# TỔNG QUAN VỀ SPRING

## Giới thiệu về Spring.



### Spring là gì?

* Spring là một Framework phát triển các ứng dụng Java được sử dụng bởi hàng triệu lập trình viên. Nó giúp tạo các ứng dụng có hiệu năng cao, dễ kiểm thử, sử dụng lại code…
* Spring nhẹ và trong suốt (nhẹ: kích thước nhỏ, version cơ bản chỉ khoảng 2MB; trong suốt: hoạt động một cách trong suốt với lập trình viên)
* Spring là một mã nguồn mở, được phát triển, chia sẻ và có cộng đồng người dùng rất lơn.
* Spring Framework được xây dựng dựa trên 2 nguyên tắc design chính là: **Dependency Injection** và **Aspect Oriented Programming**.
* Những tính năng core (cốt lõi) của Spring có thể được sử dụng để phát triển Java Desktop, ứng dụng mobile, Java Web. Mục tiêu chính của Spring là giúp phát triển các ứng dụng J2EE một cách dễ dàng hơn dựa trên mô hình sử dụng POJO (Plain Old Java Object)

### Ý nghĩa của “Spring”.

Thuật ngữ "Spring" mang nhiều ý nghĩa khác nhau trong các ngữ cảnh khác nhau. Nó có thể được dùng để chỉ chính dự án Spring Framework, nơi mọi thứ bắt đầu. Theo thời gian, các dự án Spring khác đã được xây dựng dựa trên Spring Framework. Thông thường, khi mọi người nói "Spring", họ đang ám chỉ toàn bộ họ dự án.

## Kiến trúc và Module của Spring.

### Dependency Injection và Inversion of Control.

* **Dependency Injection (DI)** là một design pattern quan trọng trong Spring Framework, giúp loại bỏ sự phụ thuộc lẫn nhau giữa các thành phần của ứng dụng. Điều này làm cho việc quản lý, mở rộng mã nguồn và kiểm tra ứng dụng trở nên dễ dàng hơn.
* **Inversion of Control (IoC)** Container trong Spring chịu trách nhiệm khởi tạo, cấu hình và kết nối các đối tượng. Spring cung cấp hai loại IoC containers: **BeanFactory** và **ApplicationContext**, giúp quản lý các đối tượng của bạn một cách thông minh.

### Các Module Quan Trọng của Spring Framework.

#### Core Container.

Core Container bao gồm các module: **spring-core**, **spring-beans**, **spring-context**, và **spring-expression** (Spring Expression Language).

* **spring-core** và **spring-beans** cung cấp các phần cơ bản của framework, bao gồm IoC và Dependency Injection. **Spring Bean được coi như linh hồn của ứng dụng Spring**.
* **spring-context** hỗ trợ đa ngôn ngữ (internationalization) và tích hợp các tính năng Java EE như EJB và JMX.
* **spring-expression** cung cấp hỗ trợ cho việc đặt và lấy giá trị, toán tử logic và truy cập các tập hợp dữ liệu.

#### Data Access/Integration.

Data Access/Integration bao gồm các module **JDBC**, **ORM**, **OXM**, **JMS** và **Transaction**.

* **spring-jdbc** giúp giảm thiểu mã JDBC và hỗ trợ nhiều cơ sở dữ liệu.
* **spring-orm** tích hợp với các orm API phổ biến như **JPA**, JDO và **Hibernate**.
* **spring-oxm** hỗ trợ Object/XML mapping với nhiều công cụ như AXB, Castor, XMLBeans và nhiều công cụ khác.
* **spring-jms** cung cấp khả năng tạo và sử dụng các message và tích hợp với spring-messaging.
* **spring-transaction** hỗ trợ quản lý giao dịch cho các POJO và các lớp đặc biệt.

#### Web.

Spring cung cấp Spring MVC để xây dựng ứng dụng web. Module bao gồm: **spring-web**, **spring-webmvc**, **spring-websocket** và **springwebmvc-portlet**.

