

	LP MMS
	Persistance des données

Consignes sur l'utilisation de Git et Github

Durant votre travail, à chaque apparition d'une indication de type

// GIT \ fix ...

commitez et pushez sur Github votre projet mis à jour avec comme message de commit le message fourni après le symbole // GIT \.

Partie 1 – Structure d'un projet Spring Boot / Écriture de classes du domaine

Nous allons étudier les différentes étapes du développement d'une application Spring Boot. Ces étapes étant dépendantes, il est indispensable que chaque partie soit entièrement réalisée avant de passer à la suivante.

Préliminaires

1. Créez un dossier de travail (« ws-persistence » par exemple) sur votre espace disque. Dans la suite du document l'expression « workspace » désignera ce dossier de travail.

Dans la suite du document, l'IDE désigne IntelliJ ou tout autre (Eclipse) permettant de développer des projets Java EE avec Spring Boot.

2. Lancez l'IDE. Si c'est la première fois que vous utilisez l'IDE IntelliJ, suivez les instructions de votre formateur.

1 – Création du projet *OurBusinessProject*

1. Depuis votre IDE, créez un nouveau projet de type « Spring Initializr » dans votre workspace.

Assurez vous que le SDK du projet SDK référence un JDK 1.8 ou supérieur.

Le nom du projet est « OurBusinessProject ». Complétez les propriétés Maven du projet en spécifiant comme nom d'artefact, nom de group et nom de package « ourbusinessproject ».

Sélectionnez les dépendances suivantes :

- Core > Validation
- Web > Rest Repositories
- SQL > JPA
- SQL > H2

Étudiez le fichier « pom.xml » et l'arborescence générés.

2. Votre IDE a créé de manière automatique la configuration de lancement permettant de lancer l'application OurBusinessProject.

Observez la configuration créée par l'IDE en ouvrant la fenêtre d'édition de configuration de lancement (Menu « Run » > « Edit Configurations... »).

Lancez l'application OurBusinessProject en utilisant la configuration créée par l'IDE et accédez à l'URL suivante à l'aide de votre navigateur :

<http://localhost:8080/>

// GIT \ fix #1.1 projet créé



Ce(tte) oeuvre est mise à disposition selon les termes de la [Licence Creative Commons Paternité - Pas d'Utilisation Commerciale 3.0 France](https://creativecommons.org/licenses/by-nc/3.0/fr/).

2 – Création de classes du domaine

L'objet de notre application est la gestion de projets proposés par des entreprises en direction d'étudiants issus de diverses formation. L'objet de cet exercice est la création des classes du domaine Project et Entreprise. La description des relations entre ces entités ne fait pas l'objet de cet exercice, elle sera étudiée dans 2ème partie.

1. Depuis votre IDE, créez la classe du domaine «ourbusinessproject.Project » (il s'agit de créer une classe Java « standard »).
2. Créez la classe de test associée à la classe du domaine «Project» en spécifiant :
 - Junit4 comme *Testing library (IntelliJ)*,
 - ProjectTest comme *Class name*,
 - ourbusinessproject comme *Destination package*.

La classe doit-être générée dans le dossier « test/java/ourbusinessproject».

Modifiez le fichier ProjectTest.java pour qu'il soit identique au contenu disponible ici : <https://gist.github.com/FranckSilvestre/4250191799372e513d22a6e361ec289a>

3. Lancez les tests de la classe ProjectTest depuis IntelliJ (cliquez droit sur la classe de tests et lancez « Run »). Lancez les tests depuis la commande Maven 'test'. Comparez les deux comportements et justifiez les.
4. Modifiez la classe Project de telle sorte que les tests passent. Vous pouvez vous aider de l'aide en ligne suivante : <http://docs.oracle.com/javase/6/tutorial/doc/gircz.html>
// GIT \ fix 1.2.4 classe Project
5. Créez la classe du domaine «ourbusinessproject.Enterprise» en suivant la même démarche que pour la classe Project. Le contenu de la classe EnterpriseTest se trouve ici : <https://gist.github.com/FranckSilvestre/458b129a0b0e3818d3aaef4034e537af>
// GIT \ fix 1.2.5 classe Enterprise

3. Persistance des données

1. Complétez les classes Project et Entreprise de telle sorte que les objets instances de ces classes puissent être sauvegardés en base de donnée relationnelle. Les annotations JPA devront-être utilisées pour décrire la correspondance objet-relationnelle.
// GIT \ fix 1.3.1 persistance Project et Entreprise
2. Créez la classe de service EnterpriseProjectService et les classes de tests associées EnterpriseProjectServiceTest et EnterpriseProjectServiceIntegrationTest.
3. Modifiez les contenus des classes de tests pour que leur contenu correspondent au code disponible en ligne : <https://gist.github.com/FranckSilvestre/51db101f54a2a4791b0862aac32f84cb>
<https://gist.github.com/FranckSilvestre/e11a4af2f3c30547ec142dc9769952ad>
4. Complétez votre projet afin que l'ensemble des tests passent.
// GIT \ fix 1.3.4 classe EnterpriseProjectService