

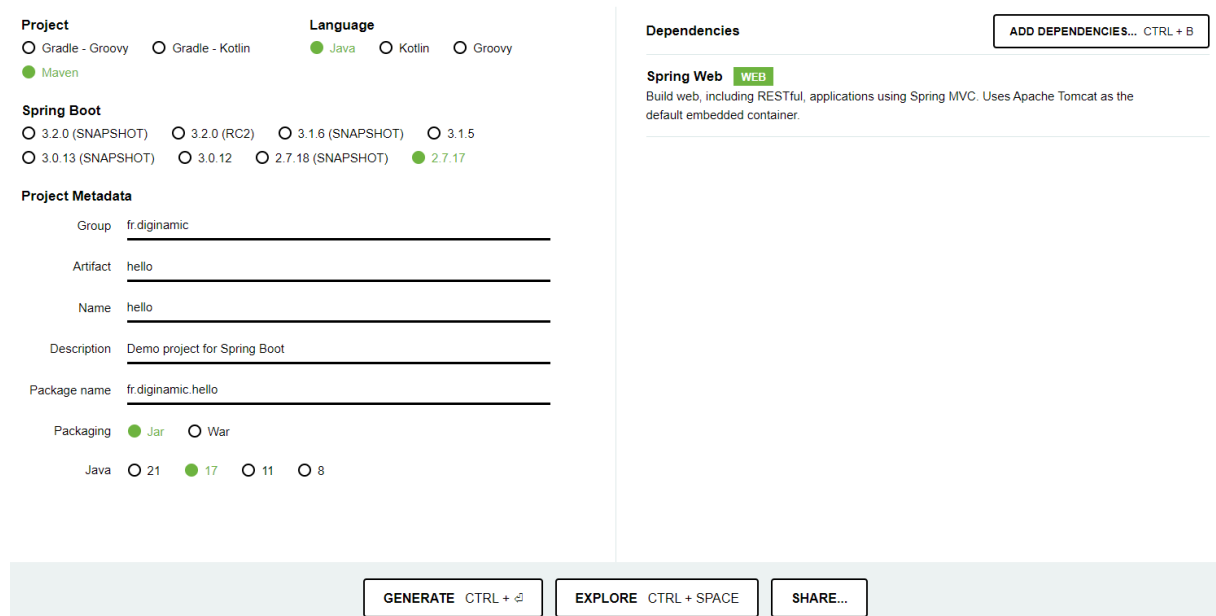
# SPRING BOOT – TP N°01

## OBJECTIF

- Initialiser et démarrer une application Spring Boot basique

## CREATION DU PROJET

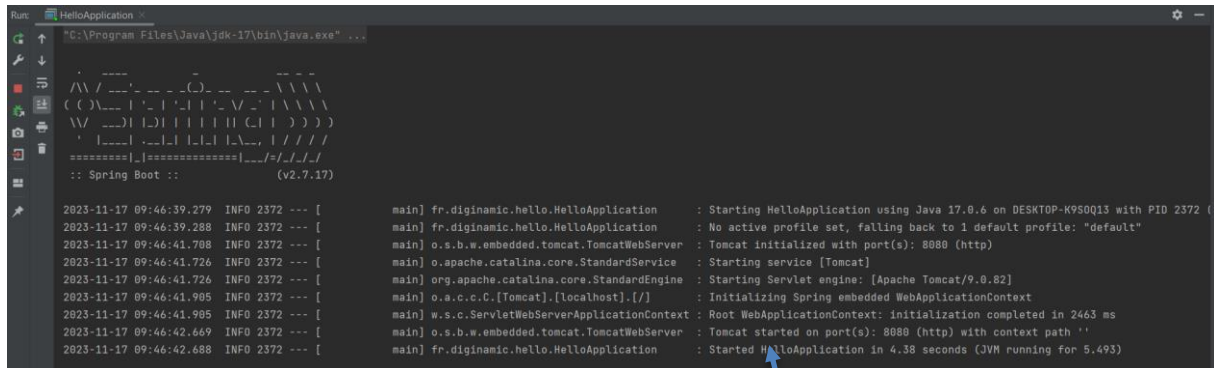
- Allez sur le site **Spring Initializr** : <https://start.spring.io/>
  - Sélectionnez **Maven**
  - Langage **Java** version **17**
  - Version 2.7.17 de Spring Boot
  - Cliquez sur le bouton **Add dependencies** et sélectionnez **Spring web**
  - **Group** : **fr.diginamic**
  - **Artifact** : **hello**



The screenshot shows the Spring Initializr web form. On the left, under 'Project', 'Maven' is selected. Under 'Language', 'Java' is selected. Under 'Spring Boot', version '2.7.17' is selected. The 'Project Metadata' section has 'Group' as 'fr.diginamic', 'Artifact' as 'hello', 'Name' as 'hello', 'Description' as 'Demo project for Spring Boot', and 'Package name' as 'fr.diginamic.hello'. Under 'Packaging', 'Jar' is selected. Under 'Java', version '17' is selected. On the right, under 'Dependencies', 'Spring Web' is selected. At the bottom, there are buttons for 'GENERATE', 'EXPLORE', and 'SHARE...'. The 'GENERATE' button has a tooltip that says 'CTRL + G'.

