每题5分，共120分；

### Vue优点是什么？

|  |
| --- |
| 轻量级框架：只关注视图层，是一个构建数据的视图集合，大小只有几十kb；  简单易学：国人开发，中文文档，不存在语言障碍 ，易于理解和学习；  双向数据绑定：保留了angular的特点，在数据操作方面更为简单；  组件化：保留了react的优点，实现了html的封装和重用，在构建单页面应用方面有着独特的优势；  视图，数据，结构分离：使数据的更改更为简单，不需要进行逻辑代码的修改，只需要操作数据就能完成相关操作；  虚拟DOM：dom操作是非常耗费性能的， 不再使用原生的dom操作节点，极大解放dom操作，但具体操作的还是dom不过是换了另一种方式；  运行速度更快:相比较与react而言，同样是操作虚拟dom，就性能而言，vue存在很大的优势。 |

### session 和 cookie 有什么区别？

|  |
| --- |
| Cookie:主要用在保存客户端，其值在客户端与服务端之间传送，不安全，存储的数据量有限。  Session:保存在服务端，每一个session在服务端有一个sessionID作一个标识。存储的数据量大，安全性高。占用服务端的内存资源  cookie不是很安全，别人可以分析存放在本地的cookie并进行cookie欺骗，相当重要的数据，应该使用session保存到服务端。  session会在一定时间内保持在服务器上，但是会占用内存资源，当访问的用户过多，会加重服务器的负载，考虑到减轻服务器的压力，可以将不重要的数据放在cookie中持久的保存。  单个cookie保存的数据不能超过4k，很多浏览器都限制站点最多保存20个cookie。  由于HTTP协议是无状态的协议，所以服务端需要记录用户的状态时，就需要用某种机制来识具体的用户，这个机制就是Session.典型的场景比如购物车，当你点击下单按钮时，由于HTTP协议无状态，所以并不知道是哪个用户操作的，所以服务端要为特定的用户创建了特定的Session，用用于标识这个用户，并且跟踪用户，这样才知道购物车里面有几本书。  这个Session是保存在服务端的，有一个唯一标识。在服务端保存Session的方法很多，内存、数据库、文件都有。  集群的时候也要考虑Session的转移，在大型的网站，一般会有专门的Session服务器集群，用来保存用户会话，这个时候 Session 信息都是放在内存的，使用一些缓存服务比如Memcached之类的来放 Session。  这个时候Cookie就登场了。每次HTTP请求的时候，客户端都会发送相应的Cookie信息到服务端。实际上大多数的应用都是用 Cookie 来实现Session跟踪的，第一次创建Session的时候，服务端会在HTTP协议中告诉客户端，需要在 Cookie 里面记录一个Session ID，以后每次请求把这个会话ID发送到服务器，我就知道你是谁了。 |

### Servlet API中forwared()和redirect()的区别？

|  |
| --- |
| a、浏览器显示  重定向会改变URL地址，请求转发不会改变URL地址。  b、资源共享  重定向不可以资源共享，请求转发可以。  c、功能  重定向可以用URL绝对路径访问其他Web服务器的资源，而请求转发只能在一个Web应用程序内进行资源转发，即服务器内部的一种操作。  d、效率  重定向效率低，相当于再一次请求；请求转发效率相对较高，跳转仅发生在服务器端。 |

### get和post的区别？

|  |
| --- |
| a、get是用来从服务器上获取数据，而post是用来向服务器传递数据；  b、get将表单中数据按照variable=value的形式，添加到action所指向的URL后面，并且两者用"？"连接，变量之间用"&"连接；  而post是将表单中的数据放在form的数据体中，按照变量与值对应的方式，传递到action所指定的URL。  c、get是不安全的，因为在传输过程中，数据是被放在请求的URL中;而post的所有操作对用户来说都是不可见的。  d、get传输的数据量小，这主要应为受url长度限制;而post可以传输大量的数据，所有上传文件只能用post提交。  e、get限制form表单的数据集必须为ASCII字符；而post支持整个IS01 0646字符集。  f、get是form表单的默认方法。 |

### 如何从form表单中获取checkbox的值？

|  |
| --- |
| 可以在页面将checkbox的name属性设置相同，value属性取每个条目的id，在后台使用request.getParametervalues（"name"）;获取一组值。 |

### 6、说出几种vue当中的指令和它的用法？

|  |
| --- |
| v-model双向数据绑定；  v-for循环；  v-if v-show 显示与隐藏；  v-on事件；v-once: 只绑定一次。 |

### 7、什么是 Thymeleaf？Thymeleaf最大的特点是什么？

|  |
| --- |
| Thymeleaf 是一个支持原生 HTML 文件的 Java 模版引擎，可以实现前后端分离的交互方式，即视图与业务数据分开响应，它可以直接将服务端返回的数据生成 HTML 格式，同时也可以处理 XML、JavaScript、CSS 等格式。  Thymeleaf 最大的特点是既可以直接在浏览器打开，就像访问静态页面一样看到样式，也可以结合服务端将业务数据填充进去看到动态生成的页面。Spring Boot 对 Thymeleaf 模版做了很好的集成，在 Spring Boot 应用中使用 Thymeleaf 非常方便。 |

