

# Installation

## Angular

1. **Téléchargez Node.js** depuis [nodejs.org](https://nodejs.org) (choisissez la version **LTS** pour une utilisation stable).
2. **Exécutez l'installateur** (.msi) et suivez les instructions par défaut.
3. **Vérifiez l'installation** dans un terminal (CMD/PowerShell) :
  - `node -v`
  - `npm -v`
4. **Installer Angular CLI (en global)** : Ouvrez un terminal et exécutez :
  - `-g @angular/cli@latest`
5. **Vérifier l'installation** :
  - `ng version`
6. **Créer un nouveau projet** :
  - `ng new mon-projet`
  - `cd mon-projet`
  - `ng serve` L'application sera disponible sur <http://localhost:4200>.

## Structure de projet :

mon-projet/

```
|— node_modules/      # Dependencies (géré par npm/yarn)
|— src/               # Code source de l'application
| |— app/             # Modules et composants principaux
| | |— app.component.ts  # Composant racine (logique)
| | |— app.component.html # Template HTML du composant
| | |— app.component.css  # Styles CSS du composant
| | |— app.component.spec.ts # Tests unitaires
| | |— app.module.ts      # Module principal (déclarations)
| |— assets/            # Images, polices, fichiers statiques
| |— environments/      # Configs dev/prod
| | |— environment.ts     # Config développement
| | |— environment.prod.ts # Config production
| |— index.html         # Page HTML principale (conteneur de l'app)
| |— main.ts            # Point d'entrée (bootstrap de l'app)
| |— styles.css          # Styles globaux
```

└─	favicon.ico	# Icône de l'application
──	.gitignore	# Fichiers ignorés par Git
──	angular.json	# Configuration CLI (build, serve, test)
──	package.json	# Dépendances et scripts npm
──	README.md	# Documentation du projet
──	tsconfig.json	# Configuration TypeScript

## Fichiers clés

### 1. Dossier src/

- Cœur de l'application : Contient tout le code source.

### 2. app/

- Composant racine (app.component.\*) :
  - .ts : Logique (TypeScript).
  - .html : Structure (bindings, directives).
  - .css : Styles scopés à ce composant.
  - .spec.ts : Tests unitaires (Jasmine).
- app.module.ts :
  - Déclare les composants/services utilisés.
  - Importe d'autres modules (ex: FormsModule).

### 3. assets/

- Stocke les ressources statiques (images, JSON, polices).

### 4. environments/

- environment.ts : Variables d'environnement (ex: apiUrl).
- environment.prod.ts : Surcharge pour la production.

### 5. Fichiers racine

- index.html :
  - Contient <app-root></app-root> (point d'injection de l'app).
  - Pas de <body> complexe (Angular génère le contenu dynamiquement).
- main.ts :
  - Lance l'application avec platformBrowserDynamic().bootstrapModule(AppModule).

- **styles.css :**
  - Styles globaux (appliqués à toute l'app).

## 6. Fichiers de configuration

- **angular.json :**
  - Configurations de build, serveur de dev, chemins des assets.
  - Exemple : "outputPath": "dist/" (dossier de build).
- **package.json :**
  - Liste des dépendances (@angular/core, rxjs, etc.).

## Springboot

### Via Spring Initializr (Méthode recommandée pour débiter)

Le moyen le plus simple pour générer un projet Spring Boot est d'utiliser [start.spring.io](https://start.spring.io) (Spring Initializr).

### Étapes :

1. **Rendez-vous sur** <https://start.spring.io>.
2. **Configurez votre projet :**
  - **Project** : Maven (ou Gradle si vous préférez).
  - **Language** : Java (par défaut), Kotlin ou Groovy.
  - **Spring Boot Version** : Choisissez la dernière version stable (ex: 3.2.0).
  - **Project Metadata** :
    - **Group** : com.example (votre organisation).
    - **Artifact** : demo (nom du projet).
  - **Dependencies** : Ajoutez celles dont vous avez besoin (ex: Spring Web, Spring Data JPA, Lombok, etc.).
3. **Cliquez sur "Generate"** pour télécharger un .zip du projet.
4. **Décompressez** et importez-le dans votre IDE (IntelliJ IDEA, Eclipse, VS Code).

The screenshot shows the Spring Initializr interface with the following configurations:

- Project:** ☒ Maven
- Language:** ☒ Java
- Spring Boot:** ☒ 3.5.0
- Project Metadata:**
  - Group: com.example
  - Artifact: projet1
  - Name: projet1
  - Description: Demo project for Spring Boot
  - Package name: com.example.projet1
  - Packaging: ☒ Jar
  - Java: ☐ 24 ☐ 21 ☒ 17
- Dependencies:**
  - Spring Web** (WEB): Build web, including RESTful, applications using Spring MVC. Uses Apache Tomcat as the default embedded container.
  - Spring Data JPA** (SQL): Persist data in SQL stores with Java Persistence API using Spring Data and Hibernate.
  - MySQL Driver** (SQL): MySQL JDBC driver.
  - Lombok** (DEVELOPER TOOLS): Java annotation library which helps to reduce boilerplate code.
  - Spring Boot DevTools** (DEVELOPER TOOLS): Provides fast application restarts, LiveReload, and configurations for enhanced development experience.
  - Spring Security** (SECURITY): Highly customizable authentication and access-control framework for Spring applications.

