Due: 17:00 04/20/2022

Raise a PR to: <https://github.com/TAIsRich/chuwa4.1ShortQuestions.git>**·Remember PR to the branch names with your name.**

***It is super encouraged to submit before 04/18 class. At least completed 1 - 9 and read the two links in 10 before next Monday’s class.***

* **What is IOC?**
* Ans: IOC, Inversion of Control is a programming principle. With IOC, computer programs receive flow of control from a generic framework. A software architecture with this design inverts control as compared to traditional procedural programming.
* With inversion of control, it is the framework that calls into the custom, or task-specific, code, instead of the custom code calls into reusable libraries.
* **What is DI?**
* Ans: dependency injection (DI) is a design pattern in which an object receives other objects that it depends on
* how many DI injections in Spring Boot? please also give me code example.(只给你能展示这几种方法的例子就行，不用写完整的代码)

**1.Setter Dependency Injection (SDI)**

public class SetterDI {

Haha ha;

public void setHaha(Haha ha){

this.ha = ha;

}

}

<bean id="Haha" class="....com.Haha">

<property name="haha">

<ref bean="CsvHA" />

</property>

</bean>

2.**Constructor Dependency Injection (CDI)**

public class ConstructorDI {

Haha ha;

ConstructorDI(Haha ha){

this.ha = ha;

}

}

<bean id="ConstructorDI" class="com.\*\*\*.org.ConstructorDI">

<constructor-arg>

<bean class="com.\*\*\*.org.ConstructorDI" />

</constructor-arg>

</bean>

**Which DI Injection is better way to use?**

Ans:when the properties are more is Constructor DI is better.

**What are HTTP request methods?**

ans：Hypertext Transfer Protocol (HTTP) is designed to enable communications between clients and servers.

HTTP request methods are the assets that indicate the specific desired action to be performed on a given resource.

**What are differences between POST and PUT?**

POST means "create new" as in "Here is the input for creating a user, create it for me".

PUT means "insert, replace if already exists" as in "Here is the data for user 5".

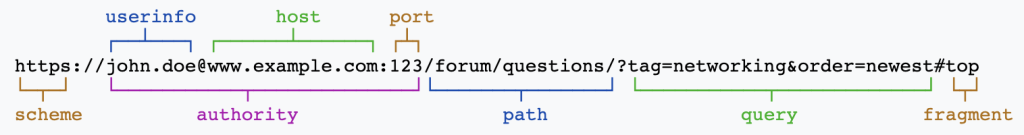
* What is URL and What is URI?

read this article **again and again and again PLEASE** (<https://danielmiessler.com/study/difference-between-uri-url/>)

ans：URL:location; URI: identifier; URN:name(ex:urn:isbn:0-486-27557-4 )

URI:only the name of a resource( ex:google.com)

URL : because it’s both the name of a resource (google.com), and the way to get there (https://).(ex: **https://google.com**)



**What is path variable? and what is Request Parameter?**

Ans: path variable: @PathVariable annotation can be used to handle template variables in the request URI mapping

* Request Parameter:
* Ex: https://www.bing.com/search?q=ToolsQA
* In this URL Request parameter is represented by the "q=ToolsQA" part of the URL. Request parameter starts with a question mark (?). Request parameters follow "Key=Value" data format.

**What are HTTP Response Status Codes?**

* [Informational responses](https://developer.mozilla.org/en-US/docs/Web/HTTP/Status#information_responses) (100–199)
* [Successful responses](https://developer.mozilla.org/en-US/docs/Web/HTTP/Status#successful_responses) (200–299)
* [Redirection messages](https://developer.mozilla.org/en-US/docs/Web/HTTP/Status#redirection_messages) (300–399)
* [Client error responses](https://developer.mozilla.org/en-US/docs/Web/HTTP/Status#client_error_responses) (400–499)
* [Server error responses](https://developer.mozilla.org/en-US/docs/Web/HTTP/Status#server_error_responses) (500–599)
* Design a collect of APIs for simple YouTube contains below modules(不求绝对最好，能力范围内做最好就行，哪怕写一行字，也要提交作业，只要交作业，就比不交强)
* <https://restfulapi.net/resource-naming/>
* <https://www.freecodecamp.org/news/rest-api-best-practices-rest-endpoint-design-examples/>
* Search. we have v1 and v2 two versions.
* v1, v2
* try to make it be powerful.
* use keyword to search
* search one video between 04/01/2022 to now
* search apple in category Tech or Fruit
* pagination. if we got 100 results, we split it to 10 each page.
* and more if you can think out.
* members
* playlist
* add new one to playlist
* remove
* comments
* add new
* update old
* delete

(edited)

package com.example.demo;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.web.bind.annotation.\*;

@SpringBootApplication

@RestController

public class DemoApplication {

public static void main(String[] args) {

SpringApplication.*run*(DemoApplication.class, args);

}

@GetMapping("/hello")

public String hello(@RequestParam(value = "name", defaultValue = "World") String name){

return String.*format*("Hello %s !", name);

}

@RequestMapping(value="/hello/{id}/{name}",method= RequestMethod.*GET*)

public String sayHello(@PathVariable("id") Integer id,@PathVariable("name") String name){

return "id:"+id+" name:"+name;

}

@PostMapping("/ha")

public String ha( ) {

return "this is a post\n";

}

@RequestMapping(value="/hello",method= RequestMethod.*GET*)

//required=false 表示url中可以无id参数，此时就使用默认参数

public String sayHello(@RequestParam(value="id",required = false,defaultValue = "1") Integer id){

return "id:"+id;

}

@RequestMapping(value="/hello",method= RequestMethod.*GET*)

public String sayHello(@RequestParam Integer id,@RequestParam String name){

return "id:"+id+ " name:"+name;

}

@RequestMapping(value="/hello",method= RequestMethod.*GET*)

public String sayHello(@RequestParam("userId") Integer id){

return "id:"+id;

}

}

Daniel Miessler**Daniel Miessler**

[**The Difference Between a URL, URI, and a URN**](https://danielmiessler.com/study/difference-between-uri-url/)

A URI is an Identifier—like a specific document, book, or website—and a URL is an identifier that also tells you how to access it, such as…

**Est. reading time**

10 minutes

<https://danielmiessler.com/study/difference-between-uri-url/>

freeCodeCamp.org**freeCodeCamp.org**

[**REST API Best Practices – REST Endpoint Design Examples**](https://www.freecodecamp.org/news/rest-api-best-practices-rest-endpoint-design-examples/)

In Web Development, REST APIs play an important role in ensuring smooth communication between the client and the server. You can think of the client as the front end and the server as the back end. Communication between the client (frontend) and the server (backend) isn't usually super direct. So

**Written by**

Kolade Chris

**Filed under**

Rest Api, API, Best Practices

Sep 16th, 2021