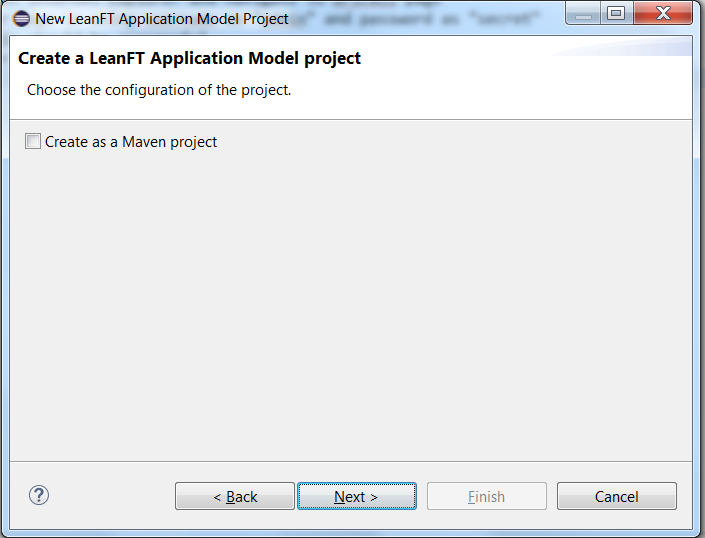
Créer un projet « LeanFT Application Model Project » :

Dans Eclipse , cliquer sur :File ->New ->Other

Puis sélectionner « LeanFT Application Model Project »:

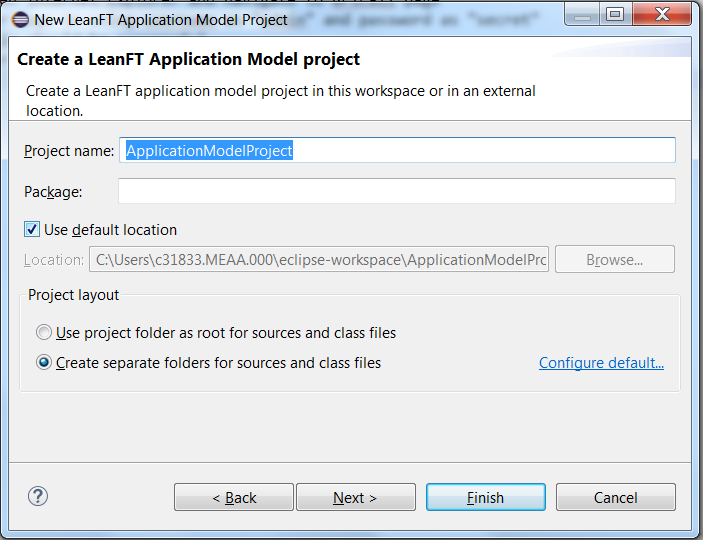


Cliquer sur « Next » :

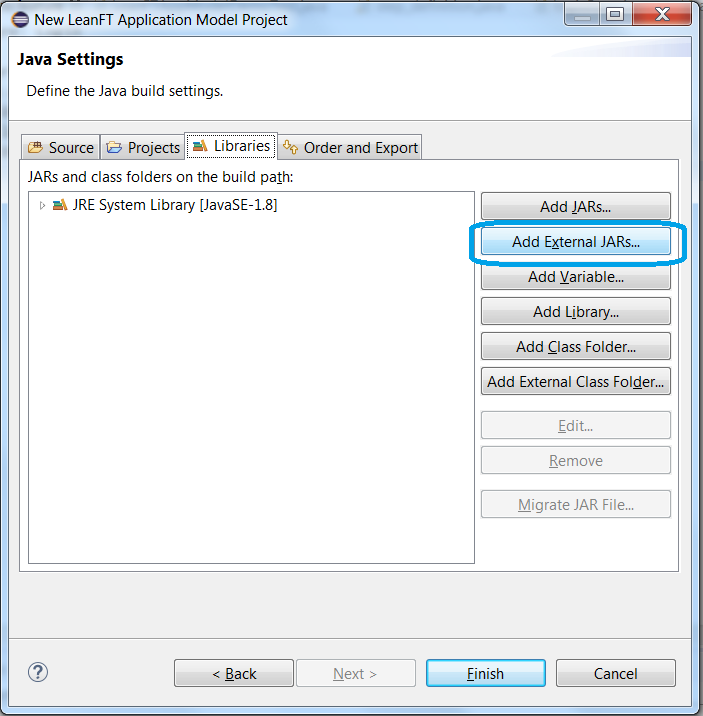


« Next »

Saisir un nom du projet :



« Finish »

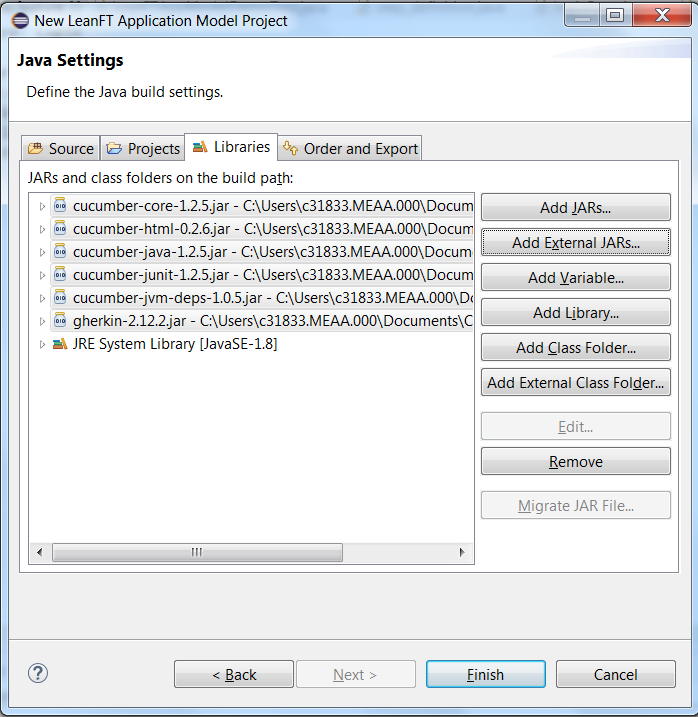


« Add External JARs »



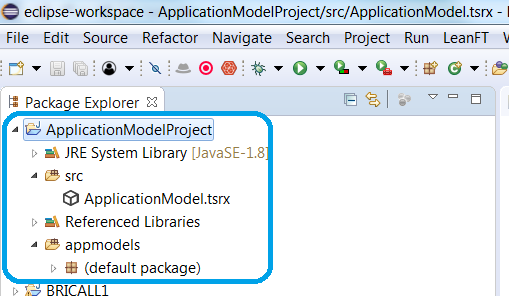
Les Jars à ajouter :

* gherkin-2.12.2.jar
* cucumber-jvm-deps-1.0.5.jar
* cucumber-junit-1.2.5.jar
* cucumber-java-1.2.5.jar
* cucumber-html-0.2.6.jar
* cucumber-core-1.2.5.jar

