

## Atelier 1

**Matière :** Spring Boot

**Classe :** GLSI 2

**Enseignant :** Mohamed Anouar DAHDEH

### Préparation de l'espace de travail

Vous aurez besoin des outils suivants :

- **Java Development Kit (JDK) :** installez la version de JDK appropriée pour votre environnement de développement.
- **Environnement de développement intégré (IDE) :** choisissez un IDE tel que Spring Tool Suite (STS) pour développer votre application Spring Boot. Configurez l'IDE avec les plug-ins nécessaires au développement de Spring Boot.

### Exercice 1 : Spring Initializr

**Objectif :** Créer un projet First Spring Boot à l'aide d'Initializr et modifier la page d'index pour afficher un message personnalisé.

1. Visitez le site **Web start.spring.io**.
2. Fournissez les détails suivants pour générer le projet :
  - **Métadonnées du projet :**
    - Group : com.example
    - Artifact : firstspringbootProject
    - Description : Un projet simple de Spring Boot
    - Nom du paquet : com.example.firstspringbootproject
  - **Projet :**
    - Sélectionnez le projet Maven.
  - **Spring Boot :**
    - Sélectionnez la dernière version stable de Spring Boot.
  - **Dépendances :**
    - Sélectionnez « Web » comme seule dépendance.
3. Cliquez sur le bouton « **Générer** » pour télécharger le projet sous forme de fichier ZIP.
4. Extrayez le fichier ZIP téléchargé à l'emplacement de votre choix sur votre ordinateur.
5. Ouvrez votre (IDE) tel que Visual Studio Code.

6. Importez le projet extrait dans votre IDE en tant que **projet Maven**.
7. Faites un clic droit sur le fichier **pom.xml** et sélectionnez « **Add Starters** ».
8. Recherchez « spring-boot-devtools » et sélectionnez-le dans la liste des suggestions.
9. Enregistrez les modifications dans le fichier **pom.xml**.
10. Dans la structure du projet, recherchez le dossier **src/main/resources/static**.
11. Créez un fichier nommé **index.html** dans le dossier **static**.
12. Ajoutez la structure HTML nécessaire au fichier **index.html**.
13. À l'intérieur de la balise <head>, créez une balise <link> pour lier un fichier CSS externe. Définissez l'attribut href sur **styles.css** (que nous créerons à l'étape suivante) et l'attribut rel sur « stylesheet ».
14. Dans la balise <body>, ajoutez un élément <h1> et incluez le message « Hello from my first Spring Boot Application ».
15. Enregistrez le fichier index.html.
16. Dans le dossier **src/main/resources/static**, créez un nouveau fichier nommé **styles.css**.
17. Ouvrez le fichier styles.css dans un éditeur de texte.
18. Ajoutez le code CSS suivant pour définir la couleur de l'élément <h1> sur bleu :

```
h1 {
    color: blue;
}
```

19. Enregistrez le fichier styles.css.
20. Exécutez l'application Spring Boot.
21. Accédez à l'application en ouvrant un navigateur Web et en saisissant l'URL appropriée (généralement http://localhost:8080).
22. Vérifiez que la page d'index affiche le message « **Hello from my first Spring Boot Application** » en bleu.

### Exercice 2 : Utilisation de STS dans Visual Studio

**Objectif :** Affichage d'un message personnalisé de couleur rouge à l'aide de STS dans Visual Studio Code, modification de l'emplacement du fichier CSS et changement du port du serveur à l'aide du fichier **application.properties**.

1. Ouvrez Visual Studio Code.
2. Lancez la palette de commandes en appuyant sur Ctrl+Maj+P (Windows/Linux).
3. Tapez « **>Spring Initializr : Create a Maven Project** » dans la palette de commandes et sélectionnez-le.
4. Fournissez les détails nécessaires au projet :

- Group : com.example
  - Artifact : secondspringbootproject
5. Pour les dépendances, sélectionnez « **Web** ».
  6. Cliquez sur « **Terminer** » pour générer le projet.
  7. Dans la vue Explorateur, recherchez le dossier **src/main/resources/static**.
  8. Créez un nouveau dossier nommé « **css** » à l'intérieur du dossier static.
  9. Dans le dossier **src/main/resources/static/css**, créez un fichier nommé **styles.css**.
  10. Ouvrez le fichier **styles.css** dans un éditeur de texte.
  11. Ajoutez le code CSS suivant pour définir la couleur de l'élément **<h1>** sur bleu :

```
h1 {
    color: red;
}
```

12. Enregistrez le fichier **styles.css**.
13. Ouvrez le fichier **index.html** dans le dossier static.
14. Ajoutez la structure HTML nécessaire au fichier **index.html**.
15. À l'intérieur de la balise **<head>**, créez une balise **<link>** pour lier le fichier CSS.
16. Dans la balise **<body>**, ajoutez un élément **<h1>** et incluez le message « **Hello from my Second Spring Boot Application** ».
17. Enregistrez le fichier **index.html**.
18. Localisez le dossier **src/main/resources**.
19. Ouvrez le fichier **application.properties**.
20. Modifiez le numéro de port à la valeur souhaitée, par exemple : 8081
21. Exécutez l'application Spring Boot.
22. Accéder à l'application dans un navigateur web
23. Vérifiez que la page d'index affiche le message « **Hello from my Second Spring Boot Application** » en rouge.
24. Localisez le dossier **src/main/resources/static**.
25. Créez un nouveau dossier nommé « **images** » à l'intérieur du dossier statique.
26. Ajoutez le fichier image du logo Spring Boot (par exemple, « **spring-boot-logo.png** ») au dossier images.
27. Dans le fichier **index.html**, après l'élément **<h1>**, insérez une balise **<img>** pour afficher l'image du logo Spring Boot.
28. Enregistrez le fichier **index.html**.
29. Créez un nouveau dossier nommé « **js** » à l'intérieur du dossier statique.

30. Créez un nouveau fichier JavaScript nommé « **script.js** » dans le dossier **js**.
31. Ouvrez le fichier **script.js** et ajoutez le code JavaScript suivant pour afficher une alerte avec le message « **Hello !** » lors du chargement de la page :

```
window.onload = function() {
    alert("Hello!");
};
```

32. Dans le fichier **index.html**, juste avant la balise de fermeture **</body>**, ajoutez une **balise <script>** pour référencer le fichier **script.js**.
33. Exécutez à nouveau l'application Spring Boot et ouvrez le navigateur pour vérifier le résultat.