Buttons at the bottom: GENERATE (CTRL + d), EXPLORE (CTRL + SPACE), and a menu icon (three dots).

## Options Essentielles à Sélectionner (via [start.spring.io](https://start.spring.io))

Catégorie	Options à cocher	Pourquoi ?
<b>Project</b>	Maven Project (plus simple pour les débutants)	Structure standard avec pom.xml
<b>Language</b>	Java	Langage principal de Spring Boot
<b>Spring Boot</b>	Version 3.2.x (LTS)	Stable et bien documentée
<b>Packaging</b>	Jar	Plus facile à déployer
<b>Java</b>	17 ou 21 ou 24	Compatible avec Spring Boot 3

**Dépendances Clés** (à ajouter dans Spring Initializr) :

### 1. Spring Web :

- Pour créer des API REST (@RestController, @GetMapping).
- 2. **Spring Data JPA :**
  - Gère la base de données et les requêtes SQL via Hibernate.
- 3. **MySQL Driver :**
  - Connexion à votre base MySQL (XAMPP).
- 4. **Lombok** (Optionnel mais recommandé) :
  - Simplifie le code (ex: @Data pour les getters/setters automatiques).
- 5. **Spring Boot DevTools :**
  - Recharge automatique pendant le développement.

## Structure de projet

### mon-projet-spring/

```

|— src/
|   |— main/
|   |   |— java/
|   |   |   |— com/example/monprojet/ # Package racine
|   |   |   |   |— MonProjetApplication.java # Classe principale
|   |   |   |   |— controllers/    # Couche contrôleur (API REST)
|   |   |   |   |— models/        # Entités (JPA/Hibernate)
|   |   |   |   |— repositories/   # Interfaces Spring Data JPA
|   |   |   |   |— services/      # Logique métier
|   |   |   |— resources/
|   |   |   |   |— static/          # JS/CSS/images
|   |   |   |   |— templates/      # Fichiers Thymeleaf/HTML
|   |   |   |   |— application.properties # Config principale
|   |   |   |   |— application.yml    # Alternative (YAML)
|   |   |— test/                    # Tests unitaires/intégration
|— target/                          # Fichiers compilés (généré)
|— pom.xml                          # Dépendances Maven
  
```

## └─ README.md

### Explications détaillées

#### 1. Fichiers racine

- pom.xml (ou build.gradle pour Gradle) :
  - Définit les dépendances (ex: Spring Boot Starter, JPA, Lombok).
  - Exemple de dépendance :

#### 2. Dossier src/main/java

- MonProjetApplication.java :
  - Classe principale avec @SpringBootApplication.
- controllers/ (ex: UserController.java) :
  - Gère les requêtes HTTP (annotations @RestController, @GetMapping).
- models/ (ex: User.java) :
  - Entités JPA (@Entity, @Id).
- repositories/ (ex: UserRepository.java) :
  - Interface Spring Data JPA (extends JpaRepository).
- services/ (ex: UserService.java) :
  - Logique métier (@Service).

#### 3. Dossier src/main/resources

- application.properties (ou application.yml) :
  - Configuration de l'app (port, base de données, etc.).
- static/ :
  - Contient les fichiers statiques (ex: style.css, app.js).
- templates/ :
  - Fichiers HTML pour les vues (si vous utilisez Thymeleaf).

#### 4. Dossier src/test

- Contient les tests (ex: MonProjetApplicationTests.java).

## Eclipse

1. Allez sur le site officiel : <https://www.eclipse.org/downloads/>
2. Choisissez votre version :
  - Pour Java/Spring Boot, téléchargez "**Eclipse IDE for Enterprise Java and Developers**" (contient outils Java EE, Maven, etc.).
3. Sélectionnez votre OS (Windows, macOS, Linux).
4. Exécutez le fichier téléchargé (.exe ou .zip).
  - Si .zip, extrayez-le dans un dossier (ex: C:\eclipse).
5. Lancez eclipse.exe (créez un raccourci si besoin).
6. Choisissez un workspace (dossier où seront vos projets).

### Configuration initiale

#### À la première ouverture :

Sélectionnez un **workspace** (dossier par défaut pour les projets).

**Installez Spring Tools (STS)** si besoin :

Allez dans **Help** → **Eclipse Marketplace**.

Cherchez "**Spring Tools**" et installez-les.