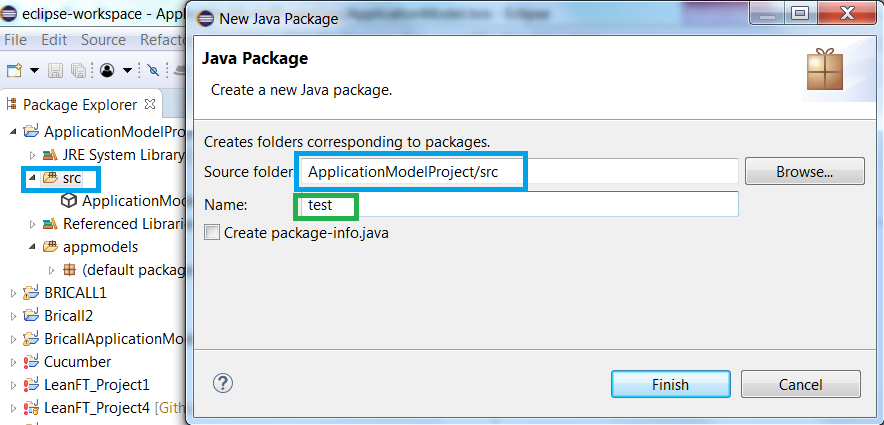


« Finish »

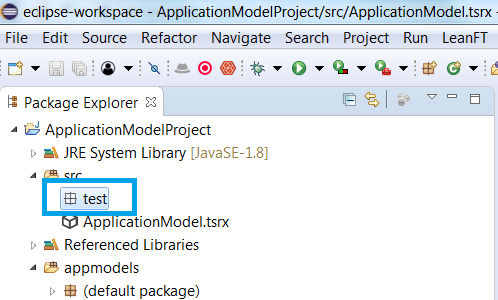
Le projet « ApplicationModelProject » est créé :



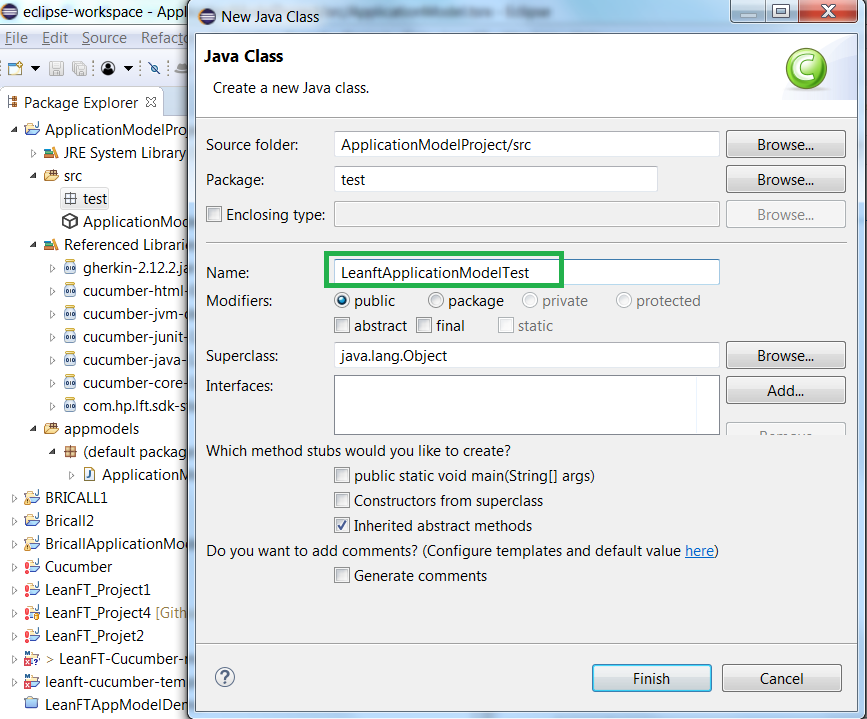
Créer un nouveau package dans le dossier « src » :



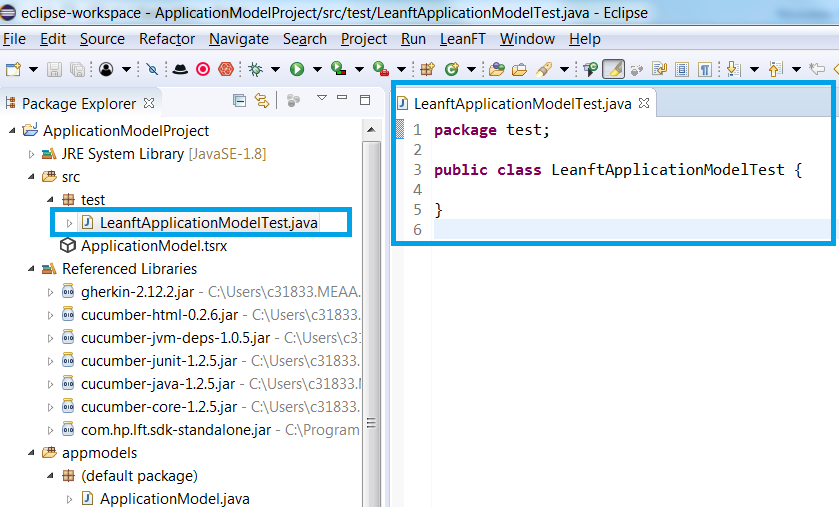
« Finish »



Créer une nouvelle class « LeanftApplicationModelTest» dans le package « test » :



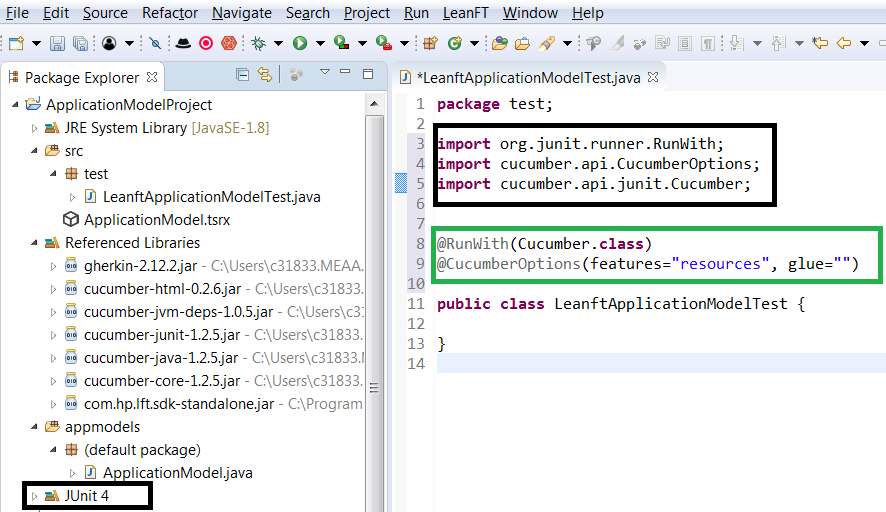
« Finish »



Ajouter les annotations Cucumber  suivantes:

