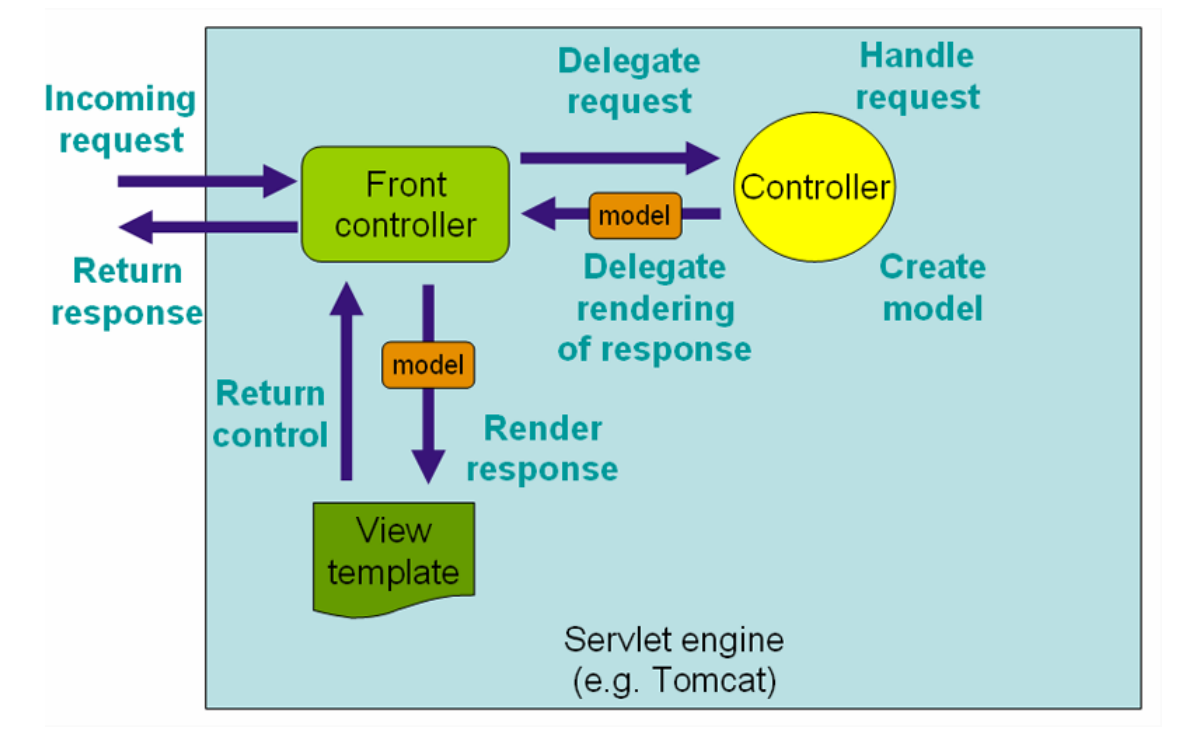
1. **Vẽ và trình bày cách Spring MVC xử lý request**

- Đầu tiên khi có request từ người dùng tới, front controller (ở đây là class dispatcher servlet) sẽ tiếp nhận toàn bộ các request đó

- Thông qua URL handle mapping mà front controller sẽ điều hướng request tới controller tương ứng

- Tầng model sẽ xử lý các nghiệp vụ tính toán logic và thông qua controller để trả về model and view cho front controller

- Lúc này thông qua view resolver để ánh xạ tên view đến view tương ứng

- Cuối cùng front controller sẽ chuyển view đó đến cho người dùng

**2 Trình bày cơ chế Dependency Injection trong Spring**

- DI – dependency injection là 1 cơ chế để triển khai IOC, được sử dụng để loại bỏ sự phụ thuộc giữa các mã của chương trình, giúp cho việc bảo trì, mở rộng chương trình dễ dàng hơn

- Các depency được cung cấp và điều khiển bởi framework hoặc container

**3 Có bao nhiêu cách để thực hiện Dependency Injection? Trình bày?**

- Có 3 cách để thực hiện DI: khác nhau ở chỗ là vị trí tiêm phụ thuộc. spring chỉ có 2 loại đầu, không có loại thứ 3

+ Constructor injection

+ Setter injection, có biến thể là field injection – là loại injection hay sử dụng phổ biến nhất trong spring boot

+ Interface injection: loại này không có trong spring

**4 Framework là gì ? Framework khác Library chỗ nào ?**

- Framework là các ứng dụng phần mềm có tính trừu tượng, cung cấp tính năng chung và thông dụng, có thể tùy biến để tạo nên những ứng dụng cụ thể khác nhau

- Mỗi framework cung cấp 1 phương pháp riêng biệt để xây dựng và triển khai ứng dụng

- Mỗi framework bao gồm một môi trường tổng thể, tái sử dụng được nhằm cung cấp các chứng năng và công cụ để hỗ trợ quá trình phát triển ứng dụng

**\* Khác biệt giữa Framework và Library**

Điểm khác biệt lớn nhất giữa Framework và Library đó chính là cơ chế **“Inversion of Control”**

• Với Library: Ứng dụng nắm quyền điều khiển (control)

• Với Framework: Framework nắm quyền điều khiển

Ngoài ra:

+Library: được tự do sd, k bị ràng buộc bởi cấu trúc hoặc quy tắc nào.

