1. Trình bày cách Spring MVC xử lý request

* Request sẽ được gửi đến front controller (DispatcherServlet)
* Gửi request đến controller tương ứng bằng cơ chể handle mapping
* Controller sẽ gọi xuống Model lớn để xử lý request và gửi lại dữ liệu cho Controller
* Controller sẽ trả về tên vật lý view và model chứa dữ liệu về front controller
* Front controller dựa vào cơ chế view resolve để tìm view phù hợp và render dữ liệu lên đó rồi trả view lại cho người dùng

1. Trình bày cơ chế Dependency Injection trong Spring

* Là 1 design pattern cho phép xóa bỏ sự phụ thuộc
* Là một cơ chế triển khai IoC, trong đó thao tác tiêm (inject) vào thuộc tính các đối tượng khác được thực hiện bởi container và framework
  + Ví dụ: có 1 đối tượng động cơ và 1 đôi tượng xe. Và xe thì phụ thuộc vào động cơ. Do đó, theo dependence injection thì cần tiêm đối tượng động cơ vào chiếc xe, và động cơ sẽ như là 1 thuộc tính của xe

1. Có bao nhiêu cách để thực hiện Dependency Injection? Trình bày?

* Có 3 cách thực hiện DI:
  + Sử dụng @Autowirse interface
  + Sử dụng phương thức @Contructor và @Autowirse (không có cũng được)
  + Sử dụng phương thức setter và @Autowirse

1. Framework là gì? Framework khác Library chỗ nào?

* Framework là ứng dụng phần mềm có tính trừu tượng cao, cung cấp các tính năng chung thông dụng nhất sử dụng cho hệ thống.
* Giống: tập hợp các tính năng, các lớp viết sẵn để có thể tái sử dụng
* Điểm khác biệt so với Library là cơ chế IoC (Inversion of Control)

|  |  |
| --- | --- |
| Library | Framework |
| * Ứng dụng sẽ nắm quyền điều khiển (code của mình sẽ gọi code của thư viện) * Tập hợp các tính năng các lớp * Hoạt động bị động * Sử dụng trực tiếp mà ko cần thay đổi cấu trúc dự án * Không có cơ chế IoC | * Nắm quyền điều khiển ứng dụng (code của framework sẽ gọi code của ứng dụng) * Tập hợp các thư viện * Hoạt động chủ động * Phải thay đổi cấu trúc dự án(nhằm đáp ứng đúng theo nguyên tắc đề ra) * Có cơ chế IoC |

1. Spring Framework là gì?

* Spring Framework cung cấp một mô hình đầy đủ cho việc phát triển và cấu hình các hệ thông java lớn
* Các tính năng lõi:
  + Core
  + Testing
  + Data Access
  + Web Servlet
  + Web Reactive
  + Intergration
  + Language

1. Lợi ích của Spring Framework?

* Giải quyết các vấn đề thông dụng để lập trình viên tập trung xử lý nghiệp vụ
* Tăng tốc độ phát triển
* Cung cấp môi trường làm việc tiêu chuẩn, dễ giao tiếp giữa các bên khi cùng tham gia phát triển
* Có cộng đồng phát triển lớn, các giải pháp được đánh giá thử nghiệm, hệ sinh thái đầy đủ giúp nhanh chóng xây dựng các giải pháp tùy biến

1. Nguyên lý đảo ngược quyền điều khiển (Inversion of Control) là gì?

* “Nguyên lý đảo ngược điều khiển” là nguyên lý trong phát triển phần mềm. Nghĩa là việc điều khiển các đối tượng hoặc thành phần của hệ thống được được thực hiện bởi framework hoặc container (Lập trình viên không cần tạo đối tượng mà chỉ cần mô tả nó sẽ được tạo thế nào và framework sẽ lo phần còn lại ).

1. Bean là gì?

* Là các module chính của chương trình, là đối tượng nằm trong container được tạo ra và quản lý bởi Framework

1. Trong Spring có bao nhiêu Bean Scope?

* Có 5 scope được định nghĩa cho Spring Bean:
  + 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.
* Trong 5 scope trên thì 3 scope cuối dùng trong ứng dụng web

1. @Autowire là gì?

* @Autowired để báo cho Spring biết tự động tìm và inject bean phù hợp vào vị trí đặt annotation.

1. @Component có ý nghĩa gì?

