web作业5

2018211314 2018211507 陈姝仪

简答题

1.在Spring框架中,什么是控制反转?什么是依赖注入?使用控制反转和依赖注入有什么好处?

答:

- 控制反转: IoC的设计原则是引入一个调控系统内所有对象的外界实体"容器",由容器来管理应用程序需要使用的各种对象和它们之间的依赖关系,控制器由调用者自身转移到容器,控制权发生了反转。控制反转是一种借助描述(XML或注解)由容器负责创建对象和维护对象间依赖关系,而不是对象自己负责创建和解决自己的依赖的方式。
- 依赖注入:对象在被创建的时候,由容器将其所依赖的对象的引用传递给它。

• 好处: 降低耦合度

2.bean的作用域分别设置为singleton、prototype、request时,bean的创建、销毁有何区别?

singleton	默认的作用域,使用singleton定义的Bean在Spring容器中只有一个Bean实例。 (非线程安全)
prototype	Spring容器每次获取prototype定义的Bean,容器都将创建一个新的Bean实例。
request	在一次HTTP请求中容器将返回一个Bean实例,不同的HTTP请求返回不同的Bean 实例。仅在Web Spring应用程序上下文中使用。

3.简述基于注解的装配方式的基本用法。

用注解声明的Bean,可以通过组件扫描让Spring IoC容器发现(@ComponentScan)

- 声明Bean的注解:主要有以下四种,表示注解的是一个组件对象(Bean),四种注解功能相同, 仅有语义区别,建议按Bean的作用使用更准确的注解。
 - @Component, 通用注解
 - @Repository, 注解数据访问层Bean[]
 - @Service, 注解业务逻辑层Bean
 - @Controller, 注解控制器层Bean
- 辅助用的注解: @Scope用于指定作用域 @Value设置值
- 注入Bean用的注解: 主要有@Autowired, @Resource, @Qualifier
 - @Autowired:可以对类成员变量、方法及构造函数进行注解,完成自动装配的工作,可以消除 setter和getter方法,默认按照Bean的类型进行装配
 - @Resource: 功能和@Autowired一样,区别在于默认按照名称来装配
 - @Qualifier:与@Autowired配合使用,当@Autowired注解需要按照名称来装配时需要和该注解一起使用,Bean实例名称由@Qualifier注解参数指定

4.假如使用前后端分离架构设计个人通讯录管理系统,请设计一下 RESTFUL风格的对通讯录进行增加(Create)、检索(Retrieve)、更新 (Update)和删除(Delete)操作的接口API

```
@RequestMapping(value = "/", method = RequestMethod.POST)
public ResponseData add(Article article) {
   articleRepository.save(article);
  // return "{success:true,message: \"添加成功\" }";
   return new ResponseData(ExceptionMsg.SUCCESS,article);
}
//删
@RequestMapping(value = "/{id}", method = RequestMethod.DELETE)
public Response delete(@PathVariable("id") long id) {
      RestTemplate client= restTemplateBuilder.build();
   String uri = "http://localhost:8080" + "/{id}";
   Map map= new HashMap();
   map. put ("orderid",id);
   Void article = client.delete(uri,map,id);*/
    articleRepository.deleteById(id);
   return result(ExceptionMsg.SUCCESS);
    //return new ResponseData(ExceptionMsg.SUCCESS,"");
  }
//改
@RequestMapping(value = "/{id}", method = RequestMethod.PUT)
public ResponseData update(Article model) {
    articleRepository.save(model);
    return new ResponseData(ExceptionMsg.SUCCESS, model);
}
@RequestMapping(value = "/{id}", method = RequestMethod.GET)
public ResponseData findArticle(@PathVariable("id") Integer id) throws
IOException {
         Article article = articleRepository.findById(id);
    if (article != null) {
        return new ResponseData(ExceptionMsg.SUCCESS,article);
   }
    return new ResponseData(ExceptionMsg.FAILED,article);
}
```

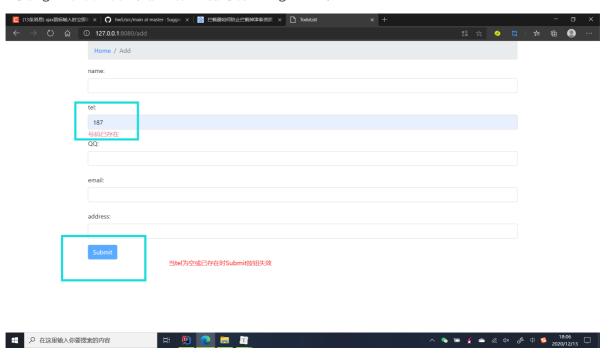
编程题

个人通讯录管理系统Part3

在上一版功能基础上进行如下改造

- 1.在添加联系人页面增加一处AJAX操作: 用户输完电话号码后使用AJAX发起请求判断该号码是否已存在并予以提示
- 2.将通讯录的增删改查操作使用JPA技术予以实现。

3.使用git进行代码管理,提交作业时提供项目的git地址即可



陈姝仪的github仓库: https://github.com/Suggie-Chen/hw5/tree/master/src