需要学习知识

1、正则表达式

2、创建表格及自动创建bean、mapper

3、seajs的格式含义 define(function(require,exports,module){})  //定义一个模块

在只传递一个参数给define时（推荐写法），这个参数就是工厂函数，此时工厂函数的三个参数分别是：

*require*——模块加载函数，用于记载依赖模块。

*exports*——接口点，将数据或方法定义在其上则将其暴露给外部调用。

*module*——模块的元数据。

这三个参数可以根据需要选择是否需要显示指定。

下面说一下module。module是一个对象，存储了模块的元信息，具体如下：

[*module.id*](http://module.id/)——模块的ID。

*module.dependencies*——一个数组，存储了此模块依赖的所有模块的ID列表。

*module.exports*——与exports指向同一个对象。

4、java 里面怎么使用antx.properties里面定义的变量

5、mybatis generator的基本知识

  仔细阅读 链接内容：<http://blog.csdn.net/isea533/article/details/42102297>

6、数据库支持类型

使用整数数据的精确数字数据类型。

**bigint**

从 -2^63 (-9223372036854775808) 到 2^63-1 (9223372036854775807) 的整型数据（所有数字）。存储大小为 8 个字节。

**int**

从 -2^31 (-2,147,483,648) 到 2^31 - 1 (2,147,483,647) 的整型数据（所有数字）。存储大小为 4 个字节。int 的 SQL-92 同义字为 integer。

**smallint**

从 -2^15 (-32,768) 到 2^15 - 1 (32,767) 的整型数据。存储大小为 2 个字节。

**tinyint**

从 0 到 255 的整型数据。存储大小为 1 字节。

7、maven 知识学习

1. mvn compile 编译源代码 2. mvn test-compile 编译测试代码 3. mvn test 运行测试 5. mvn package 打包，根据pom.xml打成war或jar 如果pom.xml中设置 war，则此命令相当于mvn war:war 如果pom.xml中设置 jar，则此命令相当于mvn jar:jar 6. mvn -Dtest package 打包但不测试。完整命令为：mvn -D maven.test.skip=true package 7. mvn install 在本地Repository中安装jar 8. mvn clean 清除产生的项目 9. mvn eclipse:eclipse 生成eclipse项目 10.mvn idea:idea 生成idea项目 11.mvn eclipse:clean 清除eclipse的一些系统设置

一般使用情况是这样， 1、通过cvs或svn下载代码到本机 2、执行mvn eclipse:eclipse生成ecllipse项目文件 3、然后导入到eclipse就行了 4、修改代码后执行mvn compile或mvn test检验

8、git 知识学习

select [t1.id](http://t1.id/), t1.channel\_number, t1.creator,t1.modifier, t1.status,

t1.channel\_type, t1.partner, t1.partner\_pinyin\_short,

t1.model, t1.uv\_tag, t1.platform\_type,t1.status\_change\_time, t1.gmt\_create,

t1.gmt\_modified

from mupp\_channel t1 where t1.status=0 AND [t1.id](http://t1.id/) not in (select

t2.channel\_id from mupp\_channel\_group\_item t2 where

t2.channel\_group\_id=#{channelGroupId})

<if test="hannelNumber !=null"]]]]>

t1.channel\_number like '%$hannelNumber$%'

</if]]]]>

<if test="channelType !=null"]]]]>

t1.channel\_type = #{channelType}

</if]]]]>

<if test="partner !=null"]]]]>

t1.partner =#{partner}

</if]]]]>

<if test="platformType !=null"]]]]>

t1.plateform\_type = #{platformType}

</if]]]]>

<if test="appkey !=null"]]]]>

t1.appkey = #{appkey}

</if]]]]>

</trim]]]]>

<if test="pageNum !=null"]]]]>

limit #{start},#{pageSize}

</if]]]]>

9 、mysql 中文排序 方法 ORDER BY CONVERT(column\_name USING gbk)

10. js控制页面跳转的方法

<http://blog.csdn.net/kongjing0815/article/details/5808063>

11、http 错误返回码

400 Bad Request

因为错误的语法导致服务器无法理解请求信息。

401 Unauthorized  如果请求需要用户验证。回送应该包含一个WWW-Authenticate头字段用来指明请求资源的权限。

402 Payment Required  保留状态码

403 Forbidden  服务器接受请求，但是被拒绝处理。

404 Not Found  服务器已经找到任何匹配Request-URI的资源。

500 Internal Server Error  服务器遭遇异常阻止了当前请求的执行

501 Not Implemented  服务器没有相应的执行动作来完成当前请求。

502 Bad Gateway

503 Service Unavailable  因为临时文件超载导致服务器不能处理当前请求。

504 Gateway Timeout

505 Http Version Not Supported

任务：

（1）jquery的知识看一遍

（2）ajax相关的知识

（3）mybatis知识

（4）git知识

（5）css布局及调用

（6）bootstrap /sublime 模板阅读

（7）阅读mtl的源码

12、写rpc服务的方法

@WebResource("userQuery”)

—引入注解WebResource，参数为target

@ResourceMapping("/aliwws”)