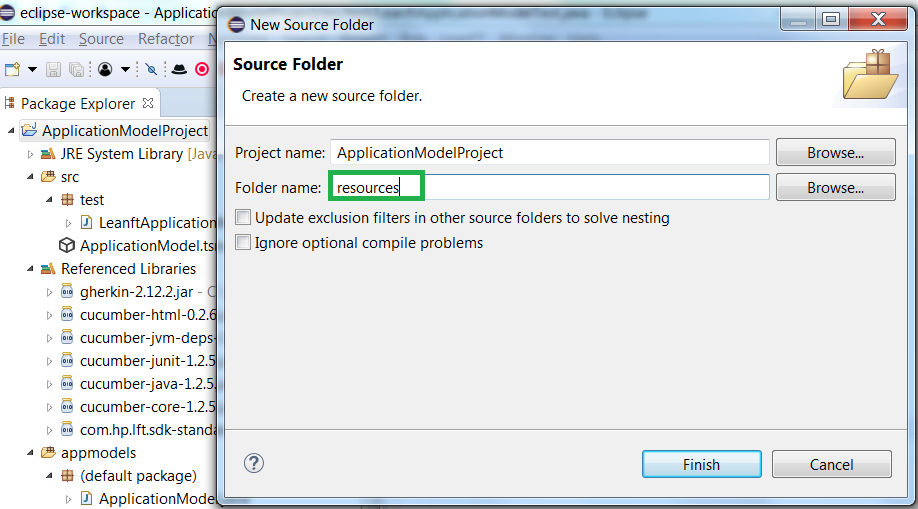
@RunWith(Cucumber.**class**)

@CucumberOptions(features="resources", glue="")



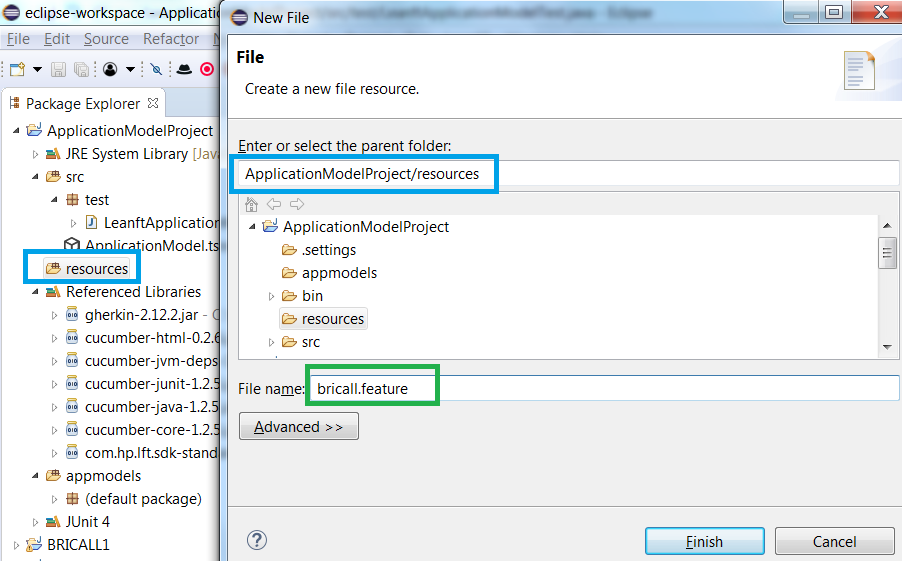
N’oubliez pas d’importer les ressources nécessaires et d’ajouter la librairie Junit.

Creér un nouveau dossier « resources » :



« Finish »

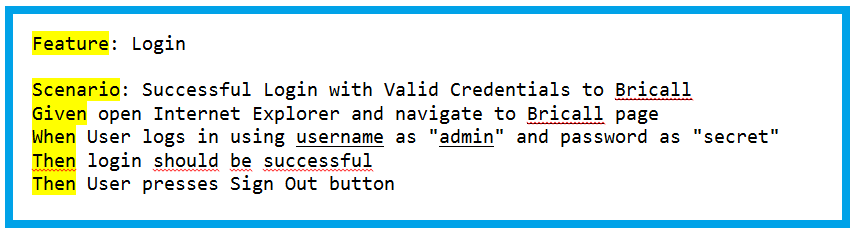
Ensuite dans le dossier « resources » ajouter un fichier avec l’extension (.feature), ce fichier représente la feature à implémenter.

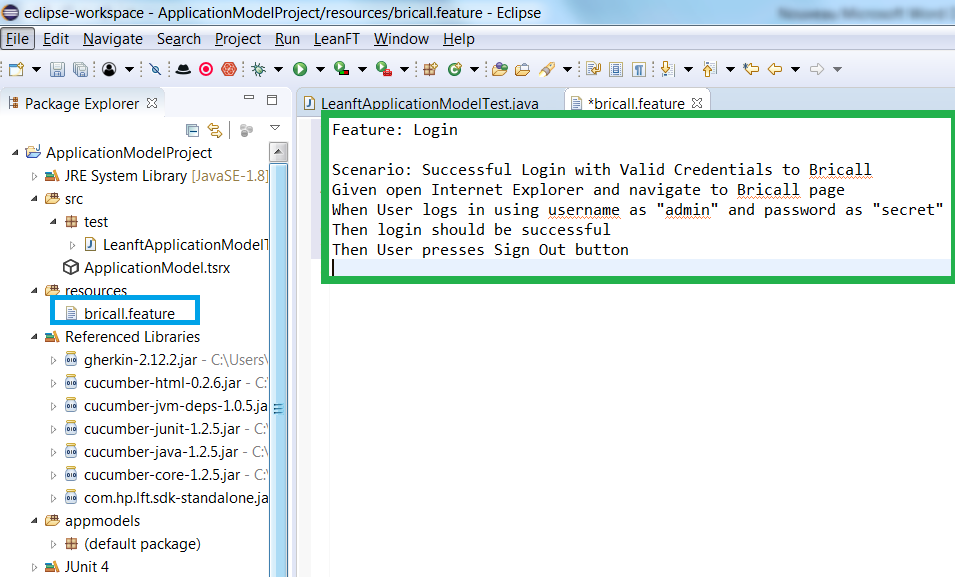


« Finish »