* **spring-web** hỗ trợ tích hợp web, chức năng tải tệp và khởi tạo IoC container sử dụng servlet.
* **spring-webmvc** triển khai Model-View-Controller (MVC) cho ứng dụng web.
* **spring-websocket** hỗ trợ WebSocket-based, giao tiếp hai chiều giữa client và server trong các ứng dụng web.
* **springwebmvc-portlet** dành cho môi trường portlet và ánh xạ chức năng của module Web-Servlet.

#### AOP, Aspects, Instrumentation và Messaging.

Những module này hỗ trợ lập trình hướng khía cạnh (Aspect Oriented Programming) và tích hợp với AspectJ.

* **Module AOP** cung cấp lập trình hướng khía cạnh, cho phép tách rời mã và các chức năng của ứng dụng.
* **Module Aspects** tích hợp với AspectJ, một khuôn khổ AOP mạnh mẽ.
* **Module Instrumentation** hỗ trợ đo đạc và triển khai lớp bộ nạp.
* **Module Messaging** hỗ trợ STOMP và WebSocket, cung cấp mô hình lập trình chú thích cho định tuyến và xử lý tin nhắn STOMP.

## Một số dự án nổi bật của Spring.



### Spring MVC.

Spring MVC được thiết kế dành cho phát triển các ứng dụng web.

### Spring Security.

Cung cấp các cơ chế xác thực và phân quyền cho ứng dụng.

### Spring Boot.

Spring Boot là một framework giúp chúng ta phát triển cũng như chạy ứng dụng một cách nhanh chóng.

### Spring Data.

Cung cấp mô hình lập trình dựa trên Spring quen thuộc và nhất quán để truy cập dữ liệu trong khi vẫn giữ được các đặc điểm đặc biệt của kho dữ liệu cơ bản.

### Spring Batch.

Dự án giúp chúng ta dễ dàng tạo các lịch trình và tiến trình cho các công việc xử lý theo mẻ.

### Spring Social.

Dự án này sẽ kết nối ứng dụng của bạn với các API bên thứ ba của Facebook, Twitter, Linkedin, ... ví dụ như đăng nhập bằng google+ , facebook.

## Mô hình MVC.

### Định nghĩa.

MVC là viết tắt của ba thành phần chính trong kiến trúc này:

#### Model.

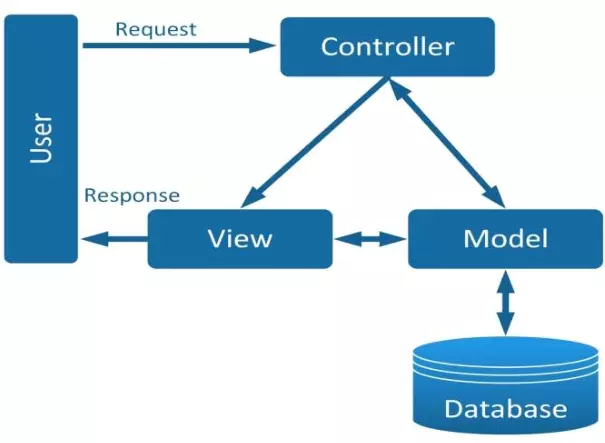
* Có nhiệm vụ thao tác với Database
* Nó chứa tất cả các hàm, các phương thức truy vấn trực tiếp với dữ liệu
* Controller sẽ thông qua các hàm, phương thức đó để lấy dữ liệu rồi gửi qua View

#### View

* Là giao diện người dùng (User Interface)
* Chứa các thành phần tương tác với người dùng như menu, button, image, text,...
* Nơi nhận dữ liệu từ Controller và hiển thị

#### Controller

* Là thành phần trung gian giữa Model và View
* Đảm nhận vai trò tiếp nhận yêu cầu từ người dùng, thông qua Model để lấy dữ liệu sau đó thông qua View để hiển thị cho người dùng