### Vue中引进组件的步骤是什么？

|  |
| --- |
| 在template中引入组件；  在script的第一行用import引入路径；  用component中写上组件名称。 |

### mybatis中#{}和${}的区别是什么？

|  |
| --- |
| #{}是预编译处理，${}是字符串替换。  Mybatis在处理#{}时，会将sql中的#{}替换为?号，调用PreparedStatement的set方法来赋值；  Mybatis在处理${}时，就是把${}替换成变量的值。  使用#{}可以有效的防止SQL注入，提高系统安全性。 |

### mybatis中当实体类中的属性名和表中的字段名不一样 ，怎么办 ？

|  |
| --- |
| 第1种： 通过在查询的sql语句中定义字段名的别名，让字段名的别名和实体类的属性名一致。  <select id=”selectorder” parametertype=”int” resultetype=”me.gacl.domain.order”>  select order\_id id, order\_no orderno ,order\_price price form orders where order\_id=#{id};  </select>  第2种： 通过<resultMap>来映射字段名和实体类属性名的一一对应的关系  <select id="getOrder" parameterType="int" resultMap="orderresultmap">  select \* from orders where order\_id=#{id}  </select>  <resultMap type=”me.gacl.domain.order” id=”orderresultmap”>  <!–用id属性来映射主键字段–>  <id property=”id” column=”order\_id”>  <!–用result属性来映射非主键字段，property为实体类属性名，column为数据表中的属性–>  <result property = “orderno” column =”order\_no”/>  <result property=”price” column=”order\_price” />  </reslutMap> |

### 讲述下Mybatis的一级、二级缓存？

|  |
| --- |
| 1）一级缓存: 基于 PerpetualCache 的 HashMap 本地缓存，其存储作用域为 Session，当 Session flush 或 close 之后，该 Session 中的所有 Cache 就将清空，默认打开一级缓存。  2）二级缓存与一级缓存其机制相同，默认也是采用 PerpetualCache，HashMap 存储，不同在于其存储作用域为 Mapper(Namespace)，并且可自定义存储源，如 Ehcache。默认不打开二级缓存，要开启二级缓存，使用二级缓存属性类需要实现Serializable序列化接口(可用来保存对象的状态),可在它的映射文件中配置<cache/> ；  3）对于缓存数据更新机制，当某一个作用域(一级缓存 Session/二级缓存Namespaces)的进行了C/U/D 操作后，默认该作用域下所有 select 中的缓存将被 clear。 |

### 12、请写出mybatis中常用的动态标签，并简单介绍其作用？

|  |
| --- |
| if 标签  if 标签通常用于 WHERE 语句、UPDATE 语句、INSERT 语句中，通过判断参数值来决定是否使用某个查询条件、判断是否更新某一个字段、判断是否插入某个字段的值。  foreach 标签  foreach 标签主要用于构建 in 条件，可在 sql 中对集合进行迭代。也常用到批量删除、添加等操作中。  choose 标签  有时候我们并不想应用所有的条件，而只是想从多个选项中选择一个。MyBatis 提供了 choose 元素，按顺序判断 when 中的条件出否成立，如果有一个成立，则 choose 结束。当 choose 中所有 when的条件都不满则时，则执行 otherwise 中的 sql。类似于 Java 的 switch 语句，choose 为 switch，when 为 case，otherwise 则为 default。  where 标签  当 if 标签较多时,如果标签返回的内容是以 AND 或 OR 开头的，则它会剔除掉.  set 标签  使用 set 标签可以将动态的配置 set关键字，和剔除追加到条件末尾的任何不相关的逗号。  trim 标签  [格式化输出，也可以通过 trim 标签设定或忽略前后缀来实现](https://blog.csdn.net/qq_39623058/article/details/88779301).   配置关联关系  collection 标签 配置一对多  association 标签 配置一对一  sql 标签  当多种类型的查询语句的查询字段或者查询条件相同时，可以将其定义为常量，方便调用。为求 <select> 结构清晰也可将 sql 语句分解。 |

### 描述下Ioc依赖注入，有哪些方式？

|  |
| --- |
| 构造器依赖注入：构造器依赖注入在容器触发构造器的时候完成，该构造器有一系列的参数，每个参数代表注入的对象。  Setter方法依赖注入：首先容器会触发一个无参构造函数或无参静态工厂方法实例化对象，之后容器调用bean中的setter方法完成Setter方法依赖注入。 |

### 说下对Spring面向切面编程(AOP)的理解？

|  |
| --- |
| 面向切面编程（AOP）：允许程序员模块化横向业务逻辑，或定义核心部分的功能，例如日志管理和事务管理。  切面(Aspect) ：AOP的核心就是切面，它将多个类的通用行为封装为可重用的模块。该模块含有一组API提供 cross-cutting功能。例如,日志模块称为日志的AOP切面。根据需求的不同，一个应用程序可以有若干切面。在Spring AOP中，切面通过带有@Aspect注解的类实现。  通知(Advice)：通知表示在方法执行前后需要执行的动作。实际上它是Spring AOP框架在程序执行过程中触发的一些代码。Spring切面可以执行一下五种类型的通知:  before(前置通知)：在一个方法之前执行的通知。  after(最终通知)：当某连接点退出的时候执行的通知（不论是正常返回还是异常退出）。  after-returning(后置通知)：在某连接点正常完成后执行的通知。  after-throwing(异常通知)：在方法抛出异常退出时执行的通知。  around(环绕通知)：在方法调用前后触发的通知。  切入点(Pointcut)：切入点是一个或一组连接点，通知将在这些位置执行。可以通过表达式或匹配的方式指明切入点。  引入：引入允许我们在已有的类上添加新的方法或属性。  目标对象：被一个或者多个切面所通知的对象。它通常是一个代理对象。也被称做被通知（advised）对象。  代理：代理是将通知应用到目标对象后创建的对象。从客户端的角度看，代理对象和目标对象是一样的。有以下几种代理：  BeanNameAutoProxyCreator：bean名称自动代理创建器  DefaultAdvisorAutoProxyCreator：默认通知者自动代理创建器  Metadata autoproxying：元数据自动代理织入：将切面和其他应用类型或对象连接起来创建一个通知对象的过程。织入可以在编译、加载或运行时完成。 |

### 解释自动装配的各种模式？

|  |
| --- |
| 自动装配提供五种不同的模式供Spring容器用来自动装配beans之间的依赖注入:  no：默认的方式是不进行自动装配，通过手工设置ref 属性来进行装配bean。  byName：通过参数名自动装配，Spring容器查找beans的属性，这些beans在XML配置文件中被设置为byName。之后容器试图匹配、装配和该bean的属性具有相同名字的bean。  byType：通过参数的数据类型自动自动装配，Spring容器查找beans的属性，这些beans在XML配置文件中被设置为byType。之后容器试图匹配和装配和该bean的属性类型一样的bean。如果有多个bean符合条件，则抛出错误。  constructor：这个同byType类似，不过是应用于构造函数的参数。如果在BeanFactory中不是恰好有一个bean与构造函数参数相同类型，则抛出一个严重的错误。  autodetect：如果有默认的构造方法，通过 construct的方式自动装配，否则使用 byType的方式自动装配。 |

### 解释Spring框架中bean的生命周期？

|  |
| --- |
| 1.首先容器启动后，会对scope为singleton且非懒加载的bean进行实例化，  2.按照Bean定义信息配置信息，注入所有的属性，  3.如果Bean实现了BeanNameAware接口，会回调该接口的setBeanName()方法，传入该Bean的id，此时该Bean就获得了自己在配置文件中的id，  4.如果Bean实现了BeanFactoryAware接口,会回调该接口的setBeanFactory()方法，传入该Bean的BeanFactory，这样该Bean就获得了自己所在的BeanFactory，  5.如果Bean实现了ApplicationContextAware接口,会回调该接口的setApplicationContext()方法，传入该Bean的ApplicationContext，这样该Bean就获得了自己所在的ApplicationContext，  6.如果有Bean实现了BeanPostProcessor接口，则会回调该接口的postProcessBeforeInitialzation()方法，  7.如果Bean实现了InitializingBean接口，则会回调该接口的afterPropertiesSet()方法，  8.如果Bean配置了init-method方法，则会执行init-method配置的方法，  9.如果有Bean实现了BeanPostProcessor接口，则会回调该接口的postProcessAfterInitialization()方法，  10.经过流程9之后，就可以正式使用该Bean了,对于scope为singleton的Bean,Spring的ioc容器中会缓存一份该bean的实例，而对于scope为prototype的Bean,每次被调用都会new一个新的对象，期生命周期就交给调用方管理了，不再是Spring容器进行管理了  11.容器关闭后，如果Bean实现了DisposableBean接口，则会回调该接口的destroy()方法，  12.如果Bean配置了destroy-method方法，则会执行destroy-method配置的方法，至此，整个Bean的生命周期结束 |

### 说一下Spring中支持的bean作用域？

|  |
| --- |
| Spring框架支持如下五种不同的作用域：  singleton：在Spring IOC容器中仅存在一个Bean实例，Bean以单实例的方式存在。  prototype：一个bean可以定义多个实例。  request：每次HTTP请求都会创建一个新的Bean。该作用域仅适用于WebApplicationContext环境。  session：一个HTTP Session定义一个Bean。该作用域仅适用于WebApplicationContext环境。  globalSession：同一个全局HTTP Session定义一个Bean。该作用域同样仅适用于WebApplicationContext环境。bean默认的scope属性是"singleton"。 |

### SpringMVC怎么样设定重定向和转发的？

|  |
| --- |
| （1）转发：在返回值前面加"forward:"，譬如"forward:user.do?name=method4"  （2）重定向：在返回值前面加"redirect:"，譬如"redirect:<http://www.baidu.com>" |

### 说下SpringMvc怎么和AJAX相互调用的？

|  |
| --- |
| 通过Jackson框架就可以把Java里面的对象直接转化成Js可以识别的Json对象。具体步骤如下 ：  （1）加入Jackson.jar  （2）在配置文件中配置json的映射  （3）在接受Ajax方法里面可以直接返回Object,List等,但方法前面要加上@ResponseBody注解。 |

### 说出springmvc常用的注解有哪些，并简单描述作用？

|  |
| --- |
| @RequestMapping：用于处理请求 url 映射的注解，可用于类或方法上。用于类上，则表示类中的所有响应请求的方法都是以该地址作为父路径。  @RequestBody：注解实现接收http请求的json数据，将json转换为java对象。  @ResponseBody：注解实现将conreoller方法返回对象转化为json对象响应给客户。 @Resource和@Autowired @Resource和@Autowired都是做bean的注入时使用，其实@Resource并不是Spring的注解，它的包是javax.annotation.Resource，需要导入，但是Spring支持该注解的注入。  　　1、共同点  　　　　两者都可以写在字段和setter方法上。两者如果都写在字段上，那么就不需要再写setter方法。  　　2、不同点  　　（1）@Autowired  　　　　@Autowired为Spring提供的注解，需要导入包org.springframework.beans.factory.annotation.Autowired;只按照byType注入。 @PathVariable 　　用于将请求URL中的模板变量映射到功能处理方法的参数上，即取出uri模板中的变量作为参数 @RequestParam 　　@RequestParam用于将请求参数区数据映射到功能处理方法的参数上 |