* @Component: là một annotation của class. Nó dùng để đánh dấu class Java là một bean. (Khi nó không thuộc @Controller,@Service,@Repository). Một class Java được đánh dấu @Component được tìm thấy trong classpath. Spring Framework chọn nó và cấu hình trong ngữ cảnh ứng dụng như một Spring Bean.

1. Trình bày ý nghĩa của Controller

* Controller trong mô hình MVC là nơi nhận request, xử lý rq và là cầu nối của model và view
* Controller trong spring MVC chia thành front controller và controller
  + Front controller sẽ nhận request từ người dùng và sử dựng cơ chế Request Mapping để chuyển request về controller tương ứng. Sau khi nhận được data và tên vật lý của view thì dùng cơ chế view resolve để có thể tìm đến view phù hợp, render dữ liệu và phản hồi cho người dùng
  + Controller thì nhận request do f controller chuyển xuống và gọi Model (lớn). Nhận lại dữ liệu từ Model (lớn) và phản hồi lại tên view vật lý và dữ liệu cho front controller

1. Trình bày ý nghĩa của ModelAndView Interface

* Lớp ModelAndView là 1 class chứa Model và View trong framework MVC web. 2 lớp này riêng biệt, không giống nhau. giúp controller có thể return được cả model và view trong 1 giá trị duy nhất. View được giải quyết bởi đối tượng ViewResolver còn Model chính là dữ liệu được chứa trong Map.

1. Trình bày ý nghĩa của ModelMap

* ModelMap là con của linked hash map, có key và value, có các phương thức hỗ trợ để thao tác với collection tốt hơn

1. Trình bày ý nghĩa của ViewResolver Interface

* Là cơ chế ánh xạ tên vật lý của view sang đối tượng view tương ứng

1. Phân biệt POST với PUT thường sử dụng để làm gì?

POST: thêm mới dữ liệu

PUT: chỉnh sửa dữ liệu( toàn bộ)

1. @RequestMapping làm gì?

* Được dùng để ánh xạ các request tới các xử lý ở controller tương ứng

1. Thuộc tính consumes trong các Request Mapping là gì?

* Định dạng kiểu dữ liệu khi client request tới server

1. Thuộc tính produces trong các Request Mapping là gì?

* Định nghĩa kiểu dữ liệu trả về cho client

1. Trình bày cơ chế Data Binding trong Spring

* Dữ liệu người dùng nhập vào => databinder => trả về binding result và bean(input mapping dữ liệu vào bean và ngược lại.
* Các thành phần của databinder là:
  + Property editors: bộ định dạng dữ liệu
  + Formatters: bộ lọc dữ liệu
  + Validations: kiểm tra dữ liệu từ phía sever
  + Conversion service: bộ chuyển đổi

1. So sánh RequestParam và PathVariable?

* Giống: đều lấy thông tin của người dùng nhập vào

|  |  |
| --- | --- |
| Path variable | Request Parameter |
| Lấy dữ liệu theo dạng value(trích xuất dữ liệu từ URL path.) | Lây dữ liệu theo cặp key value (thường dùng trong form) (trích xuất dữ liệu từ request query) |
| Thường dùng với các chức năng như detail… | Thường dùng với các chức năng có submit theo form |

1. Thymeleaf là gì?

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

Thymeleaf cho phép hỗ trợ các template : HTML, XML, JS, CSS và Text...

Cung cấp 1 cách thức xây dụng template hiện đại và dễ bảo trì

Phù hợp với tiêu chuẩn của web đặc biệt hỗ trợ là HTML5

Thân thiện gần gũi dễ học vì sử dụng cú pháp HTML

Là thuộc tính của 1 thẻ HTML bắt đầu bằng th: >> dễ sử dụng

Ta có thể xem các file view mà ko cần phải chạy server.