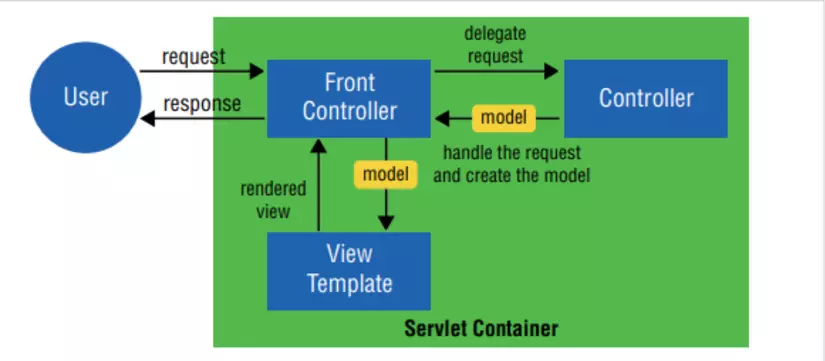
### Luồng xử lý trong MVC.

* Đầu tiên là **Request** từ người dùng được gửi từ client đến server
* Sau đó **Controller** dựa vào yêu cầu của người dùng tiến hành giao tiếp với **Model** để lấy **data từ database**
* Cuối cùng **Controller** gửi dữ liệu vừa lấy được về **View** và hiển thị ra cho người dùng trên trình duyệt

## Spring MVC.

Spring MVC (Model-View-Controller) là một phần quan trọng của Spring Framework và là một trong những framework phát triển ứng dụng web phổ biến nhất. Nó giúp bạn tách biệt logic trong ứng dụng web thành các phần riêng biệt, giúp quản lý và phát triển dễ dàng hơn.

**Flow trong Spring MVC:**



(VD: Tomcat)

* Bất kỳ request nào tới ứng dụng web đều sẽ được gửi tới Front Controller (Dispatcher Servlet)
* Front Controller sẽ sử dụng Handler Mapping để biết được controller nào sẽ xử lý request đó
* Controller nhận request, gọi tới các class service thích hợp để xử lý yêu cầu.
* Sau khi xử lý xong, Controller sẽ nhận được model từ tầng Service hoặc tầng DAO.
* Controller gửi model vừa nhận được tới Front Controller (Dispatcher Servlet)
* Dispatcher Servlet sẽ tìm các mẫu view, sử dụng view resolver và truyền model vào nó.
* View template, model, view page được build và gửi trả lại Front Controller
* Front Controller gửi một page view tới trình duyệt để hiển thị nó cho người dùng.

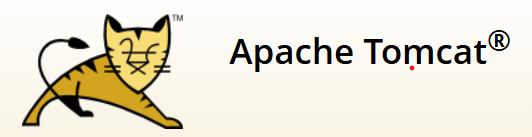
## Môi trường để phát triển ứng dụng Spring.

### JDK.



JDK, viết tắt của Java Development Kit, là một bộ phần mềm cung cấp môi trường phát triển cho ứng dụng Java. JDK bao gồm cả Java Runtime Environment (JRE), giúp lập trình viên có thể chạy thử và kiểm tra ứng dụng trong quá trình phát triển.

### Apache Tomcat.



Apache Tomcat là một máy chủ web HTTP được phát triển bởi Apache Software Foundation, dành cho việc phát triển ứng dụng Java Servlet, JavaServer Pages (JSP), Java EL và WebSocket.

### Maven



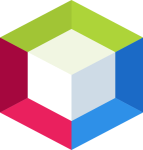
Apache Maven là một tool/công cụ/chương trình giúp quản lý mọi tác vụ xoay quanh việc viết code trong các dự án Java, bao gồm: tạo cấu trúc thư mục dự án, quản lý các thư viện đang sử dụng (được gọi là dependencies), biên dịch, kiểm thử, đóng gói sản phẩm,…

### IDE.

IDE (Integrated Development Environment) là môi trường phát triển tích hợp, giúp dễ dàng phát triển ứng dụng Spring.



