



## 第二版：Java 综合 84 道

### 目录

第二版：Java 综合 84 道	1
1.Javascript 中常用的事件有哪些？	4
2.js 如何实现页面刷新呢？	4
3.页面前进或者后退	4
4.Js 如何跳转到一个指定页面	5
5.使用 js 获取一个表单元素	5
6.如何阻止表单提交	5
7.正则表达式有那些符号？	5
8.Sql 中 delete 与 truncate 的区别	6
9.Xml 的 java 解析有几种方式？	6
10.Sql 优化有那些方法？	6
11.Json 是什么？	6
12.如何创建一个 json 对象？	7
13.如何解析 json 对象？	7
14.聚集索引与非聚集索引有什么区别？	7
15.一个类文件中能否有多个类？有什么要求？	7
16.你知道有哪些开源框架？	8
17.什么是事务？事务有那些特点？	8
18.事务的使用场景在什么地方？	8
19.Js 如何实现动态效果？	8
20.Jsp 由哪些内容组成？	9
21.Jsp 包含那些隐藏对象或者内建对象	9
22.如何使用 exception 对象？	9
23.Get 请求与 post 有什么区别？	10
24.计算机网络有几层？	10
25.常见的计算机网络协议有那些？	10
26.Tcp 协议的特点	10
27.Java 网络编程有几种？	11



28.TCP 编程与 UDP 编程有什么区别?	11
29.创建 socket 通讯的步骤?	11
30.Java 中如何实现多线程	11
31.Java 中 Synchronized 关键字的使用?	12
32.什么是线程池?	12
33.什么是游标?	13
34.游标的创建步骤?	13
35.在做文件上传的时候, form 表单的 enctype 的指是什么?	13
Html 中 a 标签的 target 属性有哪些值?	13
36.当打开其他程序的网页时, 使用的 target 属性是哪个?	14
37.Servlet 的生命周期?	14
38.Servlet 生命周期内调用的方法过程?	14
39.线程的生命周期?	14
40.Statement 与 preparedStatement 区别	15
41.Session 与 cookie 的区别?	15
42.存储过程与函数的区别	15
43.会话跟踪技术有那些?	16
44.jspServlet 中通信作用域有那些?	16
45.url 是什么? 由哪些部分组成?	16
46.你所知道的 web 服务器有哪些?	16
47.如何部署一个 web 项目?	16
48.如何修改 tomcat 的端口号?	17
49.重定向和请求转发的区别?	17
50.上传文件是如何做的?	17
51.HTTP 的状态码	18
200: 请求成功	18
400: 不是正确的请求, 大多情况下表示参数错误	18
404: 找不到请求资源	18
500: 服务器内部错误	18
403: 服务器拒绝	18
405: 请求的 method 不支持	18



52.Jsp 指令有那些？	18
53.如何使 session 失效	18
54.如何在jsp 页面上显示一些特定格式的数字或者日期	18
55.什么是线程异步？什么是线程同步？	18
56.什么是同步任务？什么是异步任务？	19
57.什么是过滤器？怎么创建一个过滤器	19
58.Int 与 integer 的区别	19
59.如何进行单元测试	19
60.使用 sql 写出一个分页程序？	19
61.监听器有哪些作用和用法？	20
62.JSP 中的静态包含和动态包含有什么区别？	20
63.Servlet 中如何获取用户提交的查询参数或表单数据？	20
64.如何设置请求的编码以及响应内容的类型？	21
65.什么是 Web Service (Web 服务)	21
66.什么是 ORM？	21
67.Hibernate 中 SessionFactory 是线程安全的吗？Session 是线程安全的吗（两个线程能够共享同一个 Session 吗）？	22
68.Hibernate 中 Session 的 load 和 get 方法的区别是什么？	22
69.Hibernate 的对象有几种状态	23
70.Session 的 save()、update()、merge()、lock()、saveOrUpdate()和 persist()方法分别是做什么的？有什么区别？	23
71.Session 加载实体对象的过程。	24
72.Query 接口的 list 方法和 iterate 方法有什么区别？	24
73.如何理解 Hibernate 的延迟加载机制？在实际应用中，延迟加载与 Session 关闭的矛盾是如何处理的？	25
74.简述 Hibernate 常见优化策略。	26
75.谈一谈 Hibernate 的一级缓存、二级缓存和查询缓存。	26
76.MyBatis 中使用#和\$书写占位符有什么区别？	27
77.什么是 IoC 和 DI？DI 是如何实现的？	27
78.Spring 中自动装配的方式有哪些？	28
79.解释一下什么叫 AOP (面向切面编程)？	28



80.Spring 中 Bean 的作用域有哪些? .....	28
81.Spring 中如何使用注解来配置 Bean? 有哪些相关的注解? .....	28
82.Spring 支持的事务管理类型有哪些? 你在项目中使用哪种方式? .....	29
83.Spring MVC 的工作原理是怎样的? .....	29
84.阐述 Spring 框架中 Bean 的生命周期? .....	30

我们的网站: <https://tech.souyunku.com>

关注我们的公众号: **搜云库技术团队**, **回复以下关键字**

回复: **【进群】** 邀请您进「**技术架构分享群**」

回复: **【内推】** 即可进: 北京, 上海, 广周, 深圳, 杭州, 成都, 武汉, 南京, 郑州, 西安, 长沙「**程序员工作内推群**」

回复 **【1024】** 送 4000G 最新架构师视频

回复 **【PPT】** 即可无套路获取, 以下最新整理调优 PPT!