1. Sử dụng lặp trong Thymeleaf như thế nào? Sử dụng điều kiện trong Thymeleaf như thế nào?

* Xử dụng vòng lặp trong thymeleaf ta dùng th:each
  + <… th:each=”item : ${itemList}”>
* Xử dụng điều kiện trong thymeleaf dùng th:if
  + < some HTML tag th:if=”condition”>

1. Formatter là gì? Converter là gì?

|  |  |
| --- | --- |
| Converter | Formatter |
| Chuyển đổi kiểu dữ liệu này sang kiểu dữ liệu khác  Kiểu dữ liệu nguồn là bất kỳ kiểu nào | Giống converter nhưng kiểu dữ liệu nguồn của formatter là string. Chuyển đổi dữ liệu trong form nên dùng hơn |

1. ORM là gì?(Object ralational mapping)

Là 1 kỹ thuật liên kết đối tượng trong ngôn ngữ lập trình và đối tượng trong cơ sở dữ liệu

(là quá trình biến đổi 1 record trong database và 1 object trong OOP)

ORM có rất nhiều triển khai (Hibernate, Eclipselink, Ibatis..) trong đó Hibernate là ORM Framework nổi tiếng nhất và sử dụng rộng rải trong Java

* Ưu

Thao tác với DB dễ dàng hơn

Ko quan tâm đến hệ CSDL đang dùng

Giúp lập trình viên tập trung vào nghiệp vụ

* Nhược:

Đối với database có dữ liệu lớn thì hiệu suất kém hiệu quả

LTV hay rơi vào bẫy do truy xuất dữ liệu quá đơn giản

1. JPA là gì? Spring Data JPA là gì?

Jpa là 1 bộ đặc tả, cung cấp bộ interface cho ORM triển khai.

Spring Data JPA là 1 phần của dự án Spring Data giúp làm việc với repository dễ dàng hơn.

1. Hibernate là gì? Hibernate làm gì với database

Là một trong những ORM framework giúp LTV viết ứng dụng Java có thể mapping các Object với hệ CSDL và hỗ trợ thực hiện các khả năng OOP với CSDL

Hibernate có thể hỗ trợ tạo bảng và các dữ liệu thông qua ngôn ngữ Java.

1. Liệt kê một số annotation của hibernate?

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| @Enttity | @Table | @Column | @Id | @GeneratedValue |
| @ManyToOne | @OneToMany | @OneToOne | @ManyToMany | @OrderBy |

1. Entity là gì?

* Là 1 thực thể đại diện cho dữ liệu trong ứng dụng
* Thường là 1 POJO(plain old java object) 1 lớp đơn giản chỉ mô tả dữ liệu ko có xử lý logic nghiệp vụ
* 1 entity đc mapping với 1 table trong CSDL(1-1)
* Mỗi entity cần phải tuân thủ:

+ Có @Entity @Id

+ Có contructor mặc định ko có tham số

+ Không được đặt final

+ Thuộc tính private,protected hoặc ở mức package private( ko public)

1. Persistence Context & Entity Manager là gì?

* Persistence Context là tập các thể hiện của entity được quản lý, tồn tại trong một kho dữ liệu

(Có thể coi là một môi trường chưa toàn bộ các đối tượng mà ta tạo ra và lưu vào csdl trong mỗi session)

* Trong hibernate, persistence context là một instance của EntityManager trong JPA
* Entity manager gồm các chức năng
  + Khai báo phương thức để tương tác với persistence context
  + Tạo hoặc xoá các thể hiện của entity
  + Tìm kiếm entity theo khoá chính
  + Thực thi các câu lệnh truy vấn lên entity

1. Câu lệnh truy vấn động là gì? Câu lệnh truy vấn tĩnh là gì?

* Câu truy vấn động có thể truyền tham số và thay đổi linh động tùy thuộc vào giá trị tham số truyền vào
* Dùng PT createQuery() của Entity Manager để tạo câu truy vấn động
* Câu truy vấn tĩnh ko thể truyền tham số trong lúc compile
* Dùng PT createNameQuery() để tạo câu truy vấn tĩnh

1. Trạng thái của Entity bao gồm những gì?

* New: mới được tạo ra chưa được lưu xuống CSDL
* Persistent: được ghi xuống CSDL và quản lý trong persistence context
* Detached: tạm thời tách ra khỏi Persistence context
* Removed: Mặc dù JPA chỉ cho phép xoá các entity đã được quản lý, hibernate vẫn có thể xoá được nhưng detached entity (entity tách ra). Entity trạng thái removed thực ra chưa hoàn toàn bị xoá cho đến khi session flush được gọi

1. Có bao nhiêu loại mapping trong Hibernate?

Có 3 loại mapping là:

* Collections Mappings: nếu một entity hoặc lớp có một tập các giá trị trong một biến collection cụ thể thì chúng ta có thể ánh xạ các giá trị đó bằng bất kỳ một trong các interface collections có sãn trong java.
* Association Mappings: việc ánh xạ các liên kết giữa các lớp Entity và các mối quan hệ giữa các bảng chính là linh hồn của ORM (many to one, many to many, one to many, one to one)
* Component Mappings: trường hợp một lớp Entity có thể có một tham chiếu đến một lớp khác như một biến thành viên. Nếu lớp được tham chiếu đó không có vòng đời của chính nó và hoàn toàn phụ thuộc vào vòng đời của lớp thực thể sở hữu, lớp được tham chiếu đó gọi là lớp component

1. Để sử dụng interface JpaRepository cần cung cấp những thông tin nào?

* Ta cần cung cấp Đối tượng và Kiểu dữ liệu của khoá chính của đối tượng

1. Validation dữ liệu là gì?

Thực hiện đánh giá xác minh kiểm tra tính hợp lệ của dữ liệu đầu vào

1. Trình bày cách triển khai validate dữ liệu trong Spring

Có 3 cách triển khai là:

* Sử dụng annotation có sẵn trong spring validator
* Dùng custom validation
* Dùng custom annotation

1. Binding Result là gì?

BindingResult là nơi Spring chứa kết quả của việc xác thực dữ liệu.

1. AOP là gì? AOP gồm những yếu tố nào?

AOP (Aspect Oriented Programming) là kỹ thuật lập trình hướng khía cạnh

* Tách Chương trình chính thành các module riêng lẻ
* Mỗi module sẽ thực hiện 1 nhiệm vụ nào đó
* Khi cần chỉ cần chỉnh sửa trong 1 module

Những yếu tố trong AOP là:

* Aspect: khía cạnh là mối quan tâm xuyên suốt
* JoinPoint: là 1 vị trí xác định trong luồng thực thi của chương trình(1 lời gọi phương thức)
* PoinCut: là tập hợp các JoinPoint
* Advice: 1 hành động cụ thể của aspect. Gồm
  + Before
  + After
  + Around
  + After returning
  + After throwing
* Target Object là đối tượng mà Advice được áp dụng
* Proxy: như tầng bảo vệ không cho truy cập trực tiếp vào các module
* Weaving: Quá trình đan xen các module lại

1. Các loại Advice?

Advice: 1 hành động cụ thể của aspect. Gồm

* 1. Before: thực thi trước khi joint point
  2. After: thực thi sau khi joint point kết thúc dù thành công hay thất bại
  3. Around: thực thi trước và sau khi joint point
  4. After returning: thực thi sau khi joint point hoàn thành bình thường
  5. After throwing: thực thi khi joint point kết thúc có ngoại lệ

1. JoinPoint là gì?

* Điểm nối, là những vị trí xác định trong luồng thực thi của chương trình mà tại đó một khía cạnh sẽ bắt đầu tham gia vào. Một Jointpoint có thể là một lời gọi phương thức, một ngoại lệ được throw ra, hay một field được thay đổi.
* Vd: Chẳng hạn chúng ta cần ghi log lại sau khi chạy method nào đó thì điểm ngay sau method đó được thực thi gọi là một Jointpoint.

1. Nêu cách triển khai AOP mà bạn dùng trong dự án

1. Khởi tạo đối tượng Session cùng annotation @ModelAttribute

2. Dùng annotation @SessionAttributes để lưu trữ thông tin của Model Attribute

3. Sử dụng annotation @CookieValue để ràng buộc giá trị của cookie HTTP với tham số phương thức trong Controller

4. Tạo Cookie và trả về cho Client

1. Mục đích xử lý ngoại lệ trong Spring Web MVC?

Để bắt các ngoại lệ và thông báo lỗi cho người dùng cũng như lỗi cụ thể cho lập trình viên

1. Các cách xử lý ngoại lệ trong Spring Web MVC?

+ Sd try catch finally (throw throws)

+ Sd @ExceptionHandler:

Tạo Exception handle controller và đánh dấu bằng @ControllerAdvice

Trong mỗi phương thức xử lý ngoại lệ thêm annotation @ExceptionHandler (exceptionName.class)

1. Cookie là gì?

Cookie là những trang web do người dùng truy cập tạo ra, giúp trải nghiệm trực tuyến dễ dàng hơn, lưu trữ ở client

Ứng dụng trong quảng cáo rượt đuổi , ghi nhớ đăng nhập

1. Phân biệt Session và Cookie

|  |  |
| --- | --- |
| Cookie | Session |
| Dữ liệu được lưu trữ ở phía trình duyệt người dùng | Dữ liệu session được lưu phía máy chủ |
| Cookie tồn tại cho đến khi expired | Sau khi đóng trình duyệt sẽ mất thông tin session |
| Dữ liệu cookie dễ dàng sửa đổi, dễ đánh cắp | Dứ liệu session không dễ dàng sửa đổi |

1. Session là gì? Cho ví dụ về session?

* Session là một phiên làm việc giữa client và server, thường được dùng để lưu trữ dữ liệu tạm thời
* Sesstion bắt đầu khi có 1 request gửi lên server và kết thúc khi hết phiên làm việc.
* Các trường hợp hết phiên làm việc:
  + Tắt trình duyệt
  + Logout
  + Hết thời gian timeout (mặc định 30p)
  + Tắt Server

1. Nêu cách triển khai Session mà bạn dùng trong dự án

B1: Khai báo tên session: sd @SessionAttributes(“sessionName”)

B2: Khơi tạo session với tên được khai báo ở trên: sd @ModelAttribute(“sessionName”) (được thực thi khi chưa có session ở B1

B3: Cập nhập session bằng cách dùng @SessionAttribute để lấy session xuống và cập nhập

B4: Gọi session để sử dụng cũng bằng @SessionAttribute

1. Web Service là gì? Lấy ví dụ về web service

là dịch vụ web cung cấp các nền tảng chung cho các ứng dụng. Các ứng dụng này có thể được viết từ nhiều ngôn ngữ khác nhau nhưng có thể giao tiếp được với nhau (thông qua data)

* Các kiểu dữ liệu thường dùng là XML hoặc JSON

1. Phân biệt Web Service và Web truyền thông

|  |  |
| --- | --- |
| Web service | Web application |
| Không có giao diện chỉ trả về dữ liệu dạng JSON hoặc XML | Có giao diện để người dùng tương tác |
| Tương tác giữa các ứng dụng với nhau thông qua các pt là HTTP Get Post Put Path Delete và HTTP status code | Tương tác giữa người dùng và ứng dụng thông qua thẻ form, link, image, button |
| Kết quả trả về là data dạng XML hoặc JSON | Kết quả trả về là view, Video, hình ảnh, âm thanh |

1. SOAP là gì? RESTful là gì?

-SOAP: là giao thức sử dụng XML( cú pháp nghiêm ngặt, môi trường đóng do sd trong doanh nghiệp)

-REST: là giao thức sử dụng xml, json (tốc độ nhanh hơn nhưng độ bảo mật thấp hơn)

1. jQuery là gì? Ajax là gì?

* Jquery là 1 thư viện dựa trên javascript được sd trong các dự án phần mềm (write less do more)
* Ưu điểm
  + Thao tác với DOM dễ dàng hơn
  + Hỗ trợ AJAX
  + Xử lý event tốt hơn >> code ngắn gọn hơn
  + Chỉ sử dụng 1 file duy nhất
  + Hỗ trợ nhiều loại trình duyệt
  + Xử lý hiệu ứng chuyển động
* AJAX (Asynchronous Javascript and XML) là 1 kỹ thuật để phát triển trang web, không cần load lại trang và chỉ cần load lại 1 phần của trang
* Ưu điểm
  + Trải nghiệm phía người dùng tốt hơn
  + Giảm tải băng thông phía server
* Nhược điểm
  + Bảo mật kém
  + Không lưu lại lịch sử => ko thể sd nút back của trình duyệt
  + Không thể tạo thẻ book mark

1. Nêu ví dụ về cách bạn triển khai Ajax trong dự án của bạn
2. @RequestHeader và @ResponseHeader có ý nghĩa gì

* @RequestHeader - các Headers định nghĩa các tham số dùng cho việc data requested hoặc đưa ra các thông tin quan trọng trong việc giúp client tạo request. Response header - Các tham số chứa thông tin về các response trả về.
* @ResponseHeader- Các trường Header này chỉ có khả năng áp dụng cho các thông báo phản hồi. Kiểu thực thể (Entity-Header): Các trường này xác định thông tin về thân-thực thể hoặc, nếu không có phần thân nào hiển thị, về nguồn được nhận diện bởi yêu cầu

1. Interceptor là gì?

Là những interceptor có cấp độ thấp hơn được sử dụng để theo dõi các request và response được truyền quan mạng. Nó thì rất hữu ích để theo dõi việc redirect, retry và tạo ra truy cập đến những chi tiết của request. Chúng không được gọi nếu response đã được lưu trữ.