- Cliquez sur le bouton **Generate**.
  - Cette action génère un fichier zip et le download
- Allez dans votre répertoire de téléchargement.
  - Dézipper le fichier
  - Copiez le répertoire dans votre workspace IntelliJ (Par défaut : `C:/users/moncompte/IdeaProjects`)
- Dans IntelliJ créez un nouveau projet :
  - File >> New >> project from existing sources
  - Sélectionnez l'emplacement du projet (directory)
  - Sélectionnez **Import project from external model** et sélectionnez **Maven**

- Cliquez sur **Create**
- Attendez la fin complète du build qui peut prendre un peu de temps (1 ou 2 minutes) compte tenu des **downloads** de librairies à télécharger.
- L'application à exécuter se trouve dans le package fr.diginamic et s'appelle HelloApplication
- Démarrez cette application :



```

Run: HelloApplication
"C:\Program Files\Java\jdk-17\bin\java.exe" ...

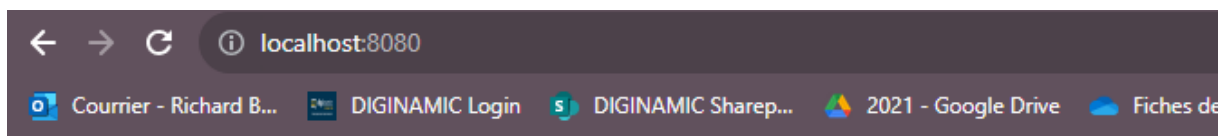
  ____  _
 / ___|| | | |
| |___| |_| |
 \___ \|  __/
      | |___
      |_|___|

:: Spring Boot ::
(v2.7.17)

2023-11-17 09:46:39.279 INFO 2372 --- [main] fr.diginamic.hello.HelloApplication : Starting HelloApplication using Java 17.0.6 on DESKTOP-K9SQ13 with PID 2372
2023-11-17 09:46:39.288 INFO 2372 --- [main] fr.diginamic.hello.HelloApplication : No active profile set, falling back to 1 default profile: "default"
2023-11-17 09:46:41.708 INFO 2372 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port(s): 8080 (http)
2023-11-17 09:46:41.726 INFO 2372 --- [main] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2023-11-17 09:46:41.726 INFO 2372 --- [main] org.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/9.0.82]
2023-11-17 09:46:41.905 INFO 2372 --- [main] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring embedded WebApplicationContext
2023-11-17 09:46:41.905 INFO 2372 --- [main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in 2463 ms
2023-11-17 09:46:42.669 INFO 2372 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 8080 (http) with context path ''
2023-11-17 09:46:42.688 INFO 2372 --- [main] fr.diginamic.hello.HelloApplication : Started HelloApplication in 4.38 seconds (JVM running for 5.493)
  
```

Un message en avant dernière ligne indique le démarrage de Tomcat sur le port 8080

- Ouvrez un navigateur web et allez sur l'URL suivante : <http://localhost:8080>



## Whitelabel Error Page

This application has no explicit mapping for /error, so you are seeing this as a fallback.

Fri Nov 17 09:39:04 CET 2023

There was an unexpected error (type=Not Found, status=404).

**Suite page suivante**

## CREATION D'UN CONTROLEUR

Un contrôleur est un type de classe particulière qui est « mappée » une URL donnée.

Par exemple je peux **créer une classe** qui sera **mappée** sur l'URL **/hello** et qui interceptera donc toutes les requêtes http arrivant sur cette URL (<http://localhost:8080/hello>).

Voici comment réaliser une telle classe :

- Commencer par créer un package **fr.diginamic.controleurs**
- Créer une classe appelée **HelloControleur**
- Afin d'indiquer à Spring Boot qu'il s'agit d'un controleur, ajoutez l'annotation

@RestController au dessus de la classe.

- Afin de mapper la classe sur l'URL /hello, ajoutez l'annotation

@RequestMapping("/hello") au dessus de la classe.

- Nous allons maintenant ajouter une méthode qui interceptera toutes les requêtes de type GET. L'annotation @GetMapping est ce qui permet ce mapping.

- Comme on le voit cette méthode retourne un simple message.

```
@RestController
@RequestMapping("/hello")
public class HelloControleur {

    @GetMapping
    public String direHello() {
        return "Hello";
    }
}
```

Démarrez maintenant votre application et vérifiez que <http://localhost:8080/hello> retourne bien le message hello.

Initialisez un dépôt Git et poussez-le sur GitHub.