+FrameWork: phải tuân thủ theo luật chơi của Framework. Framework định hình cấu trúc của ứng dụng 🡪 **Lập trình viên phải điền mã vào ô trống mà framework cung cấp.**

**5 Spring Framework là gì ?**

- Là 1 framework dựa trên java

- Được sd để xây dựng các ứng dụng web

- Được cài đặt đầy đủ các đặc tính của MVC Pattern

\* Spring framework được là một [framework](https://nq.com.vn/framework-la-gi) phổ biến của nền tảng java cung cấp **đầy đủ tính năng** cho việc **phát triển và cấu hình** các hệ thống java

1. **Lợi ích của Spring Framework?**

- Spring là một framework giúp các nhà phát triển xây dựng những hệ thống và ứng dụng chạy trên JVM một cách **đơn giản, tiện gọn, nhanh chóng và mềm dẻo.**

- Framework giải quyết các vấn đề thông dụng, giúp lập trình viên tập trung vào **xử lý nghiệp vụ**

- Giúp **tăng tốc độ phát triển**

- Cung cấp môi trường **làm việc tiêu chuẩn**, giúp dễ **giao tiếp** giữa các bên khi cùng tham gia phát triển

- Các framework thường có cộng đồng phát triển lớn, các giải pháp đã được **đánh giá và thử nghiệm, hệ sinh thái** đầy đủ giúp nhanh chóng xây dựng được các **giải pháp tuỳ biến**

7 **Nguyên lý đảo ngược quyền điều khiển (Inversion of Control) là gì ?**

- IOC là 1 **nguyên lý** phát triển phần mềm, trong đó việc điều khiển các đối tượng hoặc các thành phần của hệ thống sẽ được thực hiện bởi **framework hoặc các container**

- IOC cho phép framework **nắm giữ quyền điều khiển** luồng thực thi của hệ thống và gọi các **mã nguồn khác**

**8 Bean là gì?**

- Bean là 1 đối tượng do Spring IOC container **quản lý, tự tạo, tự quản lý vòng đời.**

- Việc tạo bean sẽ được quyết định bằng cách đặt các annotation hoặc thông qua config

**9 Trong Spring có bao nhiêu Bean Scope?**

**Có 5 Bean Scope**

* Singleton:Chỉ duy nhất một thể hiện của bean sẽ được tạo cho mỗi container. Đây là scope mặc định cho spring bean. Khi sử dụng scope này cần chắc chắn rằng các bean không có các biến/thuộc tính được share.
* Prototype: Một thể hiện của bean sẽ được tạo cho mỗi lần được yêu cầu(request)
* Request: giống với prototype scope, tuy nhiên nó dùng cho ứng dụng web, một thể hiện của bean sẽ được tạo cho mỗi HTTP request.
* Session: Mỗi thể hiện của bean sẽ được tạo cho mỗi HTTP Session
* Global-Session: Được sử dụng để tạo global sesion bean cho các ứng dụng Portlet.

**10 @Autowired là gì?**

- @Autowired là 1 annotation của Spring, mục đích đánh dấu vị trí sẽ tiêm phụ thuộc (DI)

**11 @Component có ý nghĩa gì?**

- @Component được sử dụng để định nghĩa một khuôn mẫu chung cho toàn bộ dự án.

- Class được đánh dấu với annotation @Component thì sẽ được tạo thành 1 bean và sẽ được spring container quản lý, đồng thời các class này sẽ phải không thuộc tầng controller, service hay repository thì mới dùng @component được

**12 Trình bày ý nghĩa của Front Controller và Controller**

- Mục đích của Front controller là để handle tất cả request từ client tới server sau đó dựa vào URL handle mapping để điều hướng tới controller tương ứng và là nơi xử lý tất cả các yêu cầu 🡪 có thể dễ dàng xác thực, phân quyền, ghi log,…

- Ý nghĩa của Controller là: nhận ủy quyền xử lý các request từ phía front controller. Công việc cơ bản của nó là nhận dữ liệu cần thiết từ front controller, xác định view cần sử dụng, triệu gọi lớp xử lý nghiệp vụ cần thiết, sắp xếp dữ liệu mà view cần đến vào model, gửi thông tin về view cùng model lại cho front controller.

**13 Trình bày ý nghĩa của ModelAndView Interface, ModelAndView Class**

- ModelAndView Interface : Mục đích là để vừa trả về dữ liệu vừa trả về trang view, nếu 1 tham số thì chỉ trả về view, nếu 3 tham số sẽ vừa trả về view vừa trả về dữ liệu

- ModelAndView Class: có nhiệm vụ quan trọng trong việc tạo ra các hồi đáp (response). Các thể hiện của ModelAndView giúp dễ dàng tạo liên kết giữa data model và view. Trong đó data model là đối tượng chứa dữ liệu để hiển thị lên view

**14 Trình bày ý nghĩa của ModelMap**

- Class ModelMap là lớp con của LinkedHashMap được sử dụng trong việc xây dựng data model bằng sử dụng kiểu dữ liệu cặp gồm key và value. ModelMap sử dụng phương thức addAttribute() để tạo liên kết giữa model và tên logic trong model. Thuộc tính đặt trong ModelMap có thể sử dụng để hiển thị dữ liệu dạng form trong sự kiện submit form.