## 46 页《JVM 深度调优, 演讲 PPT》



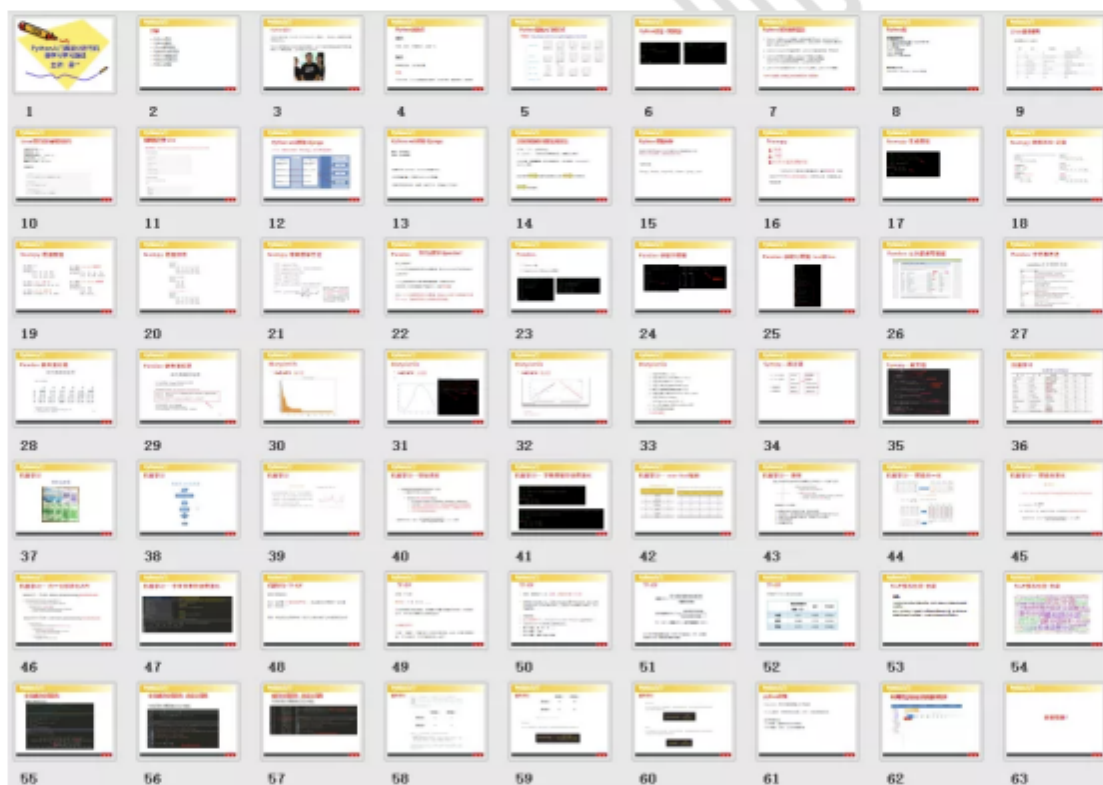
## 53 页《Elasticsearch 调优演讲 PPT》

微信搜一搜

搜云库技术团队



## 63 页《Python 数据分析入门 PPT》







微信扫一扫

<https://tech.souyunku.com>

技术、架构、资料、工作、内推  
专注于分享最有价值的互联网技术干货文章

## 1. Javascript 中常用的事件有哪些？

Onsubmit: 提交

Onblur: 失去焦点

Onclick: 单击

Onload: 加载页面

Onchange: 内容改变

onMouseMove: 鼠标移动

onMouseOver: 鼠标经过

onMouseOut: 鼠标移出

onselect: 下拉选项被选中

## 2.js 如何实现页面刷新呢？

history.go(0)

location.reload()

## 3.页面前进或者后退



前进

```
history.go(1)
history.forward
```

后退

```
History.go(-1)
History.back
```

## 4.Js 如何跳转到到一个指定页面

```
Location.href=url
History.go(url)
```

## 5.使用 js 获取一个表单元素

```
Document.getElementById()
Document.getElementsByName()
Document.getElementsByTagName()
```

## 6.如何阻止表单提交

Onsubmit= "return false"

## 7.正则表达式有那些符号?



\$: 匹配字符串结束的位置

^: 匹配字符串开始的位置

\*: 匹配零次或者多次

+: 匹配至少一次

?: 匹配零次或者一次

.: 匹配除换行符 \n 之外的任何单字符

{n}: n 是一个非负整数, 匹配确定的 n 次

{n,m}: m 和 n 均为非负整数, 表示最多和最少匹配次数, 其中  $n \leq m$

\w: 匹配单个字符(a-z,0-9,\_)