—引入方法,通过注解ResourceMapping，参数为方法名

方法体：public Map<String, UserContactInfo> infos(@RequestParam(name = "names") String names)

—具体的方法名可以不和映射的名称一致，参数使用注解RequestParam

访问方式：通过url直接进行访问

13、线程安全

如果你的代码所在的进程中有多个线程在同时运行，而这些线程可能会同时运行这段代码。如果每次运行结果和[单线程](http://baike.baidu.com/view/2554947.htm)运行的结果是一样的，而且其他的[变量](http://baike.baidu.com/view/296689.htm)的值也和预期的是一样的，就是线程安全的。

或者说:一个类或者程序所提供的接口对于线程来说是[原子操作](http://baike.baidu.com/view/809659.htm)或者多个线程之间的切换不会导致该接口的执行结果存在二义性,也就是说我们不用考虑同步的问题。

线程安全问题都是由[全局变量](http://baike.baidu.com/view/261041.htm)及[静态变量](http://baike.baidu.com/view/675642.htm)引起的。

14、自定义velocity的标签（指令）

velocity支持的指令有：#set,#foreach,#if #else #end,#parse,#include,#evaluate,#define，#macro，

在velocity的jar包中的directive.properties中定义了这些实现：

自定义标签和指定，比如我们定义了下面的remoteVelocity指令

要对这个指令的实现要继承Directive这个类，这个宏我们可以从其他服务获取vm的内容，动态渲染，这种方式可以统一管理公共模板，

node.jjtGetChild(2) 这个方法可以获取对应指令的参数，下标从0开始，

在web工程的WEB-INF下面定义velocity.properties这个配置文件，用户扩展的指令最好放到这个文件里面，velocity的jar包里面提供了默认实现，我们可以覆盖重新定义自己的扩展，类就是对应自己的扩展类的类名

 #自定义标签

1. userdirective=com.alibaba.click.test.RemoteVelocity
2. 这样启动后就可以正常使用了。
3. Directive的三个方法：
4. Java代码  
5. getName：指令的名称
6. getType:当前有LINE,BLOCK两个值，line行指令，不要end结束符，block块指令，需要end结束符
7. public boolean render(InternalContextAdapter context, Writer writer,   Node node) 具体处理过程

15、过滤器filter

//配置方法和配置普通的Servlet一样的哦

<filter>

      <filter-name>PrivFilter</filter-name>

      <filter-class>com.myPriv.filter.PrivFilter</filter-class>

   </filter>

<filter-mapping>

    <filter-name>PrivFilter</filter-name>

    <url-pattern>/resource/\*</url-pattern> //过滤的匹配资源、模式，对匹配的资源进行过滤

</filter-mapping>

然后你可以把过滤条件写出来,写在doFilter方法里面

java会在处理每一个Servlet时都走一次Filter的。调用dofilter 而且chain.doFilter(request.response);是必须的。没有他，程序就不会继续了

16、**ThreadLocal**

17、改变url的地址，不刷新（pjax）

然而HTML5的新接口pushState / replaceState就可以比较完美的解决问题，它避免了改变hash的问题，避免了用户不理解URL的形式感到疑惑，同时还有onpopstate提供监听，良好响应后退前进。而且它不需要这个URL真实存在。

HTML5 的 pushState+Ajax

HTML5提供history接口，把URL以state的形式添加或者替换到浏览器中，其实现函数正是 pushState 和 replaceState。

pushState 例子

pushState() 的基本参数是：

window.history.pushState(state, title, url);

其中state和title都可以为空，但是推荐不为空，应当创建state来配合popstate监听。

replaceState 同理

window.history.replaceState( state, *~title*, *~href*);

pushState、replaceState 的区别

pushState()可以创建历史，可以配合popstate事件，而replaceState()则是替换掉当前的URL，不会产生历史。

if(history.pushState){

var **state**=({

url: uri, title: newTitle

});

window.history.pushState(**state**, newTitle, uri);

}else{ window.location.href="#!"+*~fakeURI*; } // 如果不支持，使用旧的解决方案

return false;

});

**18、传递参数**

vm模板传递参数方法：$control.setTemplate(‘xxx.vm’).setParameter('参数名称’,参数值)

 xxx.vm中可以直接使用传递进去的参数进行显示

页面的传参数：$xxx.setTarget(‘xxx.htm’).addQueryData('参数名称’,参数值)

**19、时间格式化的方式：**$!{date.format('yyyy-MM-dd HH:mm:ss', 时间值)}

20、java服务端跳转方法：

（1）TurbineURIBroker uri = (TurbineURIBroker) uriBrokerService.getURIBroker("muppProjectModule”);//参数为module变量

    redirectUrl = uri.setTarget("projectDetail.htm").addQueryData("projectId",projectPublish.getMuppProjectId()).render();

（2） nav.redirectToLocation("h5MonitorUrlList”); //外部重定向

     nav.forwardTo("h5MonitorUrlEdit").withParameter("id", String.valueOf(h5MonitorUrl.getId()));//内部重定向

21、同一个form中提交不同的方法处理：（场景：提交同一个form,但是处理的方法不同）

<form class="form-horizontal " method="post" action="">

      <input type="hidden" name="action" value="SystemNoticeAction”/>//定义action

#if(${systemNotice.id})