IntelliJ IDEA



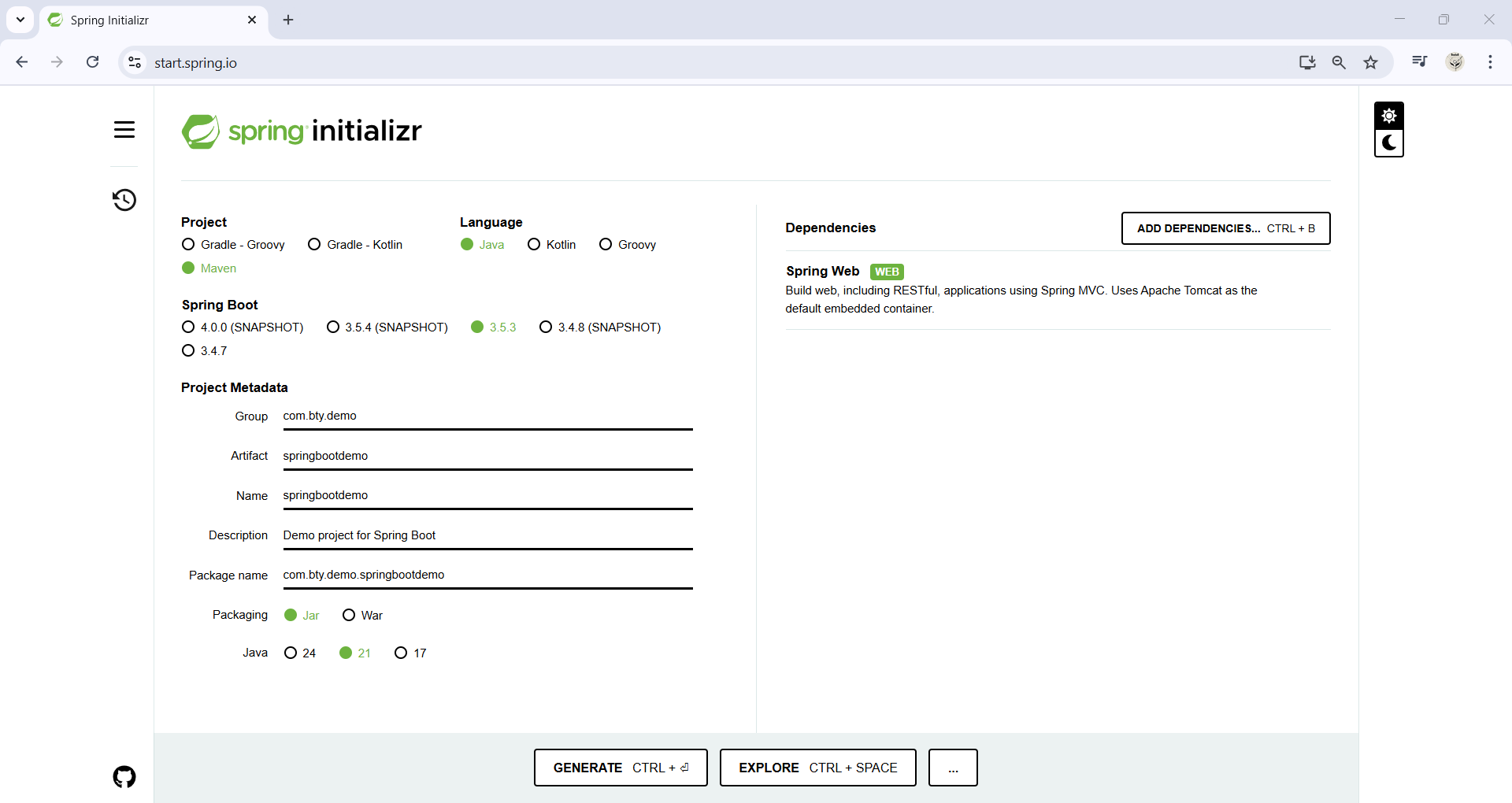
Apache NetBeans



Eclipse IDE

## Chạy chương trình Spring đầu tiên.

Khởi tạo dự án Spring Boot qua trang web: <https://start.spring.io/>



Trong Spring Boot, để chạy ứng dụng, cần chỉ cho Spring Boot biết nơi ứng dụng bắt đầu. Điều này được thực hiện bằng cách thêm annotation @SpringBootApplication trên class chính và gọi SpringApplication.run(App.class, args); để khởi tạo ứng dụng.



=> Kết quả sau khi chạy dự án Spring Boot:

