## (1) Spring在SSM起什么作用？

Spring是一个轻量级框架，也是一个容器，Spring实质上讲就是一个Bean工厂，主要用来管理Bean的生命周期和框架集成。有IOC控制反转，DI依赖注入，控制反转是把dao依赖注入到servic层，然后service层反转给action层，Spring的顶层容器为BeanFactory，常用的ApplicationContext为它的子接口，实现了工厂模式，Spring还提供了AOP的支持，方便在切面级开发。

## (2) 谈谈你mvc的理解?

MVC是Model—View—Controller的简称。即模型—视图—控制器。MVC是一种设计模式，它强制性的把应用程序的输入、处理和输出分开。

MVC中的模型、视图、控制器它们分别担负着不同的任务。

视图: 视图是用户看到并与之交互的界面。视图向用户显示相关的数据，并接受用户的输入。视图不进行任何业务逻辑处理。

模型: 模型表示业务数据和业务处理。相当于JavaBean。一个模型能为多个视图提供数据。这提高了应用程序的重用性.

控制器: 当用户单击Web页面中的提交按钮时,控制器接受请求并调用相应的模型去处理请求。

然后根据处理的结果调用相应的视图来显示处理的结果。

MVC的处理过程：首先控制器接受用户的请求，调用相应的模型来进行业务处理，并返回数据给控制器。控制器调用相应的视图来显示处理的结果。并通过视图呈现给用户。

## (3) #{}和${}的区别是什么？

答：${}是Properties文件中的变量占位符，它可以用于标签属性值和sql内部，属于静态文本替换。

比如:${driver}会被静态替换为com.mysql.jdbc.Driver。#{}是sql的参数占位符，Mybatis会将sql中的#{}替换为?号，

在sql执行前会使用PreparedStatement的参数设置方法，按序给sql的?号占位符设置参数值，比如ps.setInt(0, parameterValue)，

#{item.name}的取值方式为使用反射从参数对象中获取item对象的name属性值，相当于param.getItem().getName()。

## (4) spring mvc常用4个基本注解

@Component、@Repository 、@Service、@Controller

@Controller 控制层 ，通常我们所说的action层

@Service 业务逻辑层，通常我们所说的service层或者manger层

@Repository 持久层，通常我们所说的Dao层

@Component 组件，书面意思就是我们不知道在哪一层时使用

虽然对于spring来说，这4个注解的效果都是一样的，当spring扫描到这些注解时，都会当做需要注入的bean加载到上下文中

但是在项目中，我们应该严格按照这几个注解的含义来使用，这样有利于我们web架构.

## (5) 什么是控制反转(IOC)？什么是依赖注入？

控制反转是应用于软件工程领域中的，在运行时被装配器对象来绑定耦合对象的一种编程技巧，对象之间耦合关系在编译时通常是未知的。在传统的编程方式中，业务逻辑的流程是由应用程序中的早已被设定好关联关系的对象来决定的。在使用控制反转的情况下，业务逻辑的流程是由对象关系图来决定的，该对象关系图由装配器负责实例化，这种实现方式还可以将对象之间的关联关系的定义抽象化。而绑定的过程是通过“依赖注入”实现的。

控制反转是一种以给予应用程序中目标组件更多控制为目的设计范式，并在我们的实际工作中起到了有效的作用。

依赖注入是在编译阶段尚未知所需的功能是来自哪个的类的情况下，将其他对象所依赖的功能对象实例化的模式。这就需要一种机制用来激活相应的组件以提供特定的功能，所以依赖注入是控制反转的基础。否则如果在组件不受框架控制的情况下，框架又怎么知道要创建哪个组件？

## (6) SpringMVC的工作流程?

1. 用户发送请求至前端控制器DispatcherServlet

2. DispatcherServlet收到请求调用HandlerMapping处理器映射器。

3. 处理器映射器根据请求url找到具体的处理器，生成处理器对象及处理器拦截器(如果有则生成)一并返回给DispatcherServlet。

4. DispatcherServlet通过HandlerAdapter处理器适配器调用处理器

5. 执行处理器(Controller，也叫后端控制器)。

6. Controller执行完成返回ModelAndView

7. HandlerAdapter将controller执行结果ModelAndView返回给DispatcherServlet

8. DispatcherServlet将ModelAndView传给ViewReslover视图解析器

9. ViewReslover解析后返回具体View

10. DispatcherServlet对View进行渲染视图(即将模型数据填充至视图中)。

11. DispatcherServlet响应用户

## (7) 简单介绍下你对mybatis的理解？

1. mybatis配置

2. SqlMapConfig.xml，此文件作为mybatis的全局配置文件，配置了mybatis的运行环境等信息。

3. mapper.xml文件即sql映射文件，文件中配置了操作数据库的sql语句。此文件需要在SqlMapConfig.xml中加载。

4. 通过mybatis环境等配置信息构造SqlSessionFactory即会话工厂

5. 由会话工厂创建sqlSession即会话，操作数据库需要通过sqlSession进行。

6. mybatis底层自定义了Executor执行器接口操作数据库，Executor接口有两个实现，一个是基本执行器、一个是缓存执行器。

7. Mapped Statement也是mybatis一个底层封装对象，它包装了mybatis配置信息及sql映射信息等。mapper.xml文件中一个sql对应一个Mapped Statement对象，sql的id即是Mapped statement的id。

8. Mapped Statement对sql执行输入参数进行定义，包括HashMap、基本类型、pojo，Executor通过Mapped Statement在执行sql前将输入的java对象映射至sql中，输入参数映射就是jdbc编程中对preparedStatement设置参数。

9. Mapped Statement对sql执行输出结果进行定义，包括HashMap、基本类型、pojo，Executor通过Mapped Statement在执行sql后将输出结果映射至java对象中，输出结果映射过程相当于jdbc编程中对结果的解析处理过程。