\W: 与 \w 相反

\d: 匹配数字

\D: 与 \d 相反

## 8.Sql 中 delete 与 truncate 的区别

DELETE: 选择性地删除数据, 当删除整张表的数据时效率较低; 只能删除整张表的数据, 但是效率高于使用 DELETE 语句; 不能选择性地删除。当 truncate 执行删除之后, 自动生成的主键值重新从默认值开始。

## 9.Xml 的 java 解析有几种方式?

Java API 解析 xml 主要有两种方式;

Dom 解析: 一次性加载整个文档, 生成树形结构。在生成的文档对象中, 可以对节点进行增删改查的操作。当 xml 文本当较小的时候, 可以使用 dom 解析。

Sax 解析: 基于事件的解析方式, 解析速度比较快, 解析的文档大小理论上是没有限制的。





还有一些开源的技术可以解析 xml, dom4j 或者 jdom。

## 10.Sql 优化有那些方法?

表的设计要规范, 即要符合数据库设计三范式。

适当建立索引, 在频繁作为检索条件, 更新较少的字段上建立索引, 以提高查询速度。

分表查询, 有水平分割、垂直分割。

读写分离, 读(read)、写(create、update、delete)。

建立存储过程。

## 11.Json 是什么?

Json (javascript simple object notation) : 轻量级的数据交换格式。

## 12.如何创建一个 json 对象?

使用{}实例化一个 json 对象, json 对象多个元素使用逗号隔开, 每个元素都是一个键值对

## 13.如何解析 json 对象?

使用 json-lib、gson、jackson 可以解析 json 对象。需要将 json 对象转换成一个 java 对象, 使用 java 对象访问属性。



## 14. 聚集索引与非聚集索引有什么区别？

所有的索引都是为了更快地检索数据，索引存放在索引页中，数据存放在数据页中，索引以 B（balance）树的形式存储

**聚集索引：**聚集索引用于决定数据表中的物理存储顺序，一张表最多有一个聚集索引。聚集索引的字段值尽量不能修改，因为修改后，因为修改后数据表的物理顺序需要重写排序。通常主键就是聚集索引

**非聚集索引：**非聚集索引的关键自是 index，不会决定表的物理存储顺序，在一张表内最多可以有 249 个非聚集索引。

## 15. 一个类文件中能否有多个类？有什么要求？

可以。但是只能有一个 public 类，而且 public 修饰的类名与文件名必须一致

## 16. 你知道有哪些开源框架？

Poi：操作 office 文档

Jfreechat：生成各种图表

Ckeditor：论坛中的富文本输入框

Lucena：用于搜索技术

Pinyin4j：操作汉字的拼音

## 17. 什么是事务？事务有那些特点？

事务：



单个逻辑单元执行的一系列操作，要么全部执行，要么全部不执行。

特点：

原子性 (Atomicity)：事务中各元素不可分割，全部执行成功或者撤销所有的操作

一致性 (Consistency)：事务完成后数据保持一致的状态

隔离性 (Isolation)：事务是相对独立的，对某数据进行修改时，其他事务不变

持久性 (Durability)：事务完成后对系统的影响是永久性的。

## 18.事务的使用场景在什么地方？

但一个业务逻辑包括多个数据库操作的时候，而且需要保证每个数据表操作都执行的成功进行下一个操作，这个时候可以使用事务

## 19.Js 如何实现动态效果？

操作 dom，改变 dom 的结构

## 20.Jsp 由哪些内容组成？

指令：<%@ %>

脚本：<% %>

表达式：<%= %>

声明：<%! %>

注释：<% -- %>



动作: `<jsp: 动作名称 属性=" " >`

静态内容: html 内容

## 21.Jsp 包含那些隐藏对象或者内建对象

Page  
pageContext  
request  
response  
out  
session  
application  
config  
exception

## 22.如何使用 exception 对象?

用于处理 JSP 文件执行时发生的所有错误和异常，只有在 page 指令中设置 `isErrorPage` 值为 `true` 的页面中才可以被使用，在一般的 JSP 页面中使用该对象，将无法编译 JSP 文件。

## 23.Get 请求与 post 有什么区别?

Get 请求发送的文本内容大小有限制，而 post 请求没有限制  
Get 请求的请求参数会出现在 url 路径中，而发送 post 请求时，参数不会显示在 url 路径中  
Get 安全系数较低，但是效率较高。Post 安全系数较高，但效率较低  
Get 请求只能发送字符串，post 请求可以提交二进制数据



## 24.计算机网络有几层?

应用层

表示层

会话层

传输层

网络层

数据链路层

物理层

(物理层是最底层，应用层是最高层)

## 25.常见的计算机网络协议有那些?

TCP/IP 协议

IPX/SPX 协议

NetBEUI 协议

## 26.Tcp 协议的特点

TCP 是面向连接的传输层协议

每一条 TCP 连接只能有两个端点(endpoint),每一条 TCP 连接只能是点对点的 (一对一)