**15 Trình bày ý nghĩa của ViewResolver Interface**

- View resolver là cơ chế xử lý tầng view của spring mvc, dùng để ánh xạ tên view đến view tương ứng

**16 Phân biệt POST với PUT thường sử dụng để làm gì?**

- Trong web service, PUT dùng để update dữ liệu, trong khi POST dùng để thêm mới dữ liệu,

\* GET dùng để truy xuất dl, DELETE dùng để xóa dl

**17 @RequestMapping làm gì?**

Annotation RequestMapping dùng để ánh xạ một request tới một phương thức. Chúng ta có thể sử dụng @RequestMapping chú thích cho một phương thức hoặc một lớp. Một phương thức chú thích bởi annotation @RequestMapping sẽ trở thành phương thức xử lý request và sẽ được gọi khi dispatcher servlet nhận được một request với URL phù hợp.

**20 Trình bày cơ chế Data Binding trong Spring**

- Data binding là cơ chế liên kết dữ liệu đầu vào và đầu ra với model, giúp cho việc tương tác dl dễ dàng hơn vì được liên kết tới các đối tượng biểu diễn dữ liệu

- Data Binding trong Spring hoạt động dựa trên DataBinder

Cơ chế Data Binding trong Spring: dữ liệu do người dùng nhập vào thông qua DataBinder trả về kết quả là Binding Result. DataBinder bao gồm các bộ xử lý Property editor( biến đổi các input thành các thuộc tính), bộ định dạng Formatter ( chuyển dối các input (string) thành các kiểu dữ liệu mong muốn), bộ lọc dữ liệu Validator( dùng để validate dữ liệu) và bộ chuyển đổi ConversionService

**21 So sánh RequestParam và PathVariable ?**

- Giống nhau: đều thực hiện chung 1 nhiệm vụ là lấy tham số từ người dùng gửi lên server

- Khác nhau: RequestParam có thể lấy dl từ form submit còn PathVariable thì chỉ lấy dl trên URL ở dạng value

**22 Thymeleaf là gì?**

- Thymeleaf là 1 bộ xử lý view được sử dụng cho các ứng dụng web và các ứng dụng độc lập

- Thymeleaf được xây dựng phù hợp với các tiêu chuẩn web, đặc biệt là HTML5

- Thymeleaf cho phép xử lý 6 loại templeate: HTML, XML, Javascript, TEXT, CSS, RAW

**23 Sử dụng lặp trong Thymeleaf như thế nào? Sử dụng điều kiện trong Thymeleaf như thế nào?**

- Vòng lặp trong Thymeleaf, ta sử dụng cú pháp => th:each = “object, iter :${list}”

- Đối tượng iter trong vòng lặp th: each cung cấp các thuộc tính sau:

+ Index: trả về chỉ số lần lặp hiện tại, bắt đầu từ 0

+ Count: trả về chỉ số lần lặp hiện tại, bắt đầu từ 1

+ Size: trả về tổng số phần tử trong vòng lặp

+ Current: tương ứng với object của lần lặp hiện tại

+ Even/Odd: vòng lặp chẵn hoặc lẽ (boolean)

+ first/last: trả về true nếu vòng lặp đầu tiên/ cuối cùng

- Sử dụng điều kiện trong thymeleaf: cú pháp th:if “${conditions}”. Th if trả về true trong các tường hợp sau:

- Một giá trị Boolean là true

- Một số có giá trị # 0

- Một ký từ có giá trị # ‘0’

- Một chuỗi có giá trị # “false”. “no”, “off”

- Một giá trị không phải là Boolean, số, ký tự hoặc chuỗi

**24 Formatter là gì ? Converter là gì ?**

- Formatter: Chỉ hỗ trợ chuyển dữ liệu từ client sang Server (tầng web), giúp chuyển dữ liệu từ String (Client) sang kiểu dữ liệu của mình chọn (Server)

- Converter: Hỗ trợ chuyển đổi 2 kiểu dữ liệu bất kỳ, dùng ở đâu cũng được (dùng ở bất cứ tầng nào của ứng dụng) và hay dùng ở back-end

**25 ORM là gì ?**

**- ORM (object-relational mapping):** Là 1 kĩ thuật liên kết các đối tượng trong lập trình với các đối tượng trong database

- ORM giúp cho phép truy xuất dễ dàng đến dữ liệu thông qua đối tượng trong lập trình

- ORM giúp lập trình viên tập trung thao tác với các đối tượng mà không cần quan tâm đến database thực tế đang dùng