### 说出springmvc的加载流程？

|  |
| --- |
| 1. 用户发送请求至前端控制器DispatcherServlet； 2. DispatcherServlet收到请求后，调用HandlerMapping处理器映射器，请求获取Handle； 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 调用 HandlerAdapter处理器适配器； 5. HandlerAdapter 经过适配调用 具体处理器(Handler，也叫后端控制器)； 6. Handler执行完成返回ModelAndView； 7. HandlerAdapter将Handler执行结果ModelAndView返回给DispatcherServlet； 8. DispatcherServlet将ModelAndView传给ViewResolver视图解析器进行解析； 9. ViewResolver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。 |

### 如果前台有很多个参数传入,并且这些参数都是一个对象的,那么springmvc怎么样快速得到这个对象？

|  |
| --- |
| 直接在方法中声明这个对象,SpringMvc就自动会把属性赋值到这个对象里面 |

### Springmvc的返回值类型有哪些，具体说出一个使用场景？

|  |
| --- |
| 有String, ModelAndView。ModelAndView类把视图和数据都合并的一起的，但一般用String比较好。 |

### Sql题，有以下4张表









请写出以下sql的查询语句

1. 查询Score表中成绩在60到80之间的所有记录

|  |
| --- |
| select \* from Score where Degree between 60 and 80 |

1. 查询Score表中的最高分的学生学号和课程号

|  |
| --- |
| select SNO,CNO from Score where Degree=(select MAX(Degree) from Score) |

1. 查询成绩高于学号为“109”、课程号为“3-105”的成绩的所有记录

|  |
| --- |
| select \* from student,Score where student.Sno=Score.Sno and Score.Degree>(select Degree from Score where Cno=‘3-105’ and Sno=‘109’) |

1. 查询Student表中不姓“王”的同学记录

|  |
| --- |
| select \* from student where Sname not like (‘王%’) |

1. 以班号降序和年龄升序的顺序查询Student表中的全部记录

|  |
| --- |
| select \* from student order by Class desc,Sbirthday asc |

### 请描述转发和重定向的区别？

|  |
| --- |
| 1、请求次数： 转发一次，重定向两次  2、浏览器地址：转发不变，重定向改变  3、使用request域共享数据：转发是一次请求可以共享数据，重定向浏览器发起两次请求，不能共享数据  4、相对路径： 转发地址不变会造成转发后的页面中的相对位置发生改变引起相对路径失效，重定向不会  5、效率：转发浏览器一次请求效率高，重定向效率低  6、WEB-INF下资源：转发可以访问，重定向不可以 |

### Vue中created和mounted的区别？

|  |
| --- |
| created:在模板渲染成html前调用，即通常初始化某些属性值，然后再渲染成视图。  mounted:在模板渲染成html后调用，通常是初始化页面完成后，再对html的dom节点进行一些需要的操作。 |

### 请描述get请求 和 post请求的区别？

|  |
| --- |
| ① 浏览器和表单的默认提交方式是get，get请求效率比post高  ② get请求参数在url地址后拼接，所以有以下特点：  请求报文没有请求体  少了和请求体相关的请求头参数  参数在url地址中拼接，上传参数大小有限制，不能用来上传文件，相对post请求不安全  ③ post请求参数在请求报文的请求体中携带，有以下特点：  请求报文有请求体，相对安全  请求头多了和请求体相关的参数  请求体数据没有大小限制可以用来上传文件 |

### 第一次页面加载会触发哪几个钩子？

|  |
| --- |
| beforeCreate， created， beforeMount， mounted |

### 什么是 Thymeleaf？

|  |
| --- |
| Thymeleaf 是一个支持原生 HTML 文件的 Java 模版引擎，可以实现前后端分离的交互方式，即视图与业务数据分开响应，它可以直接将服务端返回的数据生成 HTML 格式，同时也可以处理 XML、JavaScript、CSS 等格式。 |

### 说出几种vue当中的指令和它的用法？

|  |
| --- |
| v-model双向数据绑定；  v-for循环；  v-if v-show 显示与隐藏；  v-on事件；v-once: 只绑定一次。 |

### 请解释Spring Bean的生命周期？

|  |
| --- |
| 1.默认情况下，IOC容器中bean的生命周期分为五个阶段:   1. 调用构造器 或者是通过工厂的方式创建Bean对象 2. 给bean对象的属性注入值 3. 调用初始化方法，进行初始化， 初始化方法是通过init-method来指定的. 4. 使用 5. IOC容器关闭时， 销毁Bean对象.   2.当加入了Bean的后置处理器后，IOC容器中bean的生命周期分为七个阶段:  ① 调用构造器 或者是通过工厂的方式创建Bean对象  ② 给bean对象的属性注入值  ③ 执行Bean后置处理器中的 postProcessBeforeInitialization   1. 调用初始化方法，进行初始化， 初始化方法是通过init-method来指定的. 2. 执行Bean的后置处理器中 postProcessAfterInitialization 3. 使用 4. IOC容器关闭时， 销毁Bean对象 |

