

# Le Framework Spring

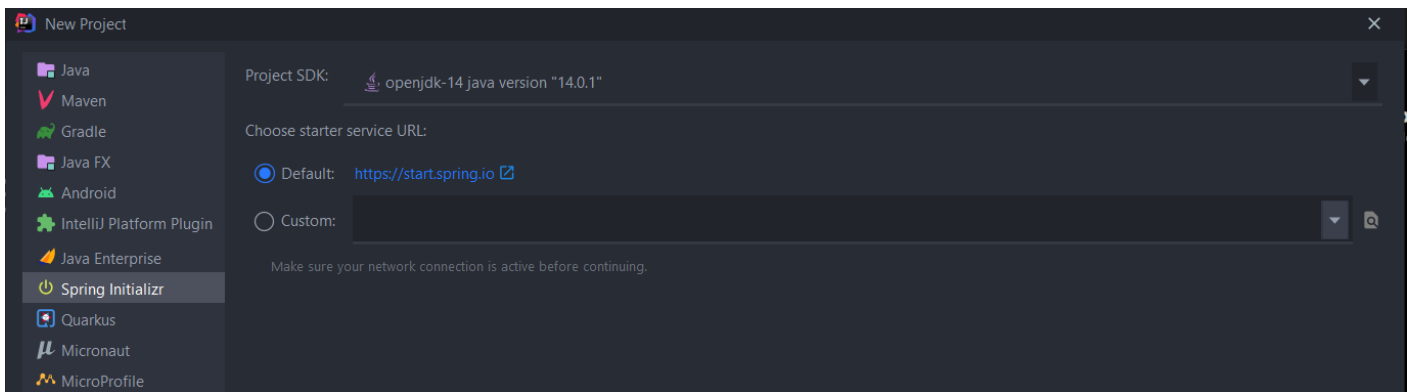
Spring est un Framework d'application Java open source qui est centré sur l'injection de dépendances. Il fournit également beaucoup de briques logicielles permettant de faciliter le développement d'application. On peut citer par exemple :

- Spring MVC pour les interactions Web
- Spring Data JPA pour se connecter à des bases de données
- Spring Security pour sécuriser les interactions avec le client
- Spring Boot pour l'autoconfiguration

Dans ce tutoriel nous allons apprendre à utiliser ces briques afin de développer des applications web exposant des services sous la forme d'API ReST.