TCP 提供可靠交付的服务

TCP 提供全双工通信

面向字节流



## 27.Java 网络编程有几种？

TCP 编程

UDP 编程

## 28.TCP 编程与 UDP 编程有什么区别？

TCP 协议：传输控制协议，提供可靠无差错的数据传输，效率较低

UDP 协议：用户数据报协议，不可靠的数据传输，效率较高

## 29.创建 socket 通讯的步骤？

服务器程序创建一个 ServerSocket，然后再用 accept 方法等待客户来连接

客户端程序创建一个 Socket 并请求与服务器建立连接

服务器接收客户的连接请求,并创建一个新的 Socket 与该客户建立专线连接

刚才建立了连接的两个 Socket 在一个线程上对话

服务器开始等待新的连接请求

## 30.Java 中如何实现多线程

继承 Thread 类

实现 Runnable 接口

推荐使用 Runnable 接口

## 31.Java 中 Synchronized 关键字的使用？





`Synchronized` 关键字在方法签名上，可以防止多个线程同时访问这个对象的 `synchronized` 修饰的方法。如果一个对象有多个 `synchronized` 方法，只要一个线程访问其中的一个同步方法，那么其他线程就不能访问对象其他的任何一个同步方法。不同对象实例的 `synchronize` 方法是互不干扰的，也就是说，其他对象还可以访问这个类中的同步方法。

`Synchronized` 如果修饰的是静态方法，防止多个线程同时访问这个类中的静态同步方法，它对类中所有对象都能起作用。也就是说，只有一个对象一个线程可以访问静态同步方法

`Synchronized` 修饰方法中的某段代码块，只对当前代码块实行互斥访问。当多个线程同步访问同步代码块，同一时间只能有一个线程得到执行，其他线程必须等待当前线程执行完代码块之后才能执行。当一个线程访问同步代码块时，其他线程可以访问非同步的代码。当一个线程访问同步代码块时，那么其他线程访问对其他同步代码块的访问将会被阻塞

`Synchronized` 关键字是不能继承的，如果父类的 `synchronized` 在继承时并不自动是 `synchronized` 修饰的，需要显示地声明。

`Synchronized` 修饰 `this` 时，会得到这个对象的对象锁，当一个线程访问时，那么其他线程访问对象的所有同步代码块或者同步方法，将会被阻塞。

## 32.什么是线程池？

在一个应用程序中初始化一个线程集合，然后在需要执行新的任务时重用线程池中的线程，而不是创建一个新的线程。线程池中的每个线程都有被分配一个任务，一旦任务完成，线程就回到线程池中，等待下一次的任务分配

## 33.什么是游标？



游标是 sql 查询结果集的一个指针，与 select 语句相关联。

游标关键字是 cursor，主要包含两个部分：游标结果集和游标位置。

游标结果集：执行 select 语句后的查询结果

游标位置：一个指向游标结果集内某条记录的指针。

游标主要有两个状态：打开和关闭。

只有当游标处于打开状态时，才能够操作结果集中的数据

当游标关闭后，查询结果集就不存在了

## 34.游标的创建步骤？

定义游标

打开游标

操作游标数据

关闭游标

## 35.在做文件上传的时候，form 表单的 enctype 的指是什么？

Multipart/form-data

## Html 中 a 标签的 target 属性有哪些值？



\_blank: 在新的窗口打开网页

\_self: 在本页面打开新网页

\_parent: 在父 frameset 框架打开网页

\_top: 去掉所有页面框架, 使用 document.html 替代 frameset 文档

### 36.当打开其他程序的网页时, 使用的 target 属性是哪个?

\_top

### 37.Servlet 的生命周期?

加载: 判断 servlet 实例是否存在, 如果不存在, 就加载 servlet

实例化:

初始化

服务

销毁

### 38.Servlet 生命周期内调用的方法过程?

Init()

Service()