<input type="submit" class="btn btn-primary" name="event\_submit\_do\_update\_notice" value="修改" />//方法名为：doUpdateNotice()

#else

<input type="submit" class="btn btn-primary" name="event\_submit\_do\_create\_notice" value="创建" />方法名为：doCreateNotice()

#end

在action类SystemNoticeAction中定义两个方法，名字分别为：doUpdateNotice()\doCreateNotice()，分别对应两个submit事件，注意：SystemNoticeAction类的定义重新定义一个action包

22、按照特定的状态排序

     projectQuery.setOrderByClause("field(project\_status,\"INTEGRATE\",\"GRADIENT\",\"DEVELOP\",\"PUBLISHED\"),expect\_publish\_date desc”);

    中文排序 CONVERT(列名 USING gbk);

23、list 直接初始化方法：

ArrayList<String> list = new ArrayList<String>(){{add("str01"); add("str02");}};

24、常用的mysql函数：

—字符串操作---

select locate('sdfdf','sdfsdfdsfsfsdfdf');

select position('ff' in 'sdfff');

select instr('hellp','lp');

select lpad('xdsdd',8,'??');

select rpad('sd',4,'xx');

select left("xcddfdf",4);

select right("xcddfdf",4);

select substring('strrr',2,5);

select SUBSTRING\_INDEX('[www.mysql.com](http://www.mysql.com/)', '.', -2);

select ltrim('  dffd');

select rtrim('  dffd   ');

select TRIM('  bar   ');

select TRIM(LEADING 'x'  FROM  'xxxbarxxx');

select TRIM(BOTH 'x'  FROM  'xxxbarxxx');

select TRIM(TRAILING 'x'  FROM  'xxxbarxxx');

select SPACE(6);

select REPLACE('[www.mysql.com](http://www.mysql.com/)', 'w', 'Wx');

select REPEAT('str',3);

select REVERSE('abc');

select insert('Quadratic', 3, 4, 'What');

select ELT(2, 'ej', 'Heja', 'hej', 'foo');

select FIELD('ej', 'Hej', 'ej', 'Heja', 'hej','foo');

select FIND\_IN\_SET('b','a,b,c,d');

SELECT MAKE\_SET(1 | 4,'hello','nice','world');

select LCASE('QUADRATICALLY'); select lower('QUADRATICALLY');

select UCASE('quadratically'); select upper('quadratically’);

———数学方法----------

select ABS(-2);

select SIGN(-32);

select MOD(234, 10);

select FLOOR(1.23);

select CEILING(1.23);

select ROUND(1.298, 1);

select EXP(2);

select LOG(2);

select LOG10(2);

select POW(2,2);

select SQRT(20);

select COS(PI());select sin(PI());select tan(PI());select cot(PI());select asin(PI());

select RAND();

select LEAST(34.0,3.0,5.0,767.0); select greatest(34.0,3.0,5.0,767.0);

————时间操作------------

select DAYOFWEEK('1998-02-03');

select WEEKDAY('1997-10-04 22:23:00');

select DAYOFYEAR('1998-02-03');

select MONTH('1998-02-03');

select DAYNAME("1998-02-05");

select MONTHNAME("1998-02-05");

select QUARTER('98-04-01');

select WEEK('1998-02-20');

SELECT DATE\_ADD("1997-12-31 23:59:59",INTERVAL"1:2" MINUTE\_SECOND);

select FROM\_DAYS(729669);

select DATE\_FORMAT('1997-10-04 22:23:00','%H:%i:%s');

select UNIX\_TIMESTAMP();

————转换操作-------------

select CONVERT(xxx  USING   gb2312);

CAST(xxx  AS   类型), CONVERT(xxx,类型)

##group by作用

———去重操作

select distinct(id);

———计数操作————

count()

25、Android 指定多进程：

<http://blog.csdn.net/hudashi/article/details/7858125>

26、href 中target标签:

target=“\_black” 新窗口打开

         =“\_new” 新窗口，只打开一次

         =“\_parent” 父类窗口

href中锚点的使用：<a href=“#锚点名称”>\*\*\*</a>

<div id =“锚点名称” >\*\*\*</div>

27、单元格合并

colspan =’n’ 横向合并n个

rowspan =’n’ 纵向合并n个

内嵌的样式：<style >\*\*\*</style>

28、js的匿名函数

(function(){})();//定义一个匿名函数，并执行

var GLOBAL={} //定义一个全局的对象类，赋值 GLOBAL.A={},A.str=“ddd”

29、synchronized 使用

无论synchronized关键字加在方法上还是对象上，它取得的锁都是对象，而不是把一段代码或函数当作锁――同步方法很可能还会被其他线程的对象访问

如果要保持即使不同对象也只能有一个线程来访问可以创建一个特殊的instance变量（它得是一个对象）来充当锁（用byte数组对象比Object Object lock = new Object() 高效  注意这个得是static的，让不同对象竞争同一个 byte数组对象的锁）

30、hashcode 与equal

31、序列化与反序列化