**26 JPA là gì? Spring Data JPA là gì ?**

- JPA: **java persistence API** cung cấp các đặc tả để duy trì, đọc, quản lý dữ liệu từ đối tượng java đến các quan hệ trong cơ sở dữ liệu. JPA cung cấp một mô hình POJO persistence cho phép ánh xạ các table sang các class hoặc ánh xạ các mối quan hệ giữa các table trong database sang các mối quan hệ giữa các đối tượng

**- Spring data JPA**: giúp cho việc sử dụng JPA trong ứng dụng Spring trở nên dễ dàng hơn, giúp cải tiến JPA tiêu chuẩn, đơn giản hóa tầng truy xuất, tự tạo ra các repository, tự tạo ra các câu lệnh truy vấn JPA thông qua tên method, ghi log, phân trang. Spring data JPA là 1 phần của dự án Spring data, có thể sd Spring data jpa với các framwork khác nhau như: Hibernate, OpenJPA,…

**27 Hibernate là gì ? Hibernate làm gì với database**

- Hibernate là 1 framework ORM

- JPA là tập hợp các interface được định nghĩa theo quy chuẩn thì hibernate là người triển khai ( implement) các interface này

- Hibernate cung cấp một cơ chế bộ nhớ đệm (caching) để giảm số lần truy cập vào cơ sở dữ liệu, giúp tăng hiệu suất cho ứng dụng của bạn. Khi một giao dịch được kích hoạt, Hibernate sẽ lưu trữ các đối tượng trong session. Khi một truy vấn được thực hiện liên tục, giá trị được lưu trữ trong session sẽ được sử dụng lại. Khi một giao dịch mới bắt đầu, dữ liệu sẽ được lấy lại từ cơ sở dữ liệu và được lưu trữ trong session. Điều này giúp giảm số lần truy cập vào cơ sở dữ liệu và tăng hiệu suất cho ứng dụng của bạn.

**28 Liệt kê một số annotation của hibernate?**

- @Entity: mô tả 1 class là Entity hoặc là 1 table

- @Table: Mô tả class này sẽ mapping với table trong database

- @Id: mô tả đây là primary key

- @Transient: Dung để mapping 1 kiểu dữ liệu không tạo cột trong database, mô tả column này không tồn tại trong database

- @Column: mô tả đây là 1 column, mô tả tên cột trong database, kiểu dữ liệu trong database

- @OnetoOne: quan hệ 1-1

- @Onetomany: mô ta quan hệ 1-N với 1 bảng khác

- @ManytoOne: mô tả quan hệ N-1 với 1 bảng khác

- @ManytoMany: mô tả quan hệ N-N với bảng khác

- @joinColumn: thường dùng trong mối quan hệ 1-N, N-1 với ý nghĩa foreign key của table có mối quan hệ 1

**29 Entity là gì?**

- Entity là đối tượng đại diện cho dữ liệu ở trong ứng dụng

- Entity thường là **POJO (plain old java object)**

- Entity sẽ được ánh xạ tới 1 bảng trong database

**30 Persistence Context & Entity Manager là gì?**

**- Persistence context là** tập các thể hiện của entity được quản lý, tồn tại trong 1 kho dữ liệu

**- Entity manager** là 1 interface trong JPA, cung cấp API để quản lý đối tượng và thực hiện các thao tác trên database như lưu, cập nhật, xóa dl

**31 Câu lệnh truy vấn động là gì? Câu lệnh truy vấn tĩnh là gì?**

- Câu lệnh **truy vấn động** là câu lệnh được tạo ra trong repository lúc lấy dl từ DB. Câu lệnh truy vấn động dễ sửa nhưng khi dùng sẽ chạy **chậm hơn** truy vấn tĩnh. Câu lệnh truy vấn động dùng phương thức **createQuery()** của lớp Entity Manger

- Câu lệnh **truy vấn tĩnh** là câu truy vấn trong ứng dụng, không thể thay đổi trong quá trình thực thi. Câu lệnh truy vấn tĩnh sẽ chạy **nhanh hơn** câu lệnh truy vấn động. Câu lệnh truy vấn tĩnh dùng phương thức **createNameQuery()** của lớp Entity Manger

**32 Trạng thái của Entity bao gồm những gì?**

Có 4 trạng thái gồm:

**- New:** object mới được tạo ra, chưa kết nối với database (hay còn gọi là transient – tạm thời)

**- Persistence:** Object đã được ánh xạ tới database để tạo bảng, lúc này object được quản lý bởi persistence context, lúc này mọi thay đổi của entity sẽ được hibernate ghi lại và đồng bộ xuống database

**- Detached** (tạm thời tách ra khỏi persistence context): Một khi persistence context tạm thời đóng lại, mọi entity liên kết trước đó sẽ bị tách ra, lúc này các thay đổi của entity sẽ không được đồng bộ với database

**-Remove:** Entity bị xóa khỏi cơ sở dữ liệu

**33 Có bao nhiêu loại mapping trong Hibernate?**

Có 4 loại Mapping trong Hibernate : One-to-One, One-to-Many, Many-to-One và Many-to-Many để ánh xạ các mối quan hệ giữa các bảng trong cơ sở dữ liệu quan hệ.