### 请描述MyBatis中 #{}和${}的区别是什么？

|  |
| --- |
| #{}是预编译处理，${}是字符串替换。  Mybatis在处理#{}时，会将sql中的#{}替换为?号，调用PreparedStatement的set方法来赋值；  Mybatis在处理${}时，就是把${}替换成变量的值。  使用#{}可以有效的防止SQL注入，提高系统安全性。 |

### Mybatis 结果集的映射方式有几种，并分别解释每种映射方式如何使用？

|  |
| --- |
| 自动映射 ，通过resultType来指定要映射的类型即可。  自定义映射 通过resultMap来完成具体的映射规则，指定将结果集中的哪个列映射到对象的哪个属性。 |

### 简述Springmvc中ContextLoaderListener的作用以及实现原理？

|  |
| --- |
| 1. 作用:ContextLoaderListener的作用是通过监听的方式在WEB应用服务器启动时将Spring的容器对象进行初始化. 2. 原理: ContextLoaderListener 实现了ServletContextListener接口，用于监听   ServletContext的创建，当监听到ServletContext创建时，在对应contextInitialized  方法中，将Spring的容器对象进行创建，并将创建好的容器对象设置到ServletContext域对象中，  目的是让各个组件可以通过ServletContext共享到Spring的容器对象 |

### 简述Spring声明式事务中@Transaction中常用的两种事务传播行为？

|  |
| --- |
| 通过propagation来执行事务的传播行为  REQUIRED: 使用调用者的事务，如果调用者没有事务，则启动新的事务运行  REQUIRES\_NEW: 将调用者的事务挂起，开启新的事务运行。 |

### 简述SpringMVC 的工作原理？

|  |
| --- |
| （1）用户向服务器发送请求，请求被springMVC 前端控制器 DispatchServlet 捕获；  （2）DispatcherServle 对请求 URL 进行解析，得到请求资源标识符（URL），然后根据该 URL 调用 HandlerMapping将请求映射到处理器 HandlerExcutionChain；  （3）DispatchServlet 根据获得 Handler 选择一个合适的HandlerAdapter 适配器处理；  （4）Handler 对数据处理完成以后将返回一个 ModelAndView（）对象给 DisPatchServlet;  （5）Handler 返回的 ModelAndView() 只是一个逻辑视图并不是一个正式的视图， DispatcherSevlet 通过ViewResolver 试图解析器将逻辑视图转化为真正的视图View;  （6）DispatcherServle 通过 model 解析出 ModelAndView()中的参数进行解析最终展现出完整的 view 并返回给客户端; |

### 简述Mybatis提供的两级缓存，以及缓存的查找顺序？

|  |
| --- |
| MyBatis的缓存分为一级缓存和 二级缓存。  一级缓存是SqlSession级别的缓存，默认开启。  二级缓存是NameSpace级别(Mapper)的缓存，多个SqlSession可以共享，使用时需要进行配置开启。  缓存的查找顺序: 二级缓存 一级缓存 数据库 |

### 简述REST中 HiddenHttpMethodFilter过滤器的作用？

|  |
| --- |
| 该过滤器主要负责转换客户端请求的方式，当浏览器的请求方式为POST,并且在请求中能通过 \_method获取到请求参数值,  该过滤器就会进行请求方式的转换.  一般在REST中，都是将POST请求转换为对应的DELETE 或者是PUT |

### 简述Springmvc中处理模型数据的两种方式？

|  |
| --- |
| 1. 使用ModelAndView 作为方法的返回值，将 模型数据 和 视图信息封装到ModelAndView中 2. 使用Map 或者是Model 作为方法的形参. 将模型数据添加到Map或者是Model中. |

### 简述Springmvc中如何返回JSON数据？

|  |
| --- |
| 1. 在工程最终加入jackson的jar包 2. 在请求处理方法中，将返回值改为具体返回的数据的类型， 例如 数据的集合类型 List<Employee>等 3. 在请求处理方法上使用@ResponseBody注解 |

### Springmvc中如何解决POST请求中文乱码问题？

|  |
| --- |
| Springmvc中通过CharacterEncodingFilter解决中文乱码问题.  在web.xml中加入：  <filter>  <filter-name>CharacterEncodingFilter</filter-name>  <filter-class>org.springframework.web.filter.CharacterEncodingFilter</filter-class>  <init-param>  <param-name>encoding</param-name>  <param-value>utf-8</param-value>  </init-param>  </filter>  <filter-mapping>  <filter-name>CharacterEncodingFilter</filter-name>  <url-pattern>/\*</url-pattern>  </filter-mapping> |

### 请解释@Autowired注解的工作机制及required属性的作用？

|  |
| --- |
| 首先会使用byType的方式进行自动装配，如果能唯一匹配，则装配成功，  如果匹配到多个兼容类型的bean, 还会尝试使用byName的方式进行唯一确定.  如果能唯一确定，则装配成功，如果不能唯一确定，则装配失败，抛出异常.    默认情况下， 使用@Autowired标注的属性必须被装配，如果装配不了，也会抛出异常.  可以使用required=false来设置不是必须要被装配. |

