武汉纺织大学

Web应用开发课程设计

实验三

**学 院： 数学与计算机学院**

**班 级： 物联网11804**

**姓 名： 杨溪**

**学 号： 1804280403**

**指导老师： 聂刚**

**成 绩：**

**完成日期： 2020年12月18日**

# 1 作业要求

## 

# [JavaWeb分页查询的三种思路()](https://www.cnblogs.com/ygj0930/p/6134851.html)

    在开发过程中，经常做的一件事，也是最基本的事，就是从数据库中查询数据，然后在客户端显示出来。当数据少时，可以在一个页面内显示完成。然而，如果查询记录是几百条、上千条呢？直接一个页面显示完全的话，表格得多长啊。。。。。。这时，我们可以用分页技术。

    何为分页？效果图如下：



     这里总共查询了100条记录，如果一次性显示的话表格会很多行，用户体验不佳。而我们采用分页显示的话，一页显示10条记录，共十页。用户可以自行翻阅，记录少，清晰显示。

     下面谈谈分页效果的实现，思路有三种：

    其一：纯JS实现分页。一次性查询记录并加载到html的table中。然后通过选择性地显示某些行来达到分页显示的目的。这是一种伪分页，障眼法而已。只能用于数据少的情况下。一旦数据多了，十几万条数据加载到html中会变得很慢。而且不实时，一次加载完后数据就写死在页面了，若数据库中有变化，浏览器端显示的仍是上次加载过来的数据。

     首先：用table来显示查询出来的记录们，全部显示。

[复制代码](javascript:void(0);)

<table width="500" id="idData">

<%

String user\_id, user\_name, user\_sex, user\_phone, user\_age;

while (sqlRst.next()) {

user\_id = sqlRst.getString(1);

user\_name = sqlRst.getString(2);

user\_sex = sqlRst.getString(3);

user\_phone = sqlRst.getString(4);

user\_age = sqlRst.getString(5);

%>

<tr>

<td><%=user\_id%></td>

<td><%=user\_name%></td>

<td><%=user\_sex%></td>

<td><%=user\_phone%></td>

<td><%=user\_age%></td>

</tr>

<%

}

%>

</table>

<br/>

<table width="60%" align="right">

<tr><td><div id="changePages" name="changePages"></div></td></tr>

</table>

[复制代码](javascript:void(0);)

    然后，在JS中修改table中某些行显示，某些行隐藏。

[复制代码](javascript:void(0);)

<script type="text/javascript">

function goPage(pno,psize){

var itable = document.getElementById("idData");//获取table

var num = itable.rows.length;//得到记录总数

var totalPage = 0;

var pageSize = psize;//一页显示pageSize条记录

//计算总页数

if(num/pageSize > parseInt(num/pageSize)){

totalPage=parseInt(num/pageSize)+1;

}else{

totalPage=parseInt(num/pageSize);

}

//当前页数

var currentPage = pno;

//获取当前页第一条、最后一条记录的行号

var startRow = (currentPage - 1) \* pageSize+1;

var endRow = currentPage \* pageSize;

endRow = (endRow > num)? num : endRow;

//修改table中当前页对应的行的属性为显示，非本页的记录为隐藏

for(var i=1;i<(num+1);i++){

var irow = itable.rows[i-1];

if(i>=startRow && i<=endRow){

irow.style.display = "block";

}else{

irow.style.display = "none";

}

}

//分页页码列表

var tempStr = "共"+num+"条记录 分"+totalPage+"页 当前第"+currentPage+"页";

if(currentPage>1){

tempStr += "<a href=\"#\" onClick=\"goPage("+(1)+","+psize+")\">首页</a>";

tempStr += "<a href=\"#\" onClick=\"goPage("+(currentPage-1)+","+psize+")\"><上一页</a>"

}else{

tempStr += "首页";

tempStr += "<上一页";

}

if(currentPage<totalPage){

tempStr += "<a href=\"#\" onClick=\"goPage("+(currentPage+1)+","+psize+")\">下一页></a>";

tempStr += "<a href=\"#\" onClick=\"goPage("+(totalPage)+","+psize+")\">尾页</a>";

}else{

tempStr += "下一页>";

tempStr += "尾页";

}

document.getElementById("changePages").innerHTML = tempStr;

}

</script>

[复制代码](javascript:void(0);)

     其二：一次查询，分批显示。

     就是说，我们可以执行一个数据库查询操作，得到结果集rs。然后，通过指针的移动来显示当前页面的记录。这样，就可以以   rs.absolute(当前页面号\*每页记录数)定位到当前页的第一条记录，然后通过while循环显示n条记录（n为每页显示记录数）。在跳页时，只需修改currentPage，即可在重定位到下一页时把当前页面号改掉，重新定位记录指针，通过while遍历显示n条记录。与JS选择性显示不同，这里是选择性遍历。与JS分页不同的是，这里分页每次跳页修改的是遍历的指针，每次跳页都要进行一次全面查询。同样地，不适合大数据量查询。这里比JS分页优化的地方在于——实时性。每次跳页都会查询一次数据库，保证数据的实时性。

     参考代码：

[复制代码](javascript:void(0);)

　　　　　　　　　　<%      
　　　　　　　　　　 　int intPageSize = 10; //一页显示的记录数

int intRowCount; //记录总数

int intPageCount; //总页数

String strPage; //从表单或URL传送的待显示页码

int intPage; //待显示页码 ，由strPage转换成的整数

[复制代码](javascript:void(0);)