**34 Để sử dụng interface JpaRepository cần cung cấp những thông tin nào?**

Để sử dụng interface JpaRepository, bạn cần cung cấp hai thông tin: **kiểu của entity và kiểu của khóa chính** (primary key) của entity đó. Sau đó, Spring Data JPA sẽ tự động tạo ra một **implementation** cho interface này và bạn có thể sử dụng các phương thức được cung cấp bởi JpaRepository

VD: public interface BookRepository extends JpaRepository<Book, Long> {…}

35 Validation dữ liệu là gì?

- Là thực hiện đánh giá **xác minh tính hợp lệ của dữ liệu đầu** vào

**36 Trình bày cách triển khai validate dữ liệu trong Spring**

- Trong spring có 2 cách triển khai validate dữ liệu:

+ Cách 1: **Dùng annotation,** có 2 loại là annotation có sẵn và annotation tự định nghĩa. Cách này chỉ validate được dữ liệu đơn (là trường dữ liệu có annotation trên nó)

+ Cách 2: **Dùng custom validator** do mình tự xây dựng. Cách này có thể validate liên item hoặc những trường hợp đơn item mà mình muốn dùng validator

**37 Binding Result là gì?**

**- Binding result** là 1 interface của Spring MVC, extend Erros interface. Mục đich chính của nó là cho phép thực hiện validator, chứa kết quả các lỗi có thể có. Biding result phải được đặt sau model object cần validate

**38 AOP là gì? AOP gồm những yếu tố nào?**

**- AOP (Aspect oriented programing)** Lập trình hướng khía cạnh là 1 kỹ thuật lập trình nhằm phân tách chương trình thành các module riêng lẻ, phân biệt và không phụ thuộc vào nhau

- Khi hoạt động chương trình sẽ kết hợp các module để thực hiện các chức năng nhưng khi sửa đổi 1 chức năng thì chỉ cần sửa đổi 1 module

- AOP gồm các yếu tố:

**+ core concern:** logic xử lý chính của chương trình

**+ cross cutting concern**: là những xử lý phụ, cần được thực hiện của chương trình khi logic chính được xử lý

**39 Các loại Advice?**

Có 3 loại advice:

**+ Before advice:** sẽ chạy trước khi xử lý 1 method

**+ After advice:** chạy sau khi xử lý 1 method, trong after sẽ chia thành 3 loại là after returning, after throwing, after finally

**+ Around advice:**chạy cả trước và sau khi xử lý 1 method, phải gọi phương thức, tùy mình chọn việc aspect sẽ chạy như thế nào

**40 JoinPoint là gì?**

- JointPoint là 1 điểm trong chương trình có thể chèn xử lý phụ. Joinpoint có thể là 1 phương thức, 1 ngoại lệ, hoặc 1 trường dữ liệu bị thay đổi

**41 Nêu cách triển khai AOP mà bạn dùng trong dự án**

**42 Mục đích và cách xử lý ngoại lệ trong Spring Web MVC như thế nào?**

Mục đích của việc xử lý ngoại lệ trong Spring Web MVC là để đảm bảo rằng khi có lỗi xảy ra trong ứng dụng, người dùng sẽ nhận được **một thông báo lỗi thân thiện và rõ ràng** thay vì một trang lỗi mặc định do máy chủ tạo ra.

Spring Web MVC cung cấp nhiều cách để xử lý ngoại lệ, bao gồm:

- Sử dụng annotation **@ExceptionHandler** để định nghĩa một phương thức xử lý ngoại lệ trong controller hoặc trong một lớp được đánh dấu với annotation **@ControllerAdvice.**

- Sử dụng các implementation của interface **HandlerExceptionResolver** để định nghĩa **cách xử lý ngoại lệ cho toàn bộ ứng dụng**.

Việc xử lý ngoại lệ một cách hợp lý sẽ giúp cho người dùng có trải nghiệm tốt hơn và giúp cho nhà phát triển dễ dàng xác định và khắc phục các vấn đề trong ứng dụng.

**44 Cookie là gì?**

Cookie là các tệp được trang web người dùng truy cập tạo ra. Nó giúp trải nghiệm người dùng tốt hơn bằng cách lưu thông tin duyệt web. Với cookie, các trang web có thể duy trì trạng thái đăng nhập, ghi nhớ tùy chọn trang web và cung cấp nội dung phù hợp với người dùng 🡪 Cookie thường được sd để **luu thông tin của người** dùng trên trình duyện web

**45 Phân biệt Session và Cookie**

**- Giống nhau:** Đều được dùng để lưu trữ thông tin người dùng

**- Khác nhau**

+ **Session** là các tệp được lưu trữ ở **server** chứa thông tin người dùng còn **Cookie** cũng là các tệp lưu trữ nhưng được lưu trữ ở **client**

+ Một session kết thúc khi đóng trình duyệt, cookie kết thúc tùy thuộc vào mình cài đặt thời gian sống cho nó