### 请描述一下Spring的事务管理？

|  |
| --- |
| 1、声明式事务管理的定义：用在 Spring 配置文件中声明式的处理事务来代替代码式的处理事务。这样的好处是，事务管理不侵入开发的组件，具体来说，业务逻辑对象就不会意识到正在事务管理之中，事实上也应该如此，因为事务管理是属于系统层面的服务，而不是业务逻辑的一部分，如果想要改变事务管理策划的话，也只需要在定义文件中重新配置即可，这样维护起来极其方便。  基于 TransactionInterceptor 的声明式事务管理：两个次要的属性： transactionManager，用来指定一个事务治理器， 并将具体事务相关的操作请托给它； 其他一个是 Properties 类型的transactionAttributes 属性，该属性的每一个键值对中，键指定的是方法名，方法名可以行使通配符， 而值就是表现呼应方法的所运用的事务属性。   1. 基于 @Transactional 的声明式事务管理：Spring 2.x 还引入了基于 Annotation 的体式格式，具体次要触及@Transactional 标注。@Transactional 可以浸染于接口、接口方法、类和类方法上。算作用于类上时，该类的一切public 方法将都具有该类型的事务属性。   编程式事物管理的定义：在代码中显式挪用 beginTransaction()、commit()、rollback()等事务治理相关的方法， 这就是编程式事务管理。Spring 对事物的编程式管理有基于底层 API 的编程式管理和基于 TransactionTemplate 的编程式事务管理两种方式。 |

### 20、简述Spring与Mybatis整合时，主要整合的两个地方？

|  |
| --- |
| SqlSession创建的问题，通过SqlSessionFactoryBean来配置用于创建SqlSession的信息。例如: Mybatis的核心配置文件、Mapper映射文件、数据源等  Mapper接口创建的问题， 使用MapperScannerConfigurer批量为MyBatis的Mapper接口生成代理实现类并将具体的对象交给Spring容器管理 |

Sql题：

1. 查询Score表中成绩在60到80之间的所有记录

|  |
| --- |
| select \* from Score where Degree between 60 and 80 |

1. 查询Score表中的最高分的学生学号和课程号

|  |
| --- |
| select SNO,CNO from Score where Degree=(select MAX(Degree) from Score) |

1. 查询成绩高于学号为“109”、课程号为“3-105”的成绩的所有记录

|  |
| --- |
| select \* from student,Score where student.Sno=Score.Sno and Score.Degree>(select Degree from Score where Cno=‘3-105’ and Sno=‘109’) |

1. 查询Student表中不姓“王”的同学记录

|  |
| --- |
| select \* from student where Sname not like (‘王%’) |

1. 以班号和年龄从大到小的顺序查询Student表中的全部记录

|  |
| --- |
| select \* from student order by Class desc,Sbirthday asc |

### 谈谈你对vue中MVVM开发模式的理解？

|  |
| --- |
| MVVM分为Model、View、ViewModel三者。  Model：代表数据模型，数据和业务逻辑都在Model层中定义；  View：代表UI视图，负责数据的展示；  ViewModel：负责监听Model中数据的改变并且控制视图的更新，处理用户交互操作；         Model和View并无直接关联，而是通过ViewModel来进行联系的，Model和ViewModel之间有着双向数据绑定的联系。因此当Model中的数据改变时会触发View层的刷新，View中由于用户交互操作而改变的数据也会在Model中同步。  这种模式实现了Model和View的数据自动同步，因此开发者只需要专注对数据的维护操作即可，而不需要自己操作dom。 |

### 请说下Vue 有哪些指令，并简述其作用（5个以上）？

|  |
| --- |
| v-html：用于渲染HTML标签  v-show：用于类似双大括号语法渲染数据  v-if：渲染数据的时候，也可以用于条件判断  v-for：遍历集合或者数组，用于页面渲染数据  v-bind:绑定属性，注意冒号后面跟标签的属性，属性后面的等号指向数据，它可以简写为 :class, :href。  V-model：数据双向绑定 |

### 请说下在后台Servlet代码中如何获取前端form表单提交的属性？

|  |
| --- |
| 可以使用request.getParameter();  也可以使用request.getParameterMap(),在使用beanutils.populate()方法 |

### 请简述下Cookie和Session的区别？

|  |
| --- |
| cookie数据保存在客户端，session数据保存在服务器端  区别在于session是放在服务器上的，过期与否取决于服务期的设定，cookie是存在客户端的，过去与否可以在cookie生成的时候设置进去。  cookie数据存放在客户的浏览器上，session数据放在服务器上。  cookie不是很安全，别人可以分析存放在本地的COOKIE并进行COOKIE欺骗,如果主要考虑到安全应当使用session ；  session会在一定时间内保存在服务器上。当访问增多，会比较占用你服务器的性能，如果主要考虑到减轻服务器性能方面，应当使用COOKIE  cookie和session的共同之处在于：cookie和session都是用来跟踪浏览器用户身份的会话方式。 |

### 请描述请求转发和重定向的区别？

|  |
| --- |
| 转发使用的是getRequestDispatcher()方法;重定向使用的是sendRedirect();  转发：浏览器URL的地址栏不变。重定向：浏览器URL的地址栏改变；  转发是服务器行为，重定向是客户端行为；  转发是浏览器只做了一次访问请求。重定向是浏览器做了至少两次的访问请求；  转发2次跳转之间传输的信息不会丢失，重定向2次跳转之间传输的信息会丢失（request范围）；  重定向的速度比转发慢，因为浏览器还得发出一个新的请求，如果在使用转发和重定向都无所谓的时候建议使用转发。  因为转发只能访问当前WEB的应用程序，所以不同WEB应用程序之间的访问，特别是要访问到另外一个WEB站点上的资源的情况，这个时候就只能使用重定向了。 |

