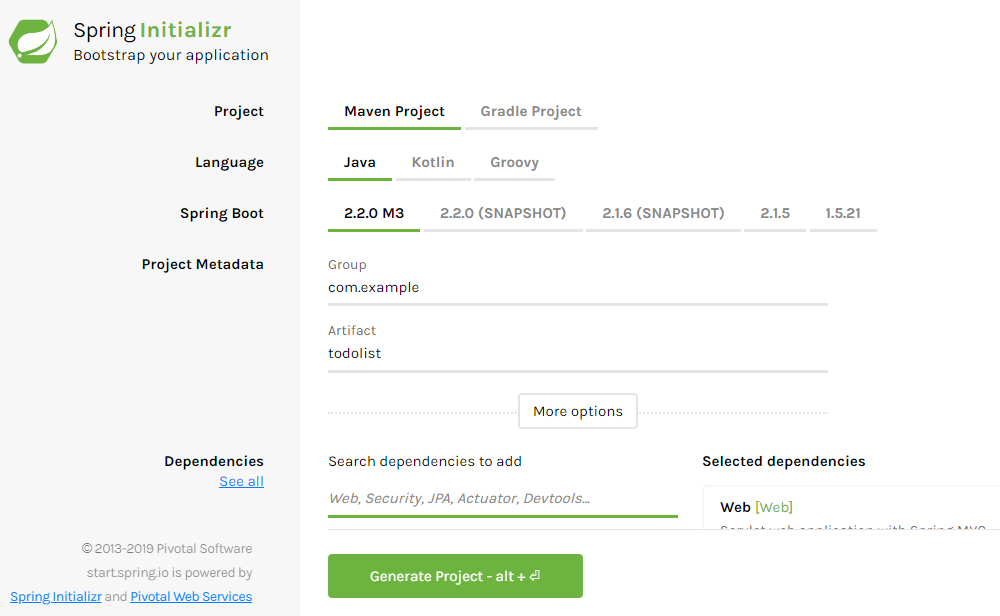
## KHỞI TẠO PROJECT

### Tạo project tại Spring Initializr

***Bước 1***: Mở trình duyệt Browser và nhập start.spring.io



***Bước 2***: Nhập đầy đủ các thông tin như hình ảnh bên dưới:

Project: Maven Project.

Language: Java.

Spring Boot: Chọn Spring Boot version mà bạn muốn phát triển, nhưng để chọn phiên bản ổn định ta không nên chọn SNAPSHOT.

Project Metadata: Nhập group cho project. Ví dụ: com.example

Artifact: cũng là tên đồ án: Ví dụ: todolist

Dependencies: Ta sẽ chọn một vài dependency cơ bản cho project: Web, JPA, Thymeleaf, Devtools, MySQL.

***Bước 3***: Nhấn nút Generate Project hoặc bấm tổ hợp phím Alt + Enter. Ứng dụng web sẽ tự động tải về một bản zip project mà ta vừa thiết lập.

***Bước 4***: Mở ứng dụng Spring Tool Suite.

***Bước 5***: Unzip project ta vừa tải về.

***Bước 6***: Ta sẽ import ứng dụng và STS. Nhấn File > Open Projects from File System… > Directory > Chọn Folder project ta vừa unzip > Chọn Finish.

Sau đây là những dependency sau khi khởi tạo project trong file **pom.xml**

<dependencies>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-data-jpa</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-thymeleaf</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-web</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-devtools</artifactId>

<scope>runtime</scope>

</dependency>

<dependency>

<groupId>mysql</groupId>

<artifactId>mysql-connector-java</artifactId>

<scope>runtime</scope>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-test</artifactId>

<scope>test</scope>

<exclusions>

<exclusion>

<groupId>org.junit.vintage</groupId>

<artifactId>junit-vintage-engine</artifactId>

</exclusion>

<exclusion>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

</exclusion>

</exclusions>

</dependency>

</dependencies>

Tuy nhiên, chúng ta vẫn cần thêm một vài dependency nâng cao vào pom.xml:

* AOP
* Transaction Management
* JPA API
* JPA Implementation - Default Hibernate
* JDBC

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-aop</artifactId>

<scope>compile</scope>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-jdbc</artifactId>

<scope>compile</scope>

</dependency>

<dependency>

<groupId>org.hibernate</groupId>

<artifactId>hibernate-core</artifactId>

<scope>compile</scope>

</dependency>

<dependency>

<groupId>javax.transaction</groupId>

<artifactId>javax.transaction-api</artifactId>

<scope>compile</scope>

</dependency>

<dependency>

<groupId>org.springframework.data</groupId>

<artifactId>spring-data-jpa</artifactId>

<scope>compile</scope>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-aspects</artifactId>

<scope>compile</scope>

</dependency>

### Tích hợp với Cơ sở dữ liệu

Ở ví dụ này, chúng ta sẽ sử dụng Hệ quản trị cơ sở dữ liệu **MySQL** và chúng ta sẽ cần lấy Database URL, username và password.

Mở file **src/main/resources/application.properties** trong project của ta và điền những thuộc tính như sau:

#DATABASE

spring.datasource.url=jdbc:mysql://localhost:3306/todolist

spring.datasource.username=<MySQL username của bạn ở đây>

spring.datasource.password=<MySQL password của bạn ở đây>

spring.datasource.driver-class-name=com.mysql.jdbc.Driver

spring.jpa.show-sql=true

# Hibernate ddl auto (create, create-drop, validate, update)

spring.jpa.hibernate.ddl-auto=update

spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL5Dialect

# Log cau lenh SQL

logging.level.org.hibernate.SQL=DEBUG

logging.level.org.hibernate.type=TRACE

Hãy điền vào MySQL **username** và **password** của bạn.

Trong MySQL, hãy tạo sẵn một Schema có tên: **todolist**. Và chúng ta sẽ phải tạo bất kỳ table nào. Các table sẽ tự động tạo bởi Hibernate là nhờ có thuộc tính: spring.jpa.hibernate.ddl-auto = update.

Ngoài ra, chúng ta còn thêm thuộc tính Log để debug các câu lệnh SQL sẽ chạy thông qua Hibernate.

### Tạo các Entity

Trước tiên, ta sẽ tạo package **com.example.todolist.Model** trong **src/main/java**

Trong package vừa tạo, ta thêm một Java class: **Todo.java**

package com.example.todolist.Model;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.Id;

@Entity

public class Todo {

@Id

@GeneratedValue

private Long id;

private String name;

private boolean isDone;

public Todo() {

super();

}

public Todo(Long id, String name, boolean isDone) {

super();

this.id = id;

this.name = name;

this.isDone = isDone;

}

public Long getId() {

return id;

}

public void setId(Long id) {

this.id = id;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public boolean isDone() {

return isDone;

}

public void setDone(boolean isDone) {

this.isDone = isDone;

}

}

Giải thích:

@Entity: Chỉ định đây là một Entity. Annotation này chỉ áp dụng cho các Entity class.

@Id: Chỉ định rằng đây là Khóa chính của Entity.

@GeneratedValue: Cung cấp tính năng cách thức tạo giá trị cho khóa chính.

@ManyToOne:

***! Chý ý:*** *Hibernate yêu cầu tất cả các Entity đều phải có hàm Contructor mặc định.*

Tương tự với các Entity khác: **TodoList.java** và **Folder.java**