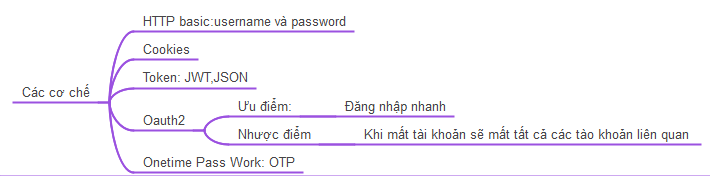
1. Bảo mật là gì? Cơ chế bảo mật trong Spring như thế nào?

* Bảo mật là sự hạn chế khả năng lạm dụng tài nguyên và tài sản
* Cơ chế bảo mật trong Spring:
  + Authentication (xác thực): là một hành động nhằm thiết lập hoặc chứng thực một thông điệp hoặc đối tượng nào đó đáng tin cậy
  + Authorization (phân quyền): là quá trình xác định xem một đối tượng có quyền truy cập một tài nguyên cụ thể để thực hiện một số hành động hay không

1. Authentication là gì? Authorization là gì?

* Authentication (xác thực): là kiểm tra xem bạn có phải là người dùng của ứng dụng không
* Authorization (Phân quyền): Là kiểm tra xem người dùng có những quyền nào

1. Các cơ chế xác thực?



1. CSRF là gì

CSRF (Cross-site Request Forgery) là một kỹ thuật tấn công giả mạo chính chủ thể của nó

1. CORS là gì

CORS (Cross-origin Resource Sharing) là một cơ chế cho phép nhiều tài nguyên khác nhau của một trang web có thể truy cấp từ một domain khác với domain của trang

1. Spring Boot là gì?

* Là một trong các module của spring được sử dụng để tạo ra các ứng dụng độc lập có thể chạy ngay với rất ít config.
* Nhúng sẵn Tomcat, Jetly
* Cung cấp sẵn các thư viện starter (tự động cấu hình nếu có thể)
* Ko yêu cầu cấu hình file.XML

1. Sự khác nhau giữa SpringBoot và SpringMVC là gì?

|  |  |
| --- | --- |
| Spring Boot | Spring MVC |
| Spring Boot là một mô-đun của Spring để đóng gói ứng dụng dựa trên Spring với các giá trị mặc định hợp lý. | Spring MVC là model-view-controller web framework trong khuôn khổ Spring framework. |
| Cung cấp các cấu hình mặc định để xây dựng khung công tác được cung cấp bởi Spring. | Cung cấp các tính năng sẵn sàng sử dụng để xây dựng một ứng dụng web. |
| Không cần phải xây dựng cấu hình theo cách thủ công. | Nó yêu cầu cấu hình xây dựng theo cách thủ công. |
| Không yêu cầu đối với bộ mô tả triển khai. | Cần có bộ mô tả Triển khai. |
| Nó tránh mã viết sẵn và kết hợp các phần phụ thuộc lại với nhau trong một đơn vị duy nhất. | Nó chỉ định từng phụ thuộc riêng biệt. |
| Nó làm giảm thời gian phát triển và tăng năng suất. | Cần nhiều thời gian hơn để phát triển |

1. Cấu hình cho Spring Boot Tìm kiếm các Bean ở nhiều package khác nhau bằng cách nào?

C1: sd @componentScan

C2: sd scanBasePackages trong @SpringBootApplication

1. Trình bày Spring Security? @EnableWebSecurity làm gì?

Spring Security là 1 framework tập trung vào việc cung cấp khả năng xác thực và phân quyền cho ứng dụng java

@EnableWebSecuriry để kích hoạt tính năng Spring security trên ứng dụng web của mình

1. Trình bày Spring Boot JPA? Cài đặt?

Spring Boot JPA là một phần trong hệ sinh thái Spring Data, nó tạo ra một layer ở giữa tầng service và database, giúp chúng ta thao tác với database một cách dễ dàng hơn, tự động config và giảm thiểu code thừa thãi.

Cài đặt: Thêm dependency spring-boot-stater-data-jpa