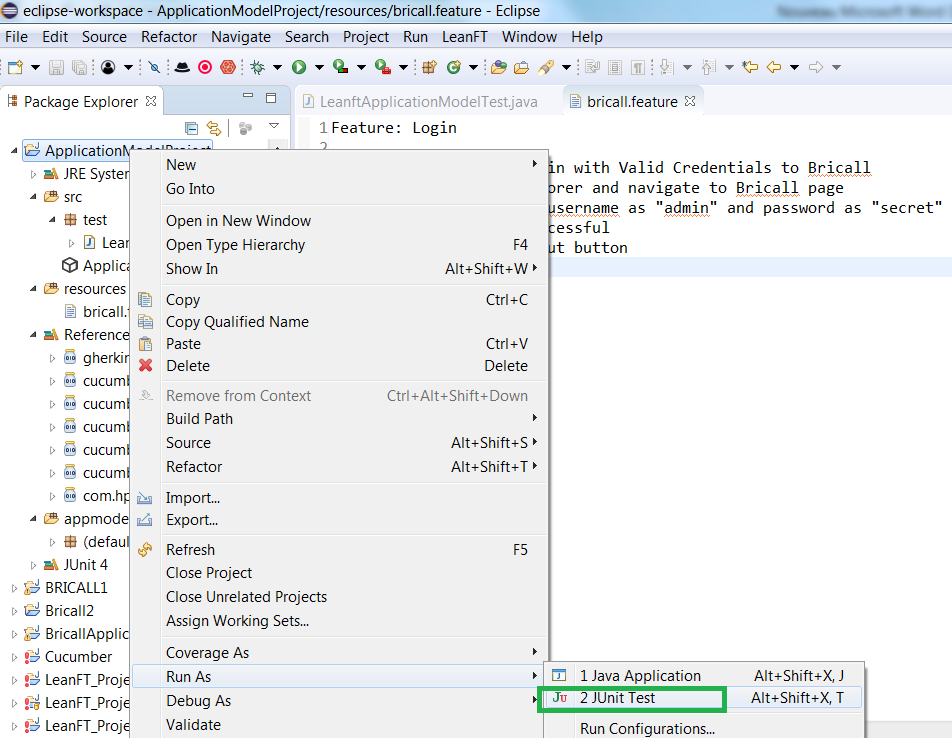
Au niveau de la feature crée, mettre le scénario à automatiser en respectant la syntaxe du langage gherkin :

La feature « bricall » ci-dessous représente le scénario d’authentification à l’application «  BRICALL ».

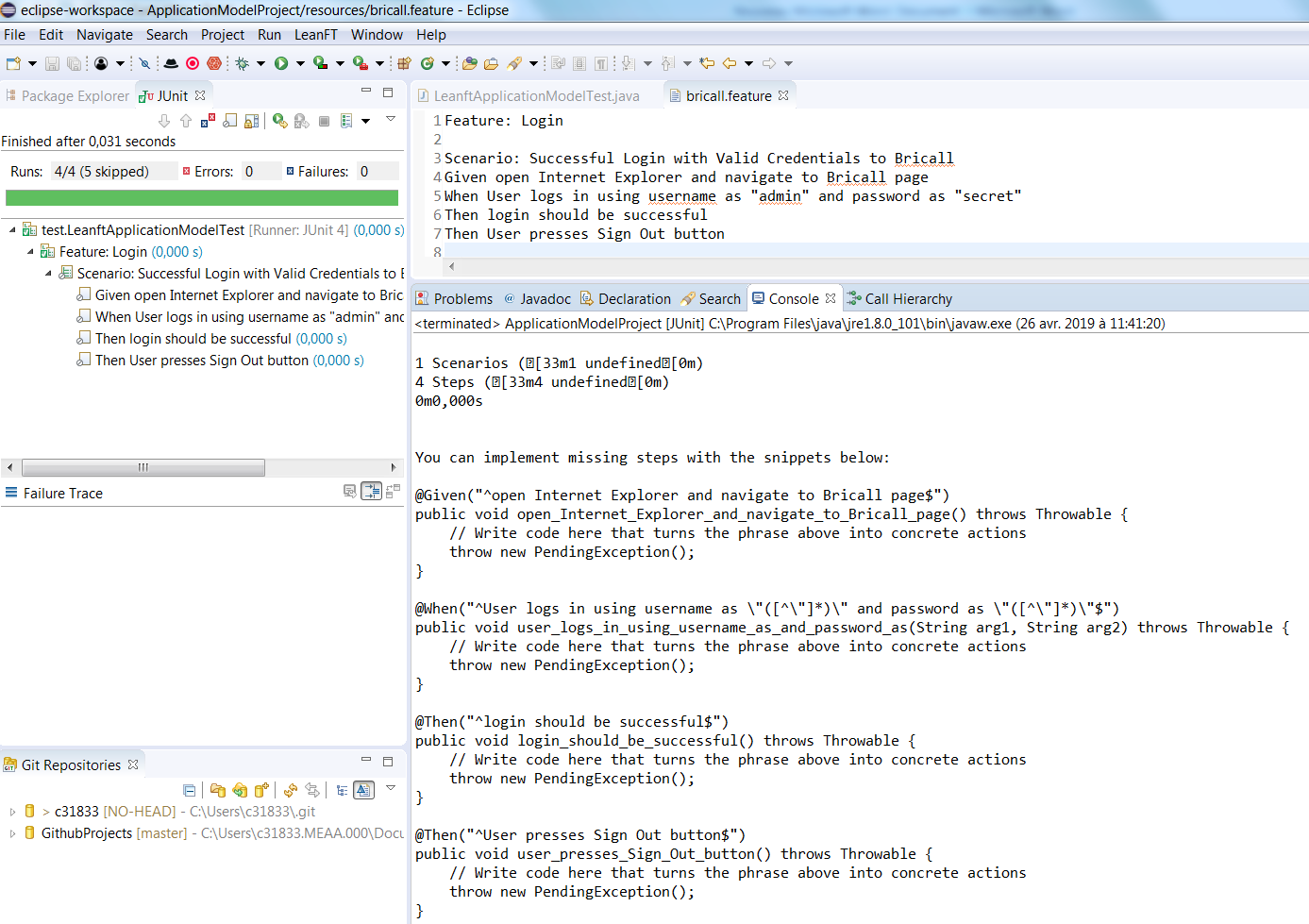




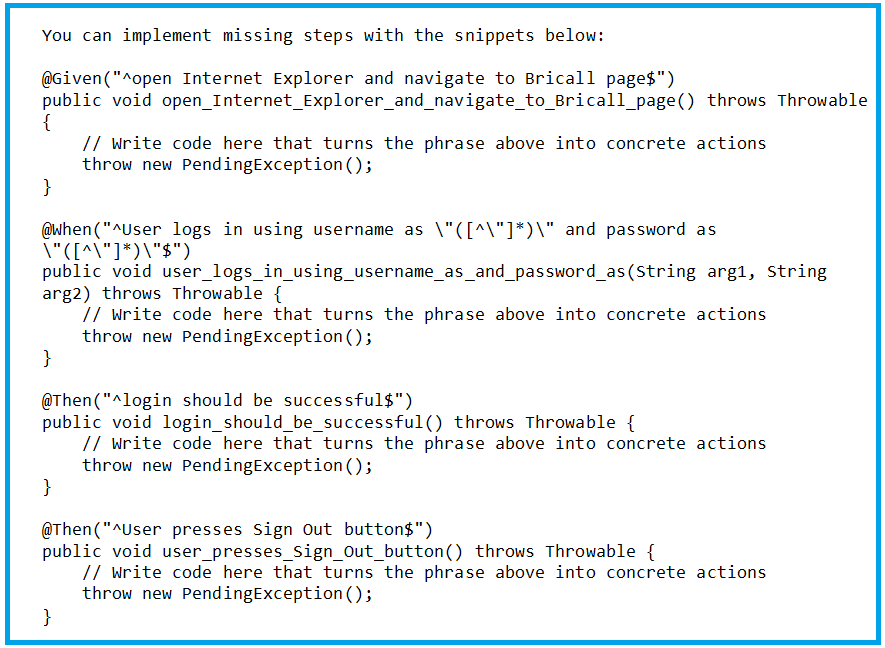
Exécuter le projet par Junit Test :