+ Một session phụ thuộc vào cookie tức là nó chỉ hiện cookie trùng với session id, còn cookie không phụ thuộc vào session, nó sẽ truyền toàn bộ cookie có lưu trữ tại máy liên quan đến trang web

+ **Session bảo mật hơn vì lưu ở server**, cookie kém bảo mật do lưu ở client nên dễ sửa đổi

**46 Session là gì? cho ví dụ về session?**

- Là 1 phiên làm việc giũa **client và server**

- Session bắt đầu khi thực hiện request đầu tiên tới server và kết thúc khi cilent dừng làm việc vs server

- Session được **lưu trữ ở trên server**

- Ta có thể **truy cập giá trị** của biến session khi

+ User không đóng brower

+ User không log out

+ Session chưa bị hết hạn (Thường server sẽ có quy định thời gian hiệu lực của session)

\* Ví dụ: Session thường được sử dụng để lưu trữ các thông tin như thông tin đăng nhập, thông tin giỏ hàng, các lựa chọn của người dùng, vv.

**47 Nêu cách triển khai Session mà bạn dùng trong dự án**

**48 Web Service là gì? Lấy ví dụ về web service**

- Web service là các thành phần ứng dựng được hiển thị dưới dạng www.

Web service có thể sử dụng để tích hợp với các ứng dụng được của các ngôn ngữ khác nhau và chạy trên các nền tang khác nhau.

Web service h**oạt động như 1 server** trong mô hình ứng dụng client-server, dử dụng giao thức http/https và chỉ thực hiện 1 tác vụ cụ thể.

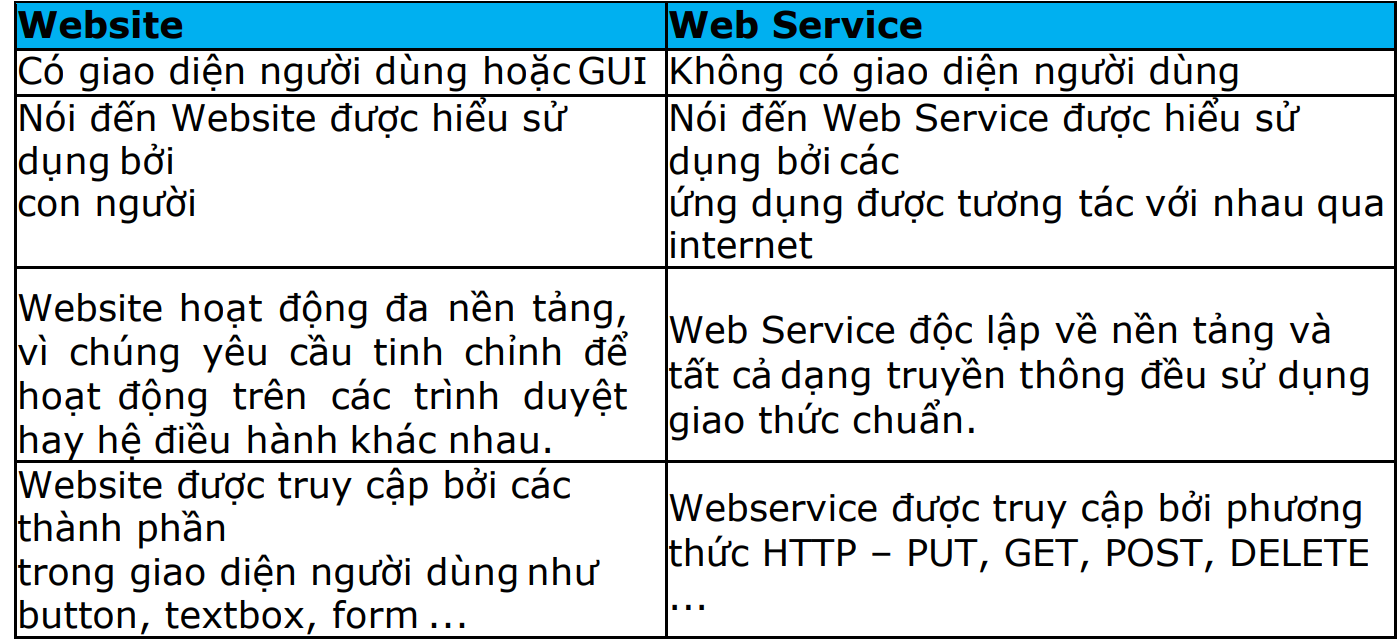
Web service sẽ xử lý và trả ra dữ liệu theo chuẩn, **đảm bảo mọi ứng dụng có thể hiểu và sử dụng** mà **không quan tâm** đến loại *thiết bị, hệ điều hành, kiến trúc phần mềm hay ngôn ngữ* được sử dụng

