**Spring framework**  
Spring framework là 1 trong những framework phổ biến trong ứng dụng web.

MVC: là mô hình thiết kế sử dụng để tổ chức mã nguồn của 1 ứng dụng.

Các thành phần:

MODEL: đại diện cho lớp dữ liệu và các logic sử lý dữ liệu liên quan.  
VIEW: là thành phần chứa thông tin giao diện trả về cho người dùng.  
CONTROLLER: là lớp trung gian giúp lấy dữ liệu xử lý từ model và trả về view thích hợp.

Spring boot: là phần mở rộng của spring, giúp phát triển ứng dụng nhanh và dễ dàng hơn bởi nó cung cấp cơ chế cấu hình tự động giúp giảm thời gian cấu hình lại trong tất cả dự án.

Spring boot còn giúp đơn giản hoá việc quản lý sự phụ thuộc.

Spring boot cung cấp sẵn môi trường máy chủ tomcat hoặc jetty làm việc và đóng gói vào 1 file JAR hoặc WAR. Người dùng chỉ cần cài JRE (Máy ảo java) để chạy.

Spring boot starter cung cấp cơ chế tự động hoá việc quản lý sự phụ thuộc.

Nó sẽ tự động kéo toàn bộ sự phụ thuộc về để sử lý 1 nhiệm vụ hoặc tính năng nhất định. Đảm bảo các sự phụ thuộc tương thích với nhau.

**Ioc**

Ioc(Inversion of Control) là nguyên tắc thiết kế trong đó các sự phụ thuộc trong đối tượng được đưa vào từ môi trường bên ngoài thay vì chính đối tượng đó sẽ quyết định sự phụ thuộc bên trong.

**DI**

DI(dependency injection) là 1 kiểu của IoC

Nó cung cấp 3 cơ chế injection: contructor injection, setter injection, interface injection.

Spring cung cấp annotations @Autowired để tự động inject sự phụ thuộc vào đối tượng

**Bean**

Bean là các đối tượng do Ioc quản lý.

Cách khởi tạo Bean.

Bean được khởi tạo trong class có annotation Configuration hoặc Component chính là Bean.

Vòng đời: Tạo đối tượng -> tiêm sự phụ thuộc -> Gọi phương thức afterPropertiesSet, hoặc phương thức đánh dấu là PostConstruct -> Huỷ bean.

Bean scope:

Singleton: mặc định, Khởi tạo instance 1 lần duy nhất và được sử dụng lại ở nhiều nơi.

Prototype: Tạo 1 instance mới mỗi khi có yêu cầu.

Request: Mỗi http request tạo ra 1 instance mới

Application: ….

Session: Tạo mỗi khi có 1 session mới được tạo ra.

Annotation khai báo bean

Component

Repository

Service

Controller

Bean

Request: là yêu cầu từ phía client đến máy chủ để yêu cầu sửa lý 1 việc gì đó và nhận về kết quả.

Các thành phần của request: url, header, body, method

Các method thường gặp: get, post, put, patch, delete

Get: lấy dữ liệu

Post: Tạo dữ liệu

Put, Patch : Chỉnh sửa dữ liệu; put sửa toàn bộ bản ghi ; patch : Sửa 1 trường dữ liệu

Delete : Xoá dữ liệu

Response là kết quả trả về của 1 request sau khi xử lý xong.

Các thành phần của response : status, header, body

Status: 200 ok, 201 created, 400 bad request, 401 authorized, 403 permission, 404 not found, 407 conflig, 408: request time out, 500 server error,

Đầu mã lỗi:

1xx: Thông tin đang được xử lý

2xx: Thành công

3xx: Chuyển hướng

4xx: Lỗi từ người dùng

5xx: Lỗi phía server

Json: là định dạng dữ liệu nhẹ dễ đọc và dễ viết, dùng để lưu hoặc truyền dữ liệu giữa client và server hoặc ngược lại.

Json bắt đầu bằng {} bên trong gồm cặp key value.

Những định dạng value: mảng, json con, chuỗi, số, boolean, null

Api là tập hợp các định nghĩa và giao thức cho phép client và server giao tiếp với nhau.

Các loại API: REST API, GraphQL API, SOAP API

REST API:

* Chủ yếu sử dụng giao thức http
* Các yêu cầu tới server không phải phụ thuộc vào yêu cầu khác và các yêu cầu để xử lý phải có trong request đó
* Url được phân cấp rõ ràng
* Có cơ chế tự động cachin

Maven giúp tự động hoá quá trình xây dựng và quản lý, thư viện và plugin

Maven sử dungj file pom.xml để lưu trữ các thông tin trên.

Thành phần trong pom:

Dependencies: Nơi lưu trữ khai báo các thư viện bên ngoài

Plugins: cấu hình công cụ giúp thuận tiện trong việc phát triển

Postman:

Các thành phần trong postman:

url, method, header, param, body, authorization, response, cookie

tiện ích trong postman:

tạo thư mục

tạo request

tạo biến dùng chung,

viết scripts

lưu lại lịch sử request

chia sẻ folder postman