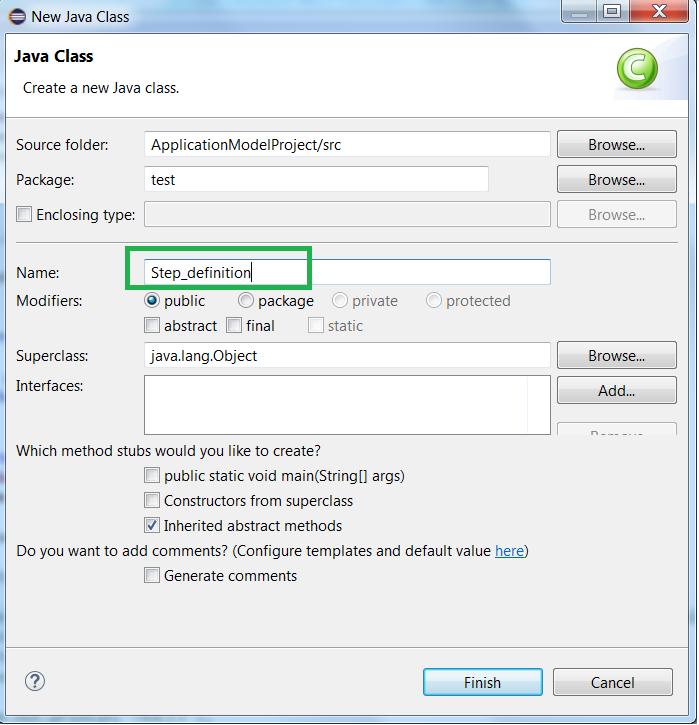
Le projet est exécuté avec succès.



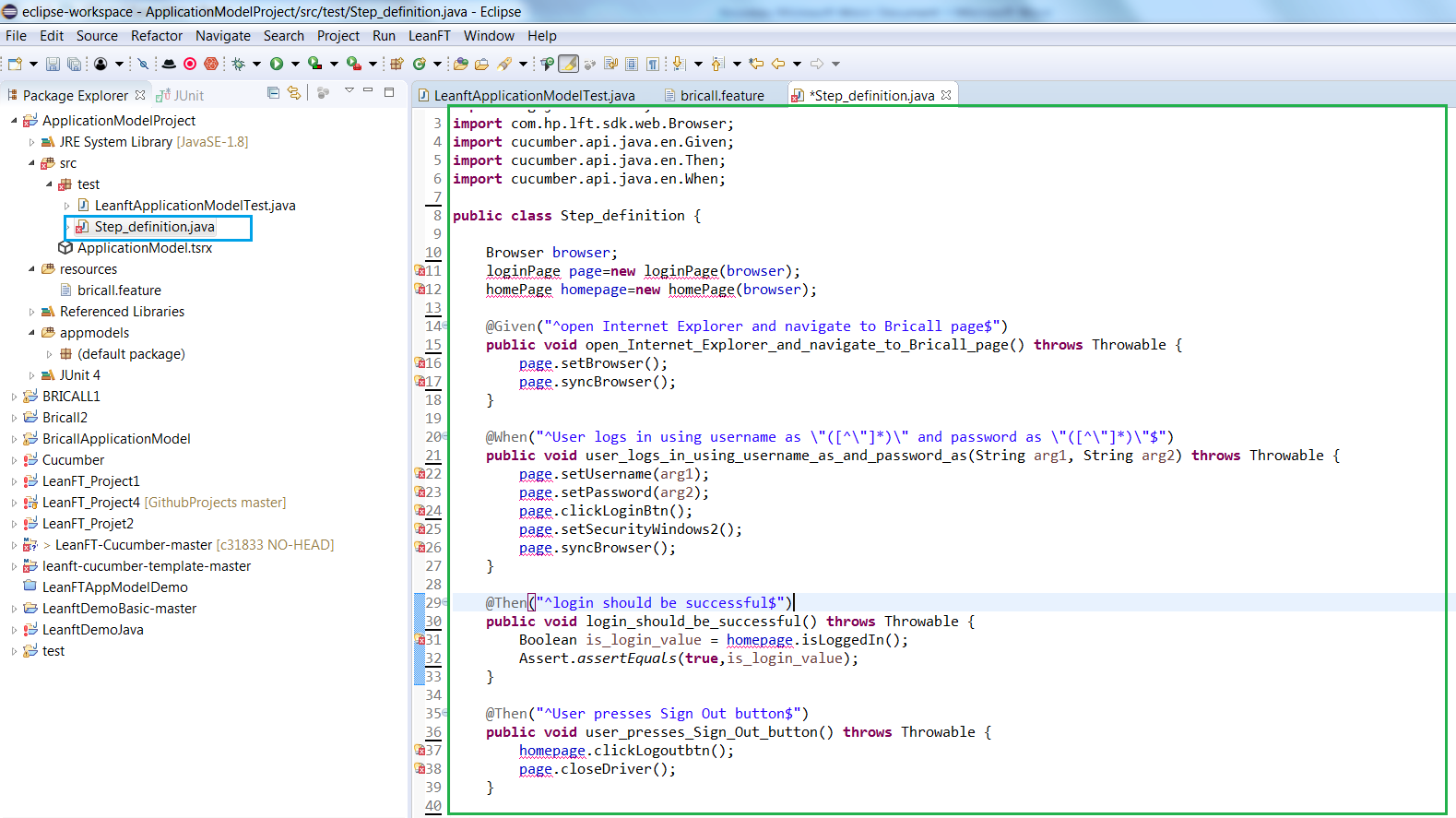
Au niveau de la console, le message suivant apparait :



Créer une nouvelle class pour implémenter les méthodes ci-dessus :