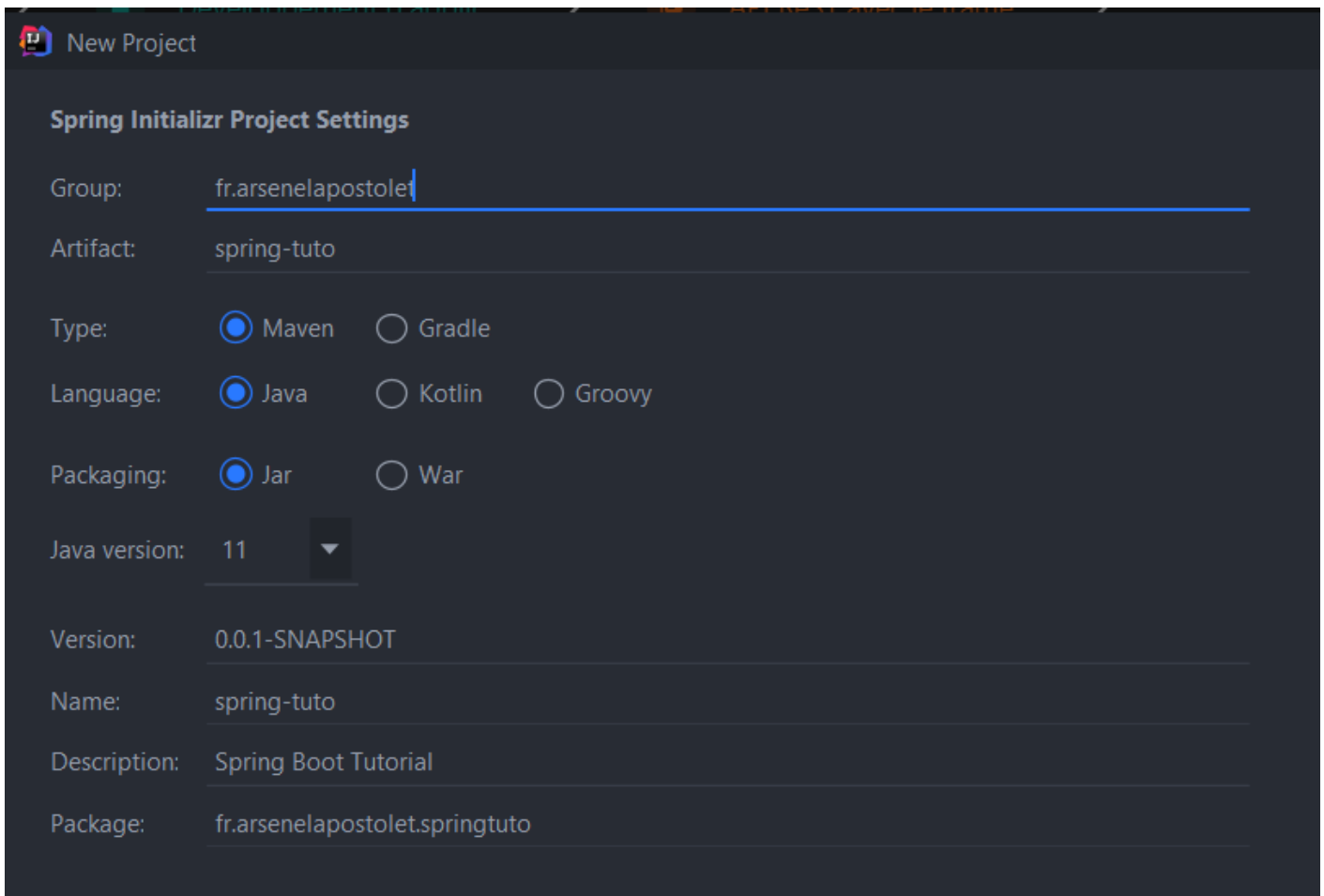
## Créer un projet

Pour créer un nouveau projet avec Spring Boot, choisissez dans IntelliJ le template de projet "Spring Initializer" :



“ Si vous êtes sur IntelliJ Community Edition, il faut installer le plugin Spring Initializer ou alors utiliser le site web.

Choisissez votre SDK Java et faites "Next". Il faut ensuite configurer votre artefact : son nom, son groupe, le langage et le build tool utilisé :



New Project

**Spring Initializr Project Settings**

Group: fr.arsenelapostolet

Artifact: spring-tuto

Type: ☒ Maven ☐ Gradle

Language: ☒ Java ☐ Kotlin ☐ Groovy

Packaging: ☒ Jar ☐ War

Java version: 11

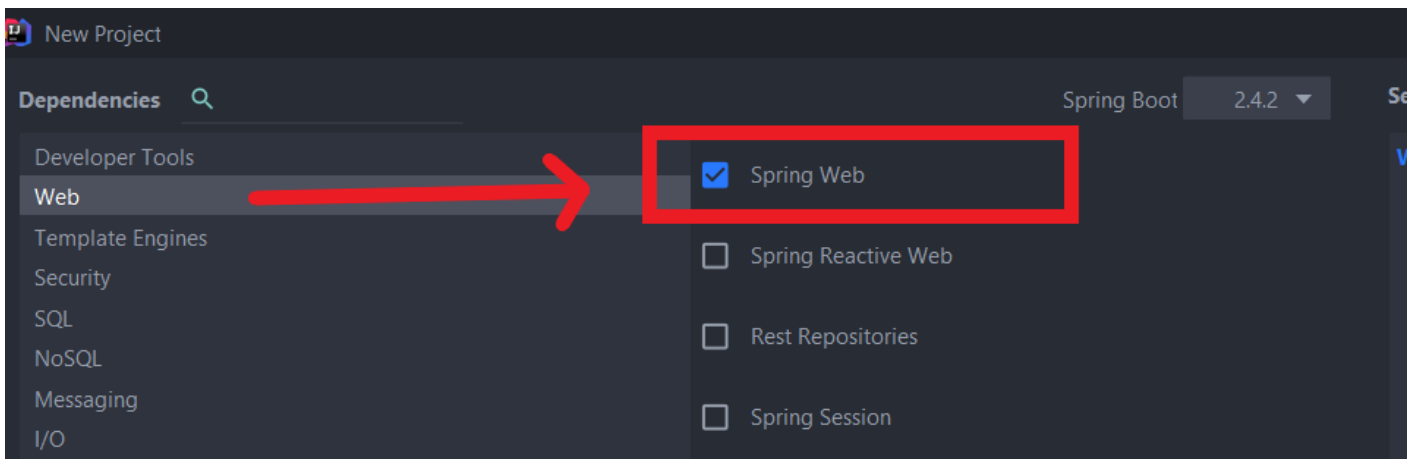
Version: 0.0.1-SNAPSHOT

Name: spring-tuto

Description: Spring Boot Tutorial

Package: fr.arsenelapostolet.springtuto

Nommez votre artéfact, puis votre groupe. Choisissez Java pour le langage et Maven pour le build tool. Une fois que c'est fait, cliquez sur "Next". Enfin, il faut choisir les dépendances de notre projet. Choisissez pour l'instant uniquement "Spring Web" dans la rubrique "Web" :



New Project

**Dependencies** 🔍

Spring Boot 2.4.2 ▼

Developer Tools

**Web** → ☒ Spring Web

Template Engines

Security

SQL

NoSQL

Messaging

I/O

☐ Spring Reactive Web

☐ Rest Repositories

☐ Spring Session

Voilà votre projet est créé !

# L'API

L'API que nous allons développer au fur et à mesure de ce cours est une API simple de gestion d'une Todo liste. Elle va se baser sur la ressource Todo suivante :

```
public class Todo {  
  
    private String id;  
    private String title;  
    private String description;  
  
    ... getters & setters ...  
}
```

---

Revision #6

Created 24 January 2021 19:59:39 by Arsène Lapostolet

Updated 28 September 2022 12:08:00 by Arsène Lapostolet