
Starting ChromeDriver 2.38.552522 (4376bfbd8f86763dc9dcb) on port 38401
Only local connections are allowed.
```

Un fois la mise en place de Maven, nous exécutons la série de tests tout en choisissant le navigateur :
`mvn test -DBROWSER=firefox`

Maven va lancer une campagne de tests sur firefox et en générer un rapport



RAPPORT

Génération d'un rapport (HTML)

@tag Feature: add articles

add articles to the editor area and verify doubles and the order

@tag @tag1 Scenario: As a storehouse manager, i can add all articles

Given I am on the login page

When I log with the username "stmgr1" and the password "stmgr1"

And I wait the url page change

Then Homepage should be displayed

When I click on Documentation

And select the product "EDC" and the version "Version 1"

And Go to articles of the brick "AZERT"

Then can add articles

Then add all contributors on the edit page

And can edit articles

@tag Feature: Edit Parameters

@tag @tag1 Scenario: As a storeHouse manager, i can add replace/delete an img

Given I am on the login page

When I log with the username "stmgr1" and the password "stmgr1"

```
org.openqa.selenium.UnhandledAlertException: :  
Build info: version: '3.4.0', revision: 'unknown', time: 'unknown'  
System info: host: 'TAPC109', ip: '172.20.16.109', os.name: 'Windows 7', os.arch: 'amd64', os.version: '6.1', java.version: '1.8.0_102'  
Driver info: org.openqa.selenium.firefox.FirefoxDriver  
Capabilities [{moz:profile=C:\Users\lejeune\AppData\Local\Temp\rust_mozprofile.BGbp1zMRZPee, rotatable=false, timeouts={implicit:0, pageLoad:300000, script:30000}, unhandledPromptBehavior:dismiss, userAgent:Mozilla/5.0 (Windows NT 6.1; rv:52.0) Gecko/20100101 Firefox/52.0}]  
*** Element info: (Using name, value=username)  
at org.openqa.selenium.remote.http.W3CHttpResponseCodec.decode(W3CHttpResponseCodec.java:113)  
at org.openqa.selenium.remote.http.W3CHttpResponseCodec.decode(W3CHttpResponseCodec.java:45)  
at org.openqa.selenium.remote.HttpCommandExecutor.execute(HttpCommandExecutor.java:164)  
at org.openqa.selenium.remote.service.DriverCommandExecutor.execute(DriverCommandExecutor.java:82)  
at org.openqa.selenium.remote.RemoteWebDriver.execute(RemoteWebDriver.java:637)  
at org.openqa.selenium.remote.RemoteWebDriver.findElement(RemoteWebDriver.java:410)  
at org.openqa.selenium.remote.RemoteWebDriver.findElementByName(RemoteWebDriver.java:485)  
at org.openqa.selenium.By$ByName.findElement(By.java:303)  
at org.openqa.selenium.remote.RemoteWebDriver.findElement(RemoteWebDriver.java:402)  
at fr.techad.edc.pages.LoginPage.setUsername(LoginPage.java:25)  
at fr.techad.edc.steps.LoginStepsDef.iEnterTheUsernameAndThePassword(LoginStepsDef.java:32)  
at fr.techad.edc.steps.LoginStepsDef.i_log_with_the_username_and_the_password(LoginStepsDef.java:59)  
at *.I log with the username "stmgr1" and the password "stmgr1"(scenario/storeHouseManager/editor/editBrickParameters.java:10)
```

And I wait the url page change

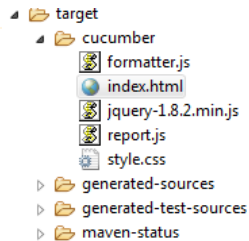
Then Homepage should be displayed

When I click on Documentation

And select the product "EDC" and the version "Version 1"

And Go to parameters of the brick "AZERT"

Then can edit the parameters page with the description "TEST" and main key "AAA" and subKey "BBB"



Après avoir lancé tous les scénarios, Cucumber génère un rapport accessible au format HTML

Phrases passées

Phrases avec erreurs

Phrases ignorées



SIEGE SOCIAL : 232 AVENUE NAPOLEON BONAPARTE 92500 RUEIL-MALMAISON
contact@tech-advantage.com | +33 (0) 1 56 84 02 00 | www.tech-advantage.com

S.A au capital de 219 300 € | R.C.S de Nanterre : B 408 832 301 | SIRET 408 832 301
00019 Code activité 6202A | T.V.A N° FR09 408 832 301