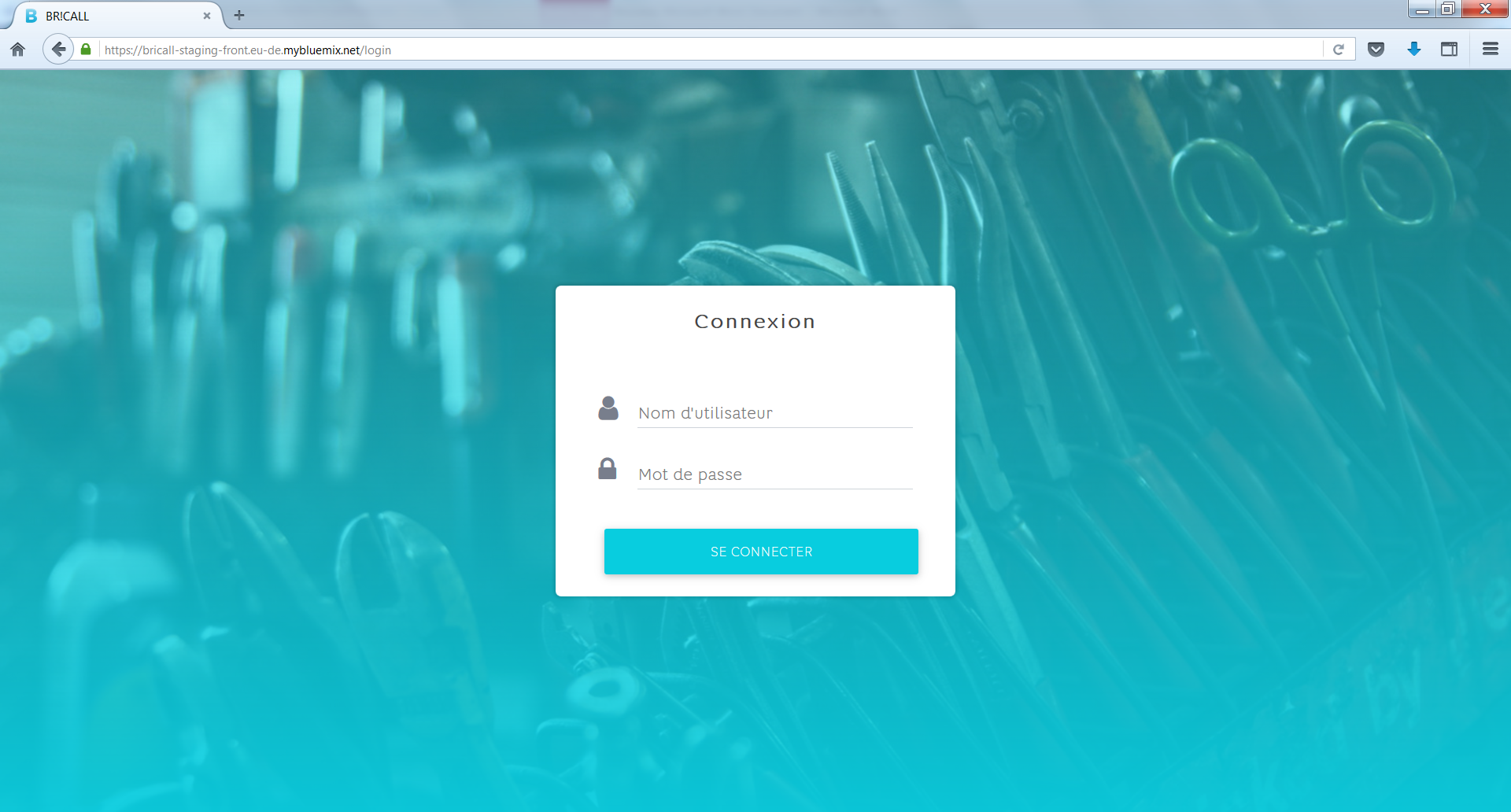
« Finish »

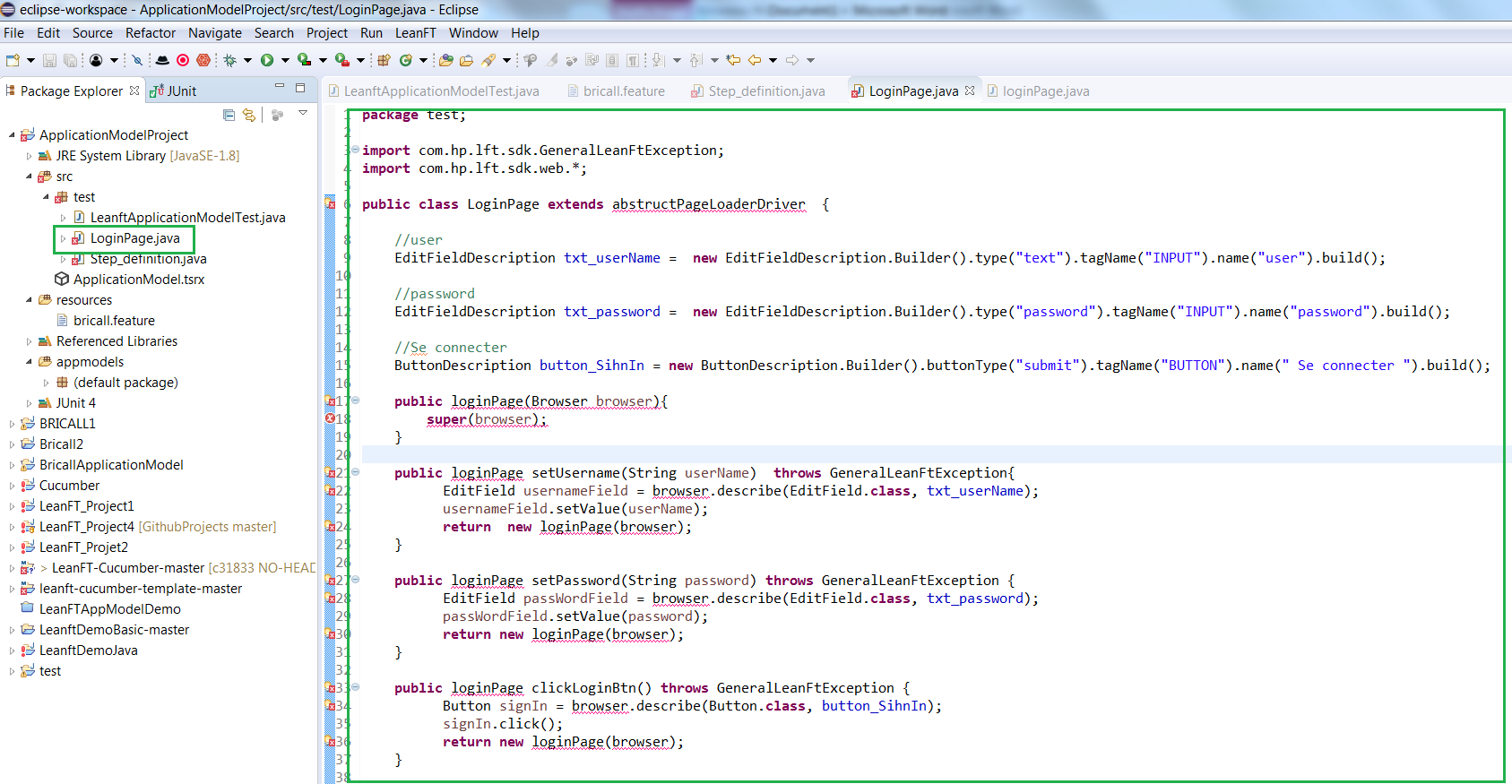


Créer les class des objets, chaque class représente une page de l’application « Bricall ».