doGet 或者 doPost

destroy

### 39.线程的生命周期?



新建

就绪

运行

死亡

阻塞

## 40.Statement 与 preparedStatement 区别

preparedStatement 会预编译 sql 语句，能够提高批量的数据操作的执行效率，Statement 执行 sql 的时候才进行编译

PreparedStatement 在第一次执行 sql 的时候，比较耗费资源。如果只对数据库进行一次操作，使用 statement 比较好。

Statement 会出现 sql 注入的问题，使用 preparedStatement 可以解决 sql 注入

## 41.Session 与 cookie 的区别？

Session 保存在服务端，cookie 保存在客户端

Session 保存是对象，cookie 只能保存字符串

Session 不能设置路径，cookie 可以设置保存路径。同一个网站不同网页的 cookie 可以保存到不同的路径下，彼此是无法相互访问的。

Session 在服务器关闭后会自动消失，cookie 则不会。

## 42.存储过程与函数的区别

存储过程的关键字是 procedure，函数关键字是 function



函数必须有返回值，存储过程没有返回值，但是有传出参数

函数注重的是结果，存储过程注重的是过程

函数可以在 select 语句中直接使用，而存储过程则不能

## 43.会话跟踪技术有那些？

Session

Cookie

表单隐藏域

url 重写

## 44.jspervlet 中通信作用域有那些？

PageContext

Request

Session

Application

## 45.url 是什么？由哪些部分组成？

统一资源定位符

http://localhost:8080/myWeb/index.html: 协议+主机地址+端口+项目名称  
+资源名称

## 46.你所知道的 web 服务器有哪些？

Tomcat



Jboss

Weblogic

Glassfish

## 47.如何部署一个 web 项目?

可以将 web 项目打包成.war 文件  
把 war 文件放在 webapps 文件夹中

## 48.如何修改 tomcat 的端口号?

在 tomcat 根目录的 conf 文件夹内打开 server.xml 文件，修改 Connector 节点的 port 属性

## 49.重定向和请求转发的区别?

请求转发只能将请求转发给同一个 Web 应用中的其他资源，而重定向不仅可以定向到当前应用程序中的其他资源，也可以重定向到其他站点上的资源。

重定向结束后，浏览器地址栏显示 URL 会发生改变，由初始的 URL 地址变成重定向的目标 URL。而请求转发过程结束后，浏览器地址栏保持初始的 URL 地址不变。

请求转发的发起者和接受者之间共享相同的 HttpServletRequest 实例和 HttpServletResponse 实例，而重定向的发起者和接受者使用各自的 HttpServletRequest 实例和 HttpServletResponse 实例。

转发是一次请求，重定向是二次请求。转发是在服务器进行的，重定向在客户端进行的。





## 50.上传文件是如何做的？

上传文件主要用的是开源组件，如 apache commoms fileupload 或者  
jspformatupload

## 51.HTTP 的状态码

200：请求成功

400：不是正确的请求，大多情况下表示参数错误

404：找不到请求资源

500：服务器内部错误

403：服务器拒绝

405：请求的 method 不支持

504：服务器临时不可用

## 52.Jsp 指令有那些？

Include

Taglib

Page

## 53.如何使 session 失效

Session.invalidate()



## 54.如何在 jsp 页面上显示一些特定格式的数字或者日期

使用 jstl 标签库，使用 numberformat 或者 dateformat 标签

## 55.什么是线程异步？什么是线程同步？

线程同步：同时只有一条线程执行一个任务

线程异步：同时有多条线程可以执行执行任务

## 56.什么是同步任务？什么是异步任务？

同步任务：当前任务没有完成之前，其他任务不能够执行

异步任务：当前任务没有完成，任然可以可以发送一个新的请求

## 57.什么是过滤器？怎么创建一个过滤器

过滤器：在请求发送之后，处理之前对请求的一次拦截，可以更改请求状态或者参数值等。

创建过滤器：实现 filter 接口，重写 doFilter 方法，最后在 web.xml 中配置过滤器

## 58.Int 与 integer 的区别



Integer 是 int 的包装类型。

Int 的默认值是 0，integer 的默认值是 null

## 59.如何进行单元测试

使用 junit

## 60.使用 sql 写出一个分页程序?

Select top 3 \* from tb\_name where id not in (select top 3 id from tb\_name)

## 61.监听器有哪些作用和用法?

Java Web 开发中的监听器 (listener) 就是 application、session、request 三个对象创建、销毁或者往其中添加修改删除属性时自动执行代码的功能组件，如下所示：

ServletContextListener：对 Servlet 上下文的创建和销毁进行监听。

ServletContextAttributeListener：监听 Servlet 上下文属性的添加、删除和替换。

HttpSessionListener：对 Session 的创建和销毁进行监听。

## 62.JSP 中的静态包含和动态包含有什么区别?

静态包含是通过 JSP 的 include 指令包含页面，动态包含是通过 JSP 标准动作 包含页面。



静态包含是编译时包含，如果包含的页面不存在则会产生编译错误，而且两个页面的"contentType"属性应保持一致，因为两个页面会合二为一，只产生一个 class 文件，因此被包含页面发生的变动再包含它的页面更新前不会得到更新。

动态包含是运行时包含，可以向被包含的页面传递参数，包含页面和被包含页面是独立的，会编译出两个 class 文件，如果被包含的页面不存在，不会产生编译错误，也不影响页面其他部分的执行。

## 63. Servlet 中如何获取用户提交的查询参数或表单数据？

可以通过请求对象 (HttpServletRequest) 的 getParameter() 方法通过参数名获得参数值。如果有包含多个值的参数（例如复选框），可以通过请求对象的 getParameterValues() 方法获得。当然也可以通过请求对象的 getParameterMap() 获得一个参数名和参数值的映射 (Map)。

## 64. 如何设置请求的编码以及响应内容的类型？

通过请求对象 (ServletRequest) 的 setCharacterEncoding(String) 方法可以设置请求的编码，其实要彻底解决乱码问题就应该让页面、服务器、请求和响应、Java 程序都使用统一的编码，最好的选择当然是 UTF-8；通过响应对象