### Vue中第一次页面加载会触发哪几个钩子函数？

|  |
| --- |
| beforeMount vue实例的$el和data都初始化了，但还是虚拟的dom节点，具体的data.filter还未替换。  mounted vue实例挂载完成，data.filter成功渲染。  beforeCreadted vue实例的挂载元素$el和数据对象data都为undefined，还未初始化。  created vue实例的数据对象data有了，$el还没有 |

### 简述mybatis中#{}和${}的区别？

|  |
| --- |
| ${}是Properties文件中的变量占位符，它可以用于标签属性值和sql内部，属于静态文本替换，比如${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()。  #{}是预编译处理，${}是字符串替换。  Mybatis在处理#{}时，会将sql中的#{}替换为?号，调用PreparedStatement的set方法来赋值；  Mybatis在处理${}时，就是把${}替换成变量的值。  使用#{}可以有效的防止SQL注入，提高系统安全性。 |

### Mybatis 结果集的映射方式有几种，并分别解释每种映射方式如何使用？

|  |
| --- |
| 自动映射 ，通过resultType来指定要映射的类型即可。  自定义映射 通过resultMap来完成具体的映射规则，指定将结果集中的哪个列映射到对象的哪个属性。 |

### 简述Mybatis提供的两级缓存，以及缓存的查找顺序？

|  |
| --- |
| 1）一级缓存: 基于 PerpetualCache 的 HashMap 本地缓存，其存储作用域为 Session，当 Session flush 或 close 之后，该 Session 中的所有 Cache 就将清空，默认打开一级缓存。  2）二级缓存与一级缓存其机制相同，默认也是采用 PerpetualCache，HashMap 存储，不同在于其存储作用域为 Mapper(Namespace)，并且可自定义存储源，如 Ehcache。默认不打开二级缓存，要开启二级缓存，使用二级缓存属性类需要实现Serializable序列化接口(可用来保存对象的状态),可在它的映射文件中配置<cache/> ；  3）对于缓存数据更新机制，当某一个作用域(一级缓存 Session/二级缓存Namespaces)的进行了C/U/D 操作后，默认该作用域下所有 select 中的缓存将被 clear。 |

### Xml映射文件中，除了常见的select|insert|updae|delete标签之外，还有哪些标签？

|  |
| --- |
| if 标签  if 标签通常用于 WHERE 语句、UPDATE 语句、INSERT 语句中，通过判断参数值来决定是否使用某个查询条件、判断是否更新某一个字段、判断是否插入某个字段的值。  foreach 标签  foreach 标签主要用于构建 in 条件，可在 sql 中对集合进行迭代。也常用到批量删除、添加等操作中。  choose 标签  有时候我们并不想应用所有的条件，而只是想从多个选项中选择一个。MyBatis 提供了 choose 元素，按顺序判断 when 中的条件出否成立，如果有一个成立，则 choose 结束。当 choose 中所有 when的条件都不满则时，则执行 otherwise 中的 sql。类似于 Java 的 switch 语句，choose 为 switch，when 为 case，otherwise 则为 default。  where 标签  当 if 标签较多时,如果标签返回的内容是以 AND 或 OR 开头的，则它会剔除掉.  set 标签  使用 set 标签可以将动态的配置 set关键字，和剔除追加到条件末尾的任何不相关的逗号。  trim 标签  [格式化输出，也可以通过 trim 标签设定或忽略前后缀来实现](https://blog.csdn.net/qq_39623058/article/details/88779301).   配置关联关系  collection 标签 配置一对多  association 标签 配置一对一  sql 标签  当多种类型的查询语句的查询字段或者查询条件相同时，可以将其定义为常量，方便调用。为求 <select> 结构清晰也可将 sql 语句分解。 |

### 当实体类中的属性名和表中的字段名不一样 ，怎么办 ？

|  |
| --- |
| 第1种： 通过在查询的sql语句中定义字段名的别名，让字段名的别名和实体类的属性名一致。  第2种： 通过<resultMap>来映射字段名和实体类属性名的一一对应的关系 |

### 谈谈你对spring框架的深入理解？

|  |
| --- |
| Spring一款容器框架，其核心是IOC和AOP  （1）IOC就是控制反转，指创建对象的控制权转移给Spring框架进行管理，并由Spring根据配置文件去创建实例和管理各个实例之间的依赖关系，对象与对象之间松散耦合，也利于功能的复用。DI依赖注入，和控制反转是同一个概念的不同角度的描述，即 应用程序在运行时依赖IoC容器来动态注入对象需要的外部依赖。  （2）最直观的表达就是，以前创建对象的主动权和时机都是由自己把控的，IOC让对象的创建不用去new了，可以由spring自动生产，使用java的反射机制，根据配置文件在运行时动态的去创建对象以及管理对象，并调用对象的方法的。  （3）AOP，一般称为面向切面，作为面向对象的一种补充，用于将那些与业务无关，但却对多个对象产生影响的公共行为和逻辑，抽取并封装为一个可重用的模块，这个模块被命名为“切面”（Aspect），减少系统中的重复代码，降低了模块间的耦合度，提高系统的可维护性。可用于权限认证、日志、事务处理。  AOP实现的关键在于 代理模式，AOP代理主要分为静态代理和动态代理。静态代理的代表为AspectJ；动态代理则以Spring AOP为代表。 |