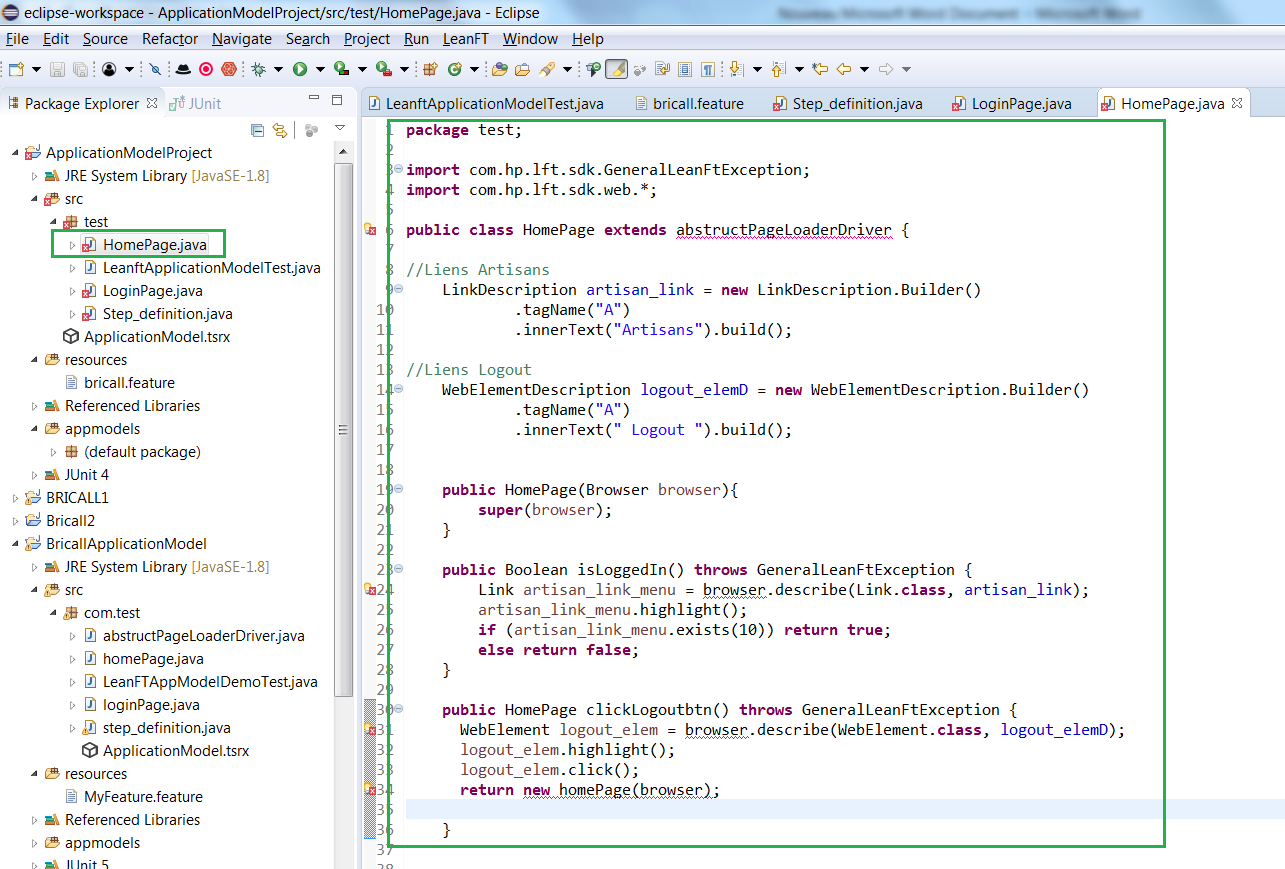
Pour le cas du scénario d’authentification, on va créer deux class :

1. **LoginPage** : qui représente la page d’authentification à Bricall :

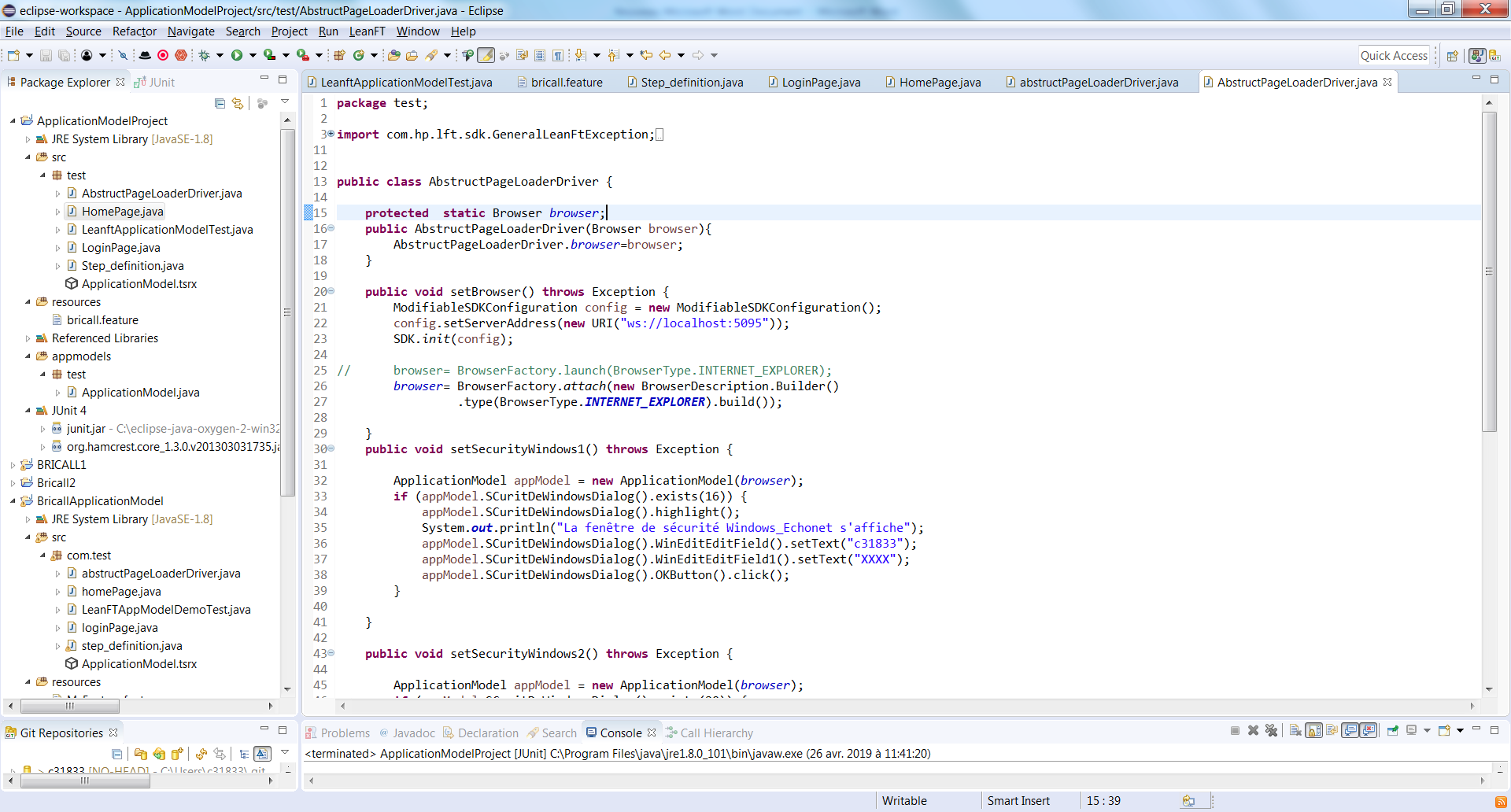




1. **HomePage** : qui représente la page afficher suite à l’authentification de l’administrateur :