(ServletResponse) 的 setContentType(String) 方法可以设置响应内容的类型，当然也可以通过 HttpServletResponse 对象的 setHeader(String, String) 方法来设置。

## 65. 什么是 Web Service (Web 服务)



从表面上看, Web Service 就是一个应用程序, 它向外界暴露出一个能够通过 Web 进行调用的 API。这就是说, 你能够用编程的方法透明的调用这个应用程序, 不需要了解它的任何细节, 跟你使用的编程语言也没有关系。例如可以创建一个提供天气预报的 Web Service, 那么无论你用哪种编程语言开发的应用都可以通过调用它的 API 并传入城市信息来获得该城市的天气预报。之所以称之为 Web Service, 是因为它基于 HTTP 协议传输数据, 这使得运行在不同机器上的不同应用无须借助附加的、专门的第三方软件或硬件, 就可相互交换数据或集成。

## 66.什么是 ORM?

对象关系映射 (Object-Relational Mapping, 简称 ORM) 是一种为了解决程序的面向对象模型与数据库的关系模型互不匹配问题的技术

## 67.Hibernate 中 SessionFactory 是线程安全的吗? Session 是线程安全的吗 (两个线程能够共享同一个 Session 吗)?

SessionFactory 对应 Hibernate 的一个数据存储的概念, 它是线程安全的, 可以被多个线程并发访问。SessionFactory 一般只会在启动的时候构建。对于应用程序, 最好将 SessionFactory 通过单例模式进行封装以便于访问。Session 是一个轻量级非线程安全的对象 (线程间不能共享 session), 它表示与数据库进行交互的一个工作单元。Session 是由 SessionFactory 创建的, 在任务完成之后它会被关闭。Session 是持久层服务对外提供的主要接口。Session 会延迟获取数据库连接 (也就是在需要的时候才会获取)。为了避免创建太多的 session, 可以使用 ThreadLocal 将 session 和当前线程绑定在一起, 这样可以让同一个线程获得的总是同一个 session。Hibernate 3 中 SessionFactory 的 getCurrentSession() 方法就可以做到。



## 68. Hibernate 中 Session 的 load 和 get 方法的区别是什么？

如果没有找到符合条件的记录，get 方法返回 null，load 方法抛出异常。

get 方法直接返回实体类对象，load 方法返回实体类对象的代理。

在 Hibernate 3 之前，get 方法只在一级缓存中进行数据查找，如果没有找到对应的数据则越过二级缓存，直接发出 SQL 语句完成数据读取；load 方法则可以从二级缓存中获取数据；从 Hibernate 3 开始，get 方法不再是对二级缓存只写不读，它也是可以访问二级缓存的。

## 69. Hibernate 的对象有几种状态

瞬时态 (transient)

持久态 (persistent)

游离态 (detached)

## 70. Session 的 save()、update()、merge()、lock()、saveOrUpdate() 和 persist() 方法分别是做什么的？有什么区别？

瞬时态的实例可以通过调用 save()、persist() 或者 saveOrUpdate() 方法变成持久态；游离态的实例可以通过调用 update()、saveOrUpdate()、lock() 或者





replicate()变成持久态。save()和 persist()将会引发 SQL 的 INSERT 语句，而 update()或 merge()会引发 UPDATE 语句。save()和 update()的区别在于一个是将瞬时态对象变成持久态，一个是将游离态对象变为持久态。merge()方法可以完成 save()和 update()方法的功能，它的意图是将新的状态合并到已有的持久化对象上或创建新的持久化对象。

对于 persist()方法，

persist()方法把一个瞬时态的实例持久化，但是并不保证标识符被立刻填入到持久化实例中，标识符的填入可能被推迟到 flush 的时间；

persist()方法保证当它在一个事务外部被调用的时候并不触发一个 INSERT 语句，当需要封装一个长会话流程的时候，persist()方法是很有必要的；

save()方法不保证第②条，它要返回标识符，所以它会立即执行 INSERT 语句，不管是在事务内部还是外部。至于 lock()方法和 update()方法的区别，update()方法是把一个已经更改过的脱管状态的对象变成持久状态；lock()方法是把一个没有更改过的脱管状态的对象变成持久状态。

## 71.Session 加载实体对象的过程。

Session 在调用数据库查询功能之前，首先会在一级缓存中通过实体类型和主键进行查找，如果一级缓存查找命中且数据状态合法，则直接返回；

如果一级缓存没有命中，接下来 Session 会在当前 NonExists 记录（相当于一个查询黑名单，如果出现重复的无效查询可以迅速做出判断，从而提升性能）中进行查找，如果 NonExists 中存在同样的查询条件，则返回 null；