### 请解释Spring Bean的生命周期？

|  |
| --- |
| 1.默认情况下，IOC容器中bean的生命周期分为五个阶段:  调用构造器 或者是通过工厂的方式创建Bean对象  给bean对象的属性注入值  调用初始化方法，进行初始化， 初始化方法是通过init-method来指定的.  使用  IOC容器关闭时， 销毁Bean对象.  2.当加入了Bean的后置处理器后，IOC容器中bean的生命周期分为七个阶段:  ① 调用构造器 或者是通过工厂的方式创建Bean对象  ② 给bean对象的属性注入值  ③ 执行Bean后置处理器中的 postProcessBeforeInitialization  调用初始化方法，进行初始化， 初始化方法是通过init-method来指定的.  执行Bean的后置处理器中 postProcessAfterInitialization  使用  IOC容器关闭时， 销毁Bean对象 |

### 请简述spring bean的作用域有哪些？

|  |
| --- |
| （1）singleton：默认作用域，单例bean，每个容器中只有一个bean的实例。  （2）prototype：为每一个bean请求创建一个实例。  （3）request：为每一个request请求创建一个实例，在请求完成以后，bean会失效并被垃圾回收器回收。  （4）session：与request范围类似，同一个session会话共享一个实例，不同会话使用不同的实例。  （5）global-session：全局作用域，所有会话共享一个实例。如果想要声明让所有会话共享的存储变量的话，那么这全局变量需要存储在global-session中。 |

### 说一下你对spring框架中通知类型的理解？

|  |
| --- |
| （1）前置通知（Before Advice）：在连接点（Join point）之前执行的通知。  （2）后置通知（After Advice）：当连接点退出的时候执行的通知（不论是正常返回还是异常退出）。  （3）环绕通知（Around Advice）：包围一个连接点的通知，这是最强大的一种通知类型。 环绕通知可以在方法调用前后完成自定义的行为。它也可以选择是否继续执行连接点或直接返回它们自己的返回值或抛出异常来结束执行。  （4）返回后通知（AfterReturning Advice）：在连接点正常完成后执行的通知（如果连接点抛出异常，则不执行）  抛出异常后通知（AfterThrowing advice）：在方法抛出异常退出时执行的通知 |

### 简述springmvc的运行流程？

|  |
| --- |
| 1.发送请求到前端控制器（dispatcherservlet）  2.前端控制器（dispatcherservlet）接收到请求后，根据URL调用HandlerMapping获得该handler配置的所有相关的对象，最后以HandlerExecutionChain对象的形式返回  3.dispatchersevlet根据获得的handler，选择一个合适的handlerAdapter  4.提取request中的额数据模型，填充handler入参，开始执行handler（也就是cotroller）  5.handler执行完成后，向dispatcherservlet返回一个modelandview对象  6.dispatcherservlet接收到modelandview对象，将modelandview对象传给viewresolve  7.viewresolver解析后返回具体的view  8.dispatcherservlet将渲染视图，返回给客户端 |

### 请说出springmvc中常用的注解，并阐明其作用？

|  |
| --- |
| 1. @Controller，使用它标记的类就是一个SpringMVC Controller 对象  2. @RequestMapping，处理请求映射地址  3. @PathVariable，用于对应restful风格url中的参数  4. @RequestParam，将请求的参数绑定到方法中的参数上  5. @ResponseBody，将返回类型直接输入到http response body中  6. @RequestBody，方法参数直接被绑定到http request body中 |

### 如果想在拦截的方法里面得到从前台传入的普通参数,怎么得到？

|  |
| --- |
| 直接在形参里面声明这个参数就可以,但必须名字和传过来的参数一样。 |

### 如果前台有很多个参数传入,并且这些参数都是一个对象的,那么怎么样快速得到这个对象？

|  |
| --- |
| 直接在方法中声明这个对象,SpringMvc就自动会把属性赋值到这个对象里面。 |

### 如何在spring mvc实现RESTful 服务？

|  |
| --- |
| 1.导入jackson2包  2.开启注解驱动[mvc:annotation-driven/](https://www.liangzl.com/mvc:annotation-driven/) |

### @RequestMapping注解用在类上面有什么作用？

|  |
| --- |
| 它是一个用来处理请求地址映射的注解，可用于类或方法上。  用于类上，表示类中的所有响应请求的方法都是以该地址作为父路径。 |

### SpringMVC怎么样设定重定向和转发的？

|  |
| --- |
| （1）在返回值前面加"forward:"就可以让结果转发,譬如"forward:user?name=method4"  （2）在返回值前面加"redirect:"就可以让返回值重定向,譬如"redirect:http://www.atguigu.com" |

### Sql题，有以下4张表（每题2分）









请写出以下sql的查询语句

1. 查询Score表中成绩在60到80之间的所有记录

|  |
| --- |
| select \* from Score where Degree between 60 and 80 |

1. 查询Score表中的最高分的学生学号和课程号

|  |
| --- |
| select SNO,CNO from Score where Degree=(select MAX(Degree) from Score) |

1. 查询成绩高于学号为“109”、课程号为“3-105”的成绩的所有记录

|  |
| --- |
| select \* from student,Score where student.Sno=Score.Sno and Score.Degree>(select Degree from Score where Cno=‘3-105’ and Sno=‘109’) |

1. 查询Student表中不姓“王”的同学记录

|  |
| --- |
| select \* from student where Sname not like (‘王%’) |

1. 以班号和年龄从大到小的顺序查询Student表中的全部记录

|  |
| --- |
| select \* from student order by Class desc,Sbirthday asc |