1. ajax异步的 JavaScript 和 XML(实现部分刷新)
2. 创建xmlhttprequest对象（ie7以下使用兼容方法）

var xmlhttp;

if (window.XMLHttpRequest)

{// code for IE7+, Firefox, Chrome, Opera, Safari

xmlhttp=new XMLHttpRequest();

}

else

{// code for IE6, IE5

xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");

}

1. 请求方法（open（”请求方法”,”请求内容uri”,”同步/异步选项”））

xmlhttp.open("GET","test1.txt",true);

xmlhttp.send();

1. 服务器响应方法（innerHtml）

document.getElementById("ID").innerHTML=xmlhttp.responseText;

1. 实例

<script type="text/javascript">

var xmlhttp;

function loadXMLDoc(url)

{

xmlhttp=null;

if (window.XMLHttpRequest)

{// code for Firefox, Opera, IE7, etc.

xmlhttp=new XMLHttpRequest();

}

else if (window.ActiveXObject)

{// code for IE6, IE5

xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");

}

if (xmlhttp!=null)

{

xmlhttp.onreadystatechange=state\_Change;

xmlhttp.open("GET",url,true);

xmlhttp.send(null);

}

else

{

alert("Your browser does not support XMLHTTP.");

}

}

function state\_Change()

{

if (xmlhttp.readyState==4)

{// 4 = "loaded"

if (xmlhttp.status==200)

{// 200 = "OK"

document.getElementById('T1').innerHTML=xmlhttp.responseText;

}

else

{

alert("Problem retrieving data:" + xmlhttp.statusText);

}

}

}

</script>

1. DButils（轻量化的JDBC封装工具）（注意强制转换时产生错误）

## QueryRunner（根据sql语句进行各项基本操作，具体参数可以看补全）

## Query

## Update

## Insert

## Delect

## ResultsetHandler

## ArrayHandler：把结果集中的第一行数据转成对象数组。

## ArrayListHandler：把结果集中的每一行数据都转成一个数组，再存放到List中。

## BeanHandler：将结果集中的第一行数据封装到一个对应的JavaBean实例中。

## BeanListHandler：将结果集中的每一行数据都封装到一个对应的JavaBean实例中，存放到List里。

## ColumnListHandler：将结果集中某一列的数据存放到List中。

## KeyedHandler(name)：将结果集中的每一行数据都封装到一个Map里，再把这些map再存到一个map里，其key为指定的key。

## MapHandler：将结果集中的第一行数据封装到一个Map里，key是列名，value就是对应的值。

## MapListHandler：将结果集中的每一行数据都封装到一个Map里，然后再存放到List