如果一级缓存查询失败则查询二级缓存，如果二级缓存命中则直接返回；



如果之前的查询都未命中，则发出 SQL 语句，如果查询未发现对应记录则将此次查询添加到 Session 的 NonExists 中加以记录，并返回 null；

根据映射配置和 SQL 语句得到 ResultSet，并创建对应的实体对象；

将对象纳入 Session（一级缓存）的管理；

如果有对应的拦截器，则执行拦截器的 onLoad 方法；

如果开启并设置要使用二级缓存，则将数据对象纳入二级缓存；

返回数据对象。

## 72.Query 接口的 list 方法和 iterate 方法有什么区别？

list()方法无法利用一级缓存和二级缓存（对缓存只写不读），它只能在开启查询缓存的前提下使用查询缓存；iterate()方法可以充分利用缓存，如果目标数据只读或者读取频繁，使用 iterate()方法可以减少性能开销。

list()方法不会引起 N+1 查询问题，而 iterate()方法可能引起 N+1 查询问题

## 73.如何理解 Hibernate 的延迟加载机制？在实际应用中，延迟加载与 Session 关闭的矛盾是如何处理的？



延迟加载就是并不是在读取的时候就把数据加载进来，而是等到使用时再加载。Hibernate 使用了虚拟代理机制实现延迟加载，我们使用 Session 的 load()方法加载数据或者一对多关联映射在使用延迟加载的情况下从一的一方加载多的一方，得到的都是虚拟代理，简单的说返回给用户的并不是实体本身，而是实体对象的代理。代理对象在用户调用 getter 方法时才会去数据库加载数据。但加载数据就需要数据库连接。而当我们把会话关闭时，数据库连接就同时关闭了。

延迟加载与 session 关闭的矛盾一般可以这样处理：

关闭延迟加载特性。这种方式操作起来比较简单，因为 Hibernate 的延迟加载特性是可以通过映射文件或者注解进行配置的，但这种解决方案存在明显的缺陷。首先，出现 "no session or session was closed" 通常说明系统中已经存在主外键关联，如果去掉延迟加载的话，每次查询的开销都会变得很大。

在 session 关闭之前先获取需要查询的数据，可以使用工具方法 Hibernate.isInitialized() 判断对象是否被加载，如果没有被加载则可以使用 Hibernate.initialize() 方法加载对象。

使用拦截器或过滤器延长 Session 的生命周期直到视图获得数据。Spring 整合 Hibernate 提供的 OpenSessionInViewFilter 和 OpenSessionInViewInterceptor 就是这种做法。

## 74. 简述 Hibernate 常见优化策略。

制定合理的缓存策略（二级缓存、查询缓存）。

采用合理的 Session 管理机制。

尽量使用延迟加载特性。

设定合理的批处理参数。

如果可以，选用 UUID 作为主键生成器。

如果可以，选用乐观锁替代悲观锁。



在开发过程中, 开启 `hibernate.show_sql` 选项查看生成的 SQL, 从而了解底层的状况; 开发完成后关闭此选项。

考虑数据库本身的优化, 合理的索引、恰当的数据分区策略等都会对持久层的性能带来可观的提升, 但这些需要专业的 DBA (数据库管理员) 提供支持。

## 75.谈一谈 Hibernate 的一级缓存、二级缓存和查询缓存。

Hibernate 的 Session 提供了一级缓存的功能, 默认总是有效的, 当应用程序保存持久化实体、修改持久化实体时, Session 并不会立即把这种改变提交到数据库, 而是缓存在当前的 Session 中, 除非显示调用了 Session 的 `flush()` 方法或通过 `close()` 方法关闭 Session。通过一级缓存, 可以减少程序与数据库的交互, 从而提高数据库访问性能。

SessionFactory 级别的二级缓存是全局性的, 所有的 Session 可以共享这个二级缓存。不过二级缓存默认是关闭的, 需要显示开启并指定需要使用哪种二级缓存实现类 (可以使用第三方提供的实现)。一旦开启了二级缓存并设置了需要使用二级缓存的实体类, SessionFactory 就会缓存访问过的该实体类的每个对象, 除非缓存的数据超出了指定的缓存空间。

一级缓存和二级缓存都是对整个实体进行缓存, 不会缓存普通属性, 如果希望对普通属性进行缓存, 可以使用查询缓存。查询缓存是将 HQL 或 SQL 语句以及它们的查询结果作为键值对进行缓存, 对于同样的查询可以直接从缓存中获取数据。查询缓存默认也是关闭的, 需要显示开启。

## 76.MyBatis 中使用#和\$书写占位符有什么区别?





#将传入的数据都当成一个字符串，会对传入的数据自动加上引号；\$将传入的数据直接显示生成在 SQL 中。注意：使用\$占位符可能会导致 SQL 注射攻击，能用#的地方就不要使用\$，写 order by 子句的时候应该用\$而不是#。、