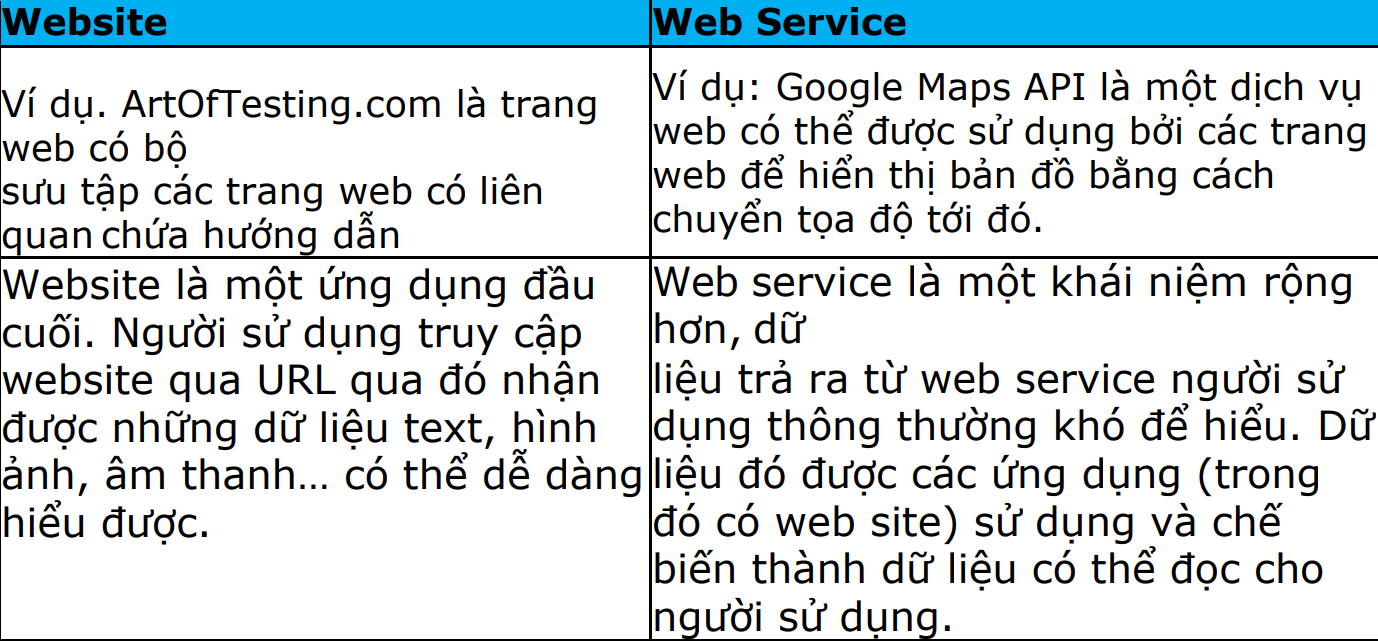
- Kiểu dl đầu ra phô biến của 1 web service thường là **XML và JSON**

- Ví dụ web service về giá vàng, giá cổ phiếu, giá xăng dầu, thời tiết,…

\* Ví dụ web service về dịch vụ thời tiết là dịch vụ này cung cấp thông tin thời tiết cho các thành phố khác nhau trên thế giới thông qua một API. Các ứng dụng khác có thể gửi yêu cầu đến API của dịch vụ thời tiết để lấy thông tin thời tiết cho một thành phố cụ thể. Dịch vụ thời tiết sẽ trả về thông tin thời tiết dưới dạng XML hoặc JSON và ứng dụng sẽ xử lý thông tin này để hiển thị cho người dùng.

**49 Phân biệt Web Service và Web truyền thống**





**50 SOAP là gì? RESTful là gì?**

**- SOAP (simple object access protocol)** là giao thức sử dụng XML để định nghĩa dữ liệu thuần văn bản và truyền dữ liệu thông qua HTTP

**- REST (Representational state transfer)** là 1 kiểu cấu trúc cung cấp các quy tắc để xây dựng web service. REST định nghĩa dữ liệu dưới dạng XML hoặc JSON và truyền thông qua HTTP. Các trang web xây dựng dựa trên REST còn được gọi là RESTful, chủ yếu xử lý các hành động CRUD

**51 jQuery là gì? Ajax là gì?**

- **Ajax (asynchoronous javascript and xml)** là 1 nhóm các công nghệ phát triển web được sử dụng để tạo ra các ứng dụng web **giàu tính tương tác, nhanh hơn, mượt hơn** với sự giúp đỡ của **XML, HTML, CSS, JS,…**

**- Ajax** giúp các trang web được cập nhật mà không cần tải lại trang

- Với Ajax, người dùng **có thể tiếp tục sử dụng** ứng dụng trong khi các chương trình trên client tạo request để lấy thông tin từ server

- Jquery là một thư việc javascript được thiết kế đơn giản hóa thao tác HTML DOM, xử lý sự kiện, CSS và Ajax. Nó có mã nguồn mở và miễn phí sử dụng

**52 Nêu ví dụ về cách bạn triển khai Ajax trong dự án của bạn**

**53 @RequestHeader và @ResponseHeader có ý nghĩa gì**

- @Request header được sử dụng khi ta muốn **lấy dữ liệu được truyền** bằng header của 1 request

- Chúng ta sử dụng @ResponseHeader khi muốn **trả về thêm dữ liệu cho client** ở phần trên cùng của mỗi response

**55 Bảo mật là gì? Cơ chế bảo mật trong Spring như thế nào?**

Bảo mật là quá trình đảm bảo tính bảo mật, toàn vẹn và sẵn sàng của dữ liệu và hệ thống thông tin. Bảo mật bao gồm việc ngăn chặn truy cập trái phép, sử dụng, tiết lộ, sửa đổi hoặc phá hủy thông tin.