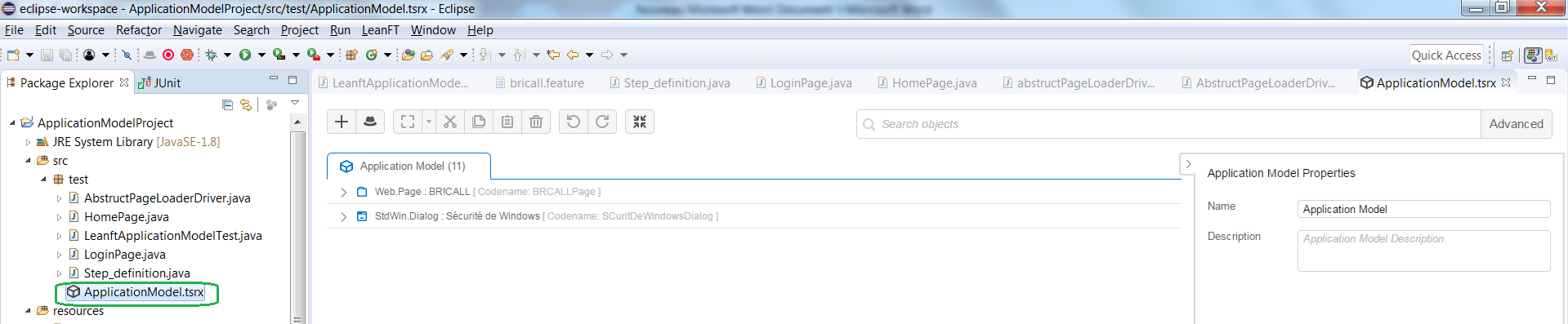


On va créer une troisième class « AbstructPageLoaderDriver » pour instancier l’objet « Browser » et configurer le SDK :

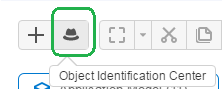


Utiliser ApplicationModel.tsrx :

ApplicationModel est le référentiel d'objets(Repository) pour LeanFT dont on peut stocker des objets et leurs descriptions de façon hiérarchique.



Double clic sur « ApplicationModel.tsrx », puis sur l’icône OIC (Object Identification Center)

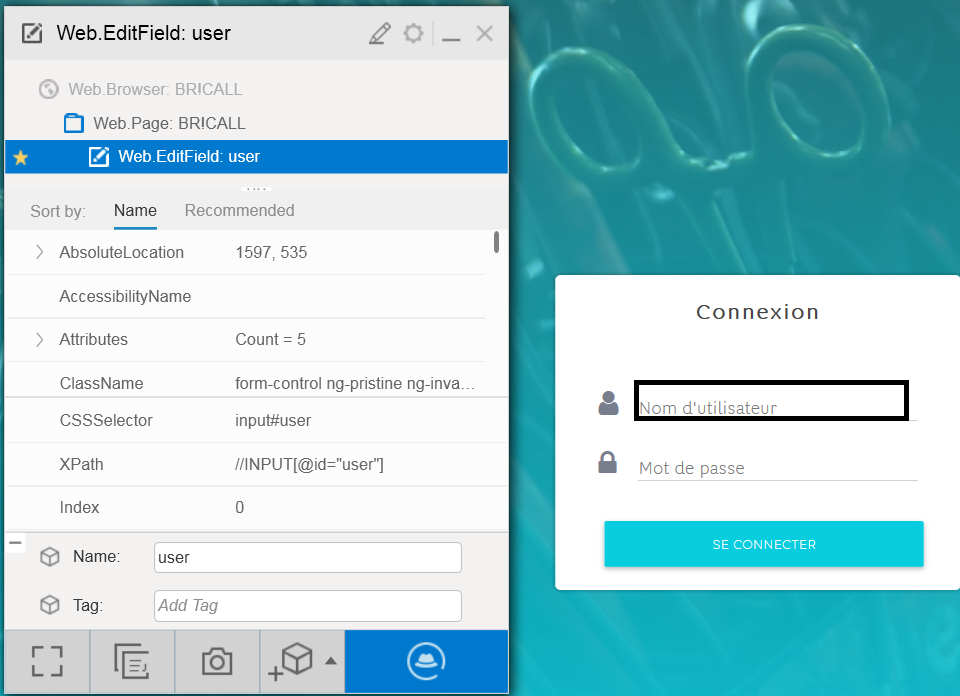


L’OIC s’affiche :

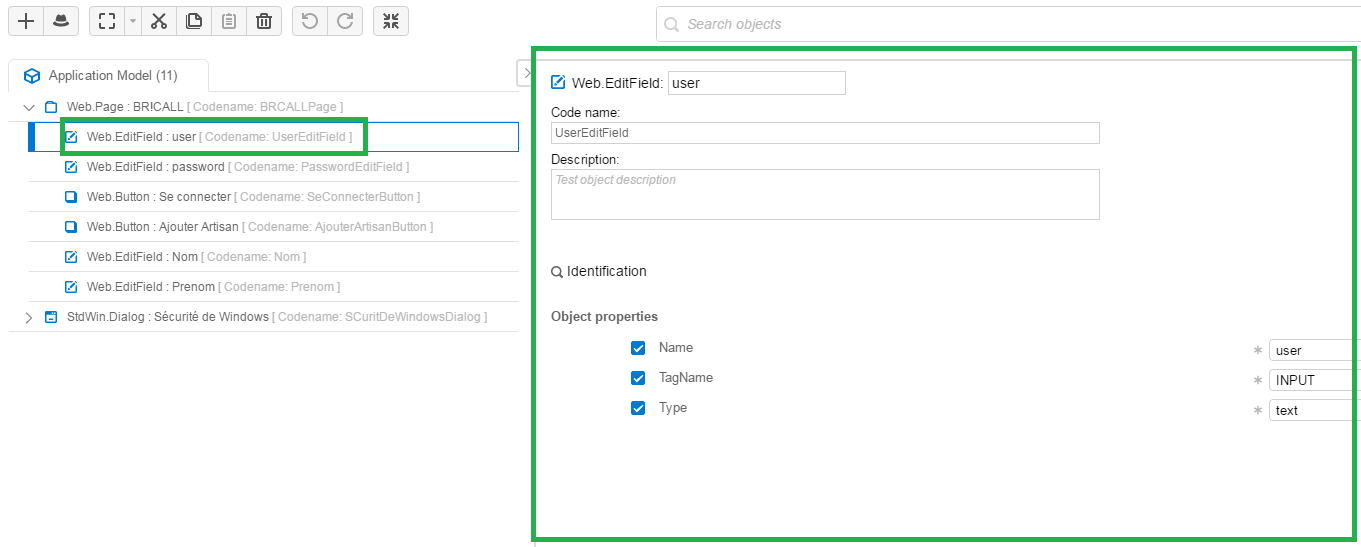


Cliquer sur le bouton « Spy ».

Puis cliquer sur l’objet à capturer.



Cliquer sur le bouton « Add object to ApplicationModel »  est l’objet « User » s’ajoute dans le « ApplicationModel.tsrx ».



Ensuite on peut manipuler cet objet dans notre code :

ApplicationModel appModel = **new** ApplicationModel(*browser*);

appModel.BRCALLPage().UserEditField().setValue("XXXX");