## 77.什么是 IoC 和 DI? DI 是如何实现的?

IoC 叫控制反转，是 Inversion of Control 的缩写，DI (Dependency Injection) 叫依赖注入，是对 IoC 更简单的诠释。控制反转是把传统上由程序代码直接操控的对象的调用权交给容器，通过容器来实现对象组件的装配和管理。所谓的“控制反转”就是对组件对象控制权的转移，从程序代码本身转移到了外部容器，由容器来创建对象并管理对象之间的依赖关系。依赖注入的基本原则是应用组件不应该负责查找资源或者其他依赖的协作对象。配置对象的工作应该由容器负责，查找资源的逻辑应该从应用组件的代码中抽取出来，交给容器来完成。DI 是对 IoC 更准确的描述，即组件之间的依赖关系由容器在运行期决定，形象的来说，即由容器动态的将某种依赖关系注入到组件之中。

依赖注入可以通过 setter 方法注入（设值注入）、构造器注入和接口注入三种方式来实现，Spring 支持 setter 注入和构造器注入，通常使用构造器注入来注入必须的依赖关系，对于可选的依赖关系，则 setter 注入是更好的选择，setter 注入需要类提供无参构造器或者无参的静态工厂方法来创建对象。

## 78.Spring 中自动装配的方式有哪些?

no：不进行自动装配，手动设置 Bean 的依赖关系。

byName：根据 Bean 的名字进行自动装配。

byType：根据 Bean 的类型进行自动装配。

constructor：类似于 byType，不过是应用于构造器的参数，如果正好有一个 Bean 与构造器的参数类型相同则可以自动装配，否则会导致错误。



autodetect: 如果有默认的构造器, 则通过 constructor 的方式进行自动装配, 否则使用 byType 的方式进行自动装配。

## 79.解释一下什么叫 AOP（面向切面编程）？

AOP (Aspect-Oriented Programming) 指一种程序设计范型, 该范型以一种称为切面 (aspect) 的语言构造为基础, 切面是一种新的模块化机制, 用来描述分散在对象、类或方法中的横切关注点 (crosscutting concern)。

## 80.Spring 中 Bean 的作用域有哪些？

Singleton: Bean 以单例的方式存在

Prototype: 表示每次从容器中调用 Bean 时, 都会返回一个新的实例, prototype 通常翻译为原型

Request: 每次 HTTP 请求都会创建一个新的 Bean

Session: 同一个 HttpSession 共享同一个 Bean, 不同的 HttpSession 使用不同的 Bean

globalSession: 同一个全局 Session 共享一个 Bean

## 81.Spring 中如何使用注解来配置 Bean? 有哪些相关的注解?

首先需要在 Spring 配置文件中增加配置:

用 @Component、@Controller、@Service、@Repository 注解来标注需要由 Spring IoC 容器进行对象托管的类。





@Controller 通常用于控制器

@Service 通常用于业务逻辑类

@Repository 通常用于 DAO 类

普通的类用@Component 来标注。

## 82.Spring 支持的事务管理类型有哪些？你在项目中 使用哪种方式？

Spring 支持编程式事务管理和声明式事务管理。声明式事务管理要优于编程式事务管理，尽管在灵活性方面它弱于编程式事务管理，因为编程式事务允许通过代码控制业务。

## 83.Spring MVC 的工作原理是怎样的？

客户端的所有请求都交给前端控制器 DispatcherServlet 来处理，它会负责调用系统的其他模块来真正处理用户的请求。

DispatcherServlet 收到请求后，将根据请求的信息（包括 URL、HTTP 协议方法、请求头、请求参数、Cookie 等）以及 HandlerMapping 的配置找到处理该请求的 Handler（任何一个对象都可以作为请求的 Handler）。

在这个地方 Spring 会通过 HandlerAdapter 对该处理器进行封装。

HandlerAdapter 是一个适配器，它用统一的接口对各种 Handler 中的方法进行调用。



Handler 完成对用户请求的处理后，会返回一个 ModelAndView 对象给 DispatcherServlet，ModelAndView 顾名思义，包含了数据模型以及相应的视图的信息。

ModelAndView 的视图是逻辑视图，DispatcherServlet 还要借助 ViewResolver 完成从逻辑视图到真实视图对象的解析工作。

当得到真正的视图对象后，DispatcherServlet 会利用视图对象对模型数据进行渲染。

客户端得到响应，可能是一个普通的 HTML 页面，也可以是 XML 或 JSON 字符串，还可以是一张图片或者一个 PDF 文件。

## 84. 阐述 Spring 框架中 Bean 的生命周期？

Spring IoC 容器找到关于 Bean 的定义并实例化该 Bean。

Spring IoC 容器对 Bean 进行依赖注入。

如果 Bean 实现了 BeanNameAware 接口，则将该 Bean 的 id 传给 setBeanName 方法。

如果 Bean 实现了 BeanFactoryAware 接口，则将 BeanFactory 对象传给 setBeanFactory 方法。

如果 Bean 实现了 BeanPostProcessor 接口，则调用其 postProcessBeforeInitialization 方法。

如果 Bean 实现了 InitializingBean 接口，则调用其 afterPropertySet 方法。

如果有和 Bean 关联的 BeanPostProcessors 对象，则这些对象的 postProcessAfterInitialization 方法被调用。

当销毁 Bean 实例时，如果 Bean 实现了 DisposableBean 接口，则调用其 destroy 方法。