\* Cơ chế bảo mật trong Spring

**- Xác thực(Authentication)**

**- Phânquyền (Authorization)**

**- Bộ lọc Servlet (Servlet Filters)**

**56 Authentication là gì ? Authorization là gì ?**

**- Authentication – xác thực** là 1 hành động nhằm **thiết lập hoặc chứng thực** 1 thông điệp hoặc đối tượng nào đó đáng tin cậy, xác thực cũng còn có ý nghĩa khác là công nhận nguồn gốc của đối tượng (nói 1 cách khác xác thực là **tiến trình thiết lập 1 principal**)

**- Authorization – phân quyền:** là tiến trình **quyết định xem 1 principal có được phép thực hiện 1 hành động** trong ứng dụng hay không. Principal có thể là 1 người, 1 thiết bị hoặc 1 hệ thống nào đó thực hiện 1 hành động trong ứng dụng. Trước khi phần quyền, principal phải được xác thực

**57 Liệt kê các cơ chế xác thực**

- **Http basic:** phương thức để client cung cấp user và password khi thực hiện yêu cầu

**- Sử dụng cookies:** lưu tại máy client nên dễ bị lộ

**- Tokens:** **JWT (Json web token)** là 1 tiêu chuẩn mở định nghĩa cách thức truyền tin an toàn giữa các thành viên 1 đối tượng json. Thông tin này có thể được xác thực và đánh dấu tin cậy dựa vào “chữ ký” của nó. Phần chữ ký của JWT được mã hóa

**-** **Signatures:** gồm 2 quá trình là tạo chữ ký số và chứng thực. Là mô hình sử dụng các kĩ thuật mật mã gắn với mỗi người sử dụng 1 cặp khóa công khai – bí mật và qua đó có thể ký các văn bản điện tử cũng như trao đổi các thông tin mật

**- One time password:** OTP mật khẩu chỉ sử dụng 1 lần

**- OAuth2 :** xác thực thông qua các ứng dụng

**58 CSRF là gì**

- CSRF: **cross site request forgery** là kỹ thuật tấn công bằng cách sử dụng quyền chứng thực của người dùng đối với 1 website nào đó

- CSRF là **kỹ thuật tấn công vào người dùng** và dựa vào đó hacker có thể thực thi những thao tác phải yêu cầu sự chứng thực

**59 CORS là gì**

- CORS: **cross origin resources sharing**: là cơ chế cho phép nhiều tài nguyên khác nhau của 1 trang web có thể được truy vấn từ domain khác với domain của trang web đó

**60 Spring Boot là gì?**

- Spring Boot là một Dự án nằm trên tầng **IO Execution** của **Spring IO Framework.**

- Spring Boot là **bước tiếp theo** của Spring để giúp Spring dễ dàng hơn trong việc thiết lập và phát triển các ứng dụng. Đối với Spring Boot, cấu hình Spring **được giảm thiểu**. Spring Boot hỗ trợ các containers nhúng trong việc cho phép các ứng dụng web có thể chạy **độc** **lập** mà không cần triển khai trên Máy chủ Web.

**61 Sự khác nhau giữa SpringBoot và SpringMVC là gì?**

- Spring Boot là **một phần mở rộng** của [Spring framework](https://hoidapcongnghe.info/spring-framework-la-gi/) cho phép các nhà phát triển **rút ngắn quy trình cần thiết** để xây dựng một ứng dụng web bằng cách sử dụng cấu hình tự động

- Spring MVC là **một thư viện** trong Spring framework. Trong khi Spring MVC là **một framework (khuôn khổ) thiết yếu** tồn tại trong Spring nói chung, thì Spring Boot là một mô-đun tùy chọn được sử dụng để **hợp lý hóa** quá trình phát triển, có thể bao gồm việc **tích hợp Spring MVC** trong quá trình xây dựng

**62 Cấu hình cho Spring Boot tìm kiếm các Bean ở nhiều package khác nhau bằng cách nào?**

- Thông qua các annotation @Controller, @Service, @Repository, @Component

**63 Trình bày Spring Security? @EnableWebSecurity làm gì?**

- Spring security là một dự án bảo mật trong hệ sinh thái spring, **cung cấp dịch vụ** bảo mật toàn diễn cho các **ứng dụng doanh nghiệp** có nền tảng Java EE

- Spring security là 1 framewwork tập trung vào việc **cung cấp khả năng xác thực** và **phân quyền** cho ứng dụng java

Annotation **@EnableWebSecurity** được sử dụng để **kích hoạt cấu hình bảo mật web** trong Spring Security. Bạn có thể thêm annotation này vào một lớp cấu hình @Configuration để định nghĩa cấu hình bảo mật cho ứng dụng của mình.