　　　　　　　　//---计算记录总数的第一种方法：查询出所有记录，移动结果集指针到最后一条，获取最后一条记录的行号  
　　　　　　　　　　//查询所有数据

　　　　　　　　ResultSet sqlRst = sqlStmt.executeQuery("select \* from user");

　　　　　　　　 //获取记录总数

　　　　　　　　　sqlRst.last(); //光标在最后一行

　　　　　　　　intRowCount = sqlRst.getRow(); //获得当前行号，即总记录数

　　　　　　　　 //记算总页数

　　　　　　　　 intPageCount = (int)Math.ceil(intRowCount/(intPageSize\*1.0));

[复制代码](javascript:void(0);)

//将记录指针定位到待显示页的第一条记录上

sqlRst.absolute((intPage - 1) \* intPageSize + 1);

//显示数据

int i=0;

String user\_id, user\_name, user\_sex, user\_phone, user\_age;

while (i < intPageSize && !sqlRst.isAfterLast()) {

user\_id = sqlRst.getString(1);

user\_name = sqlRst.getString(2);

user\_sex = sqlRst.getString(3);

user\_phone = sqlRst.getString(4);

user\_age = sqlRst.getString(5);

%>

<tr>

<td><%=user\_id%></td>

<td><%=user\_name%></td>

<td><%=user\_sex%></td>

<td><%=user\_phone%></td>

<td><%=user\_age%></td>

</tr>

<%

sqlRst.next();//移动记录指针指向下一条记录

i++;//统计当前页已显示多少条记录

}

%>

[复制代码](javascript:void(0);)

    其三：在服务端分页。跳到第n页才查询、显示第n页内容。要点就是根据客户端表格的“页面”计算出数据库要查询的当前页面的第一条记录的位置。优点：实时性：跳页才查询。数据量小：只加载当前页的记录进行显示。

    重点在于两条语句：select count(\*) from ...:查询得到记录总条数

                             select \* from .. limit pageNo,rowsCount:查询从第pageNo条开始的rowsCount条数据。

[复制代码](javascript:void(0);)

int pages=0; //待显示页面

int count=0; //总条数

int totalpages=0; //总页数

int limit=10; //每页显示记录条数

//计算记录总数的第二种办法：使用mysql的聚集函数count(\*)

ResultSet sqlRst = sqlStmt.executeQuery("select count(\*) from user");

if(sqlRst.next()){

count = sqlRst.getInt(1);//结果为count(\*)表，只有一列。这里通过列的下标索引（1）来获取值

}

//由记录总数除以每页记录数得出总页数

totalpages = (int)Math.ceil(count/(limit\*1.0));

//获取跳页时传进来的当前页面参数

String strPage = request.getParameter("pages");

//判断当前页面参数的合法性并处理非法页号（为空则显示第一页，小于0则显示第一页，大于总页数则显示最后一页）

if (strPage == null) {

pages = 1;

} else {

try{

pages = java.lang.Integer.parseInt(strPage);

}catch(Exception e){

pages = 1;

}

if (pages < 1){

pages = 1;

}

if (pages > totalpages){

pages = totalpages;

}

}

//由(pages-1)\*limit算出当前页面第一条记录，由limit查询limit条记录。则得出当前页面的记录

sqlRst = sqlStmt.executeQuery("select \* from user order by user\_id limit " + (pages - 1) \* limit + "," + limit);

while (sqlRst.next()){//遍历显示}

[复制代码](javascript:void(0);)

    跳页的实现：跳页是通过重定向来实现的，通过向当前网页传进待显示的pages，在跳转后根据pages重新算出页面显示的第一条，查limit条显示。

[复制代码](javascript:void(0);)

<form name="f1" method="POST" action="index.jsp" onSubmit="return checknum()">

<table border="0" align="center" >

<tr>

<td>第<%=pages%>页 共<%=totalpages%>页 <a href="index.jsp?pages=1">首页</a></td>

<td><a href="index.jsp?pages=<%=(pages<1)?pages:(pages-1) %>"> 上一页</a></td>

<td><a href="index.jsp?pages=<%=(pages>=totalpages)?totalpages:(pages+1)%>"> 下一页</a></td>

<td><a href="index.jsp?pages=<%=totalpages%>">最后一页</a></td>

<td>转到第:<input type="text" name="page" size="8">页<input type="submit" value="GO" name="cndok"></td>

</tr>

</table>   
 </form>

[复制代码](javascript:void(0);)

   附：常见数据库分页查询语句

[复制代码](javascript:void(0);)

1.oracle数据库分页

select \* from (select a.\*,rownum rc from 表名 where rownum<=endrow) a where a.rc>=startrow

2.DB2数据库分页

Select \* from (select rownumber() over() as rc,a.\* from (select \* from 表名 order by列名) as a) where rc between startrow and endrow

3.SQL Server 2000数据库分页

Select top pagesize \* from 表名 where 列名 not in(select top pagesize\*page 列名 from 表名 order by列名) order by列名

4.SQL Server 2005数据库分页

Select \* from (select 列名,row\_number() over(order by 列名1) as 别名from 表名) as t where t.列名1>=startrow and t.列名1<=endrow

5.MySQL数据库分页

Select \* from 表名 limit startrow,pagesize

(Pagesize为每页显示的记录条数)

6.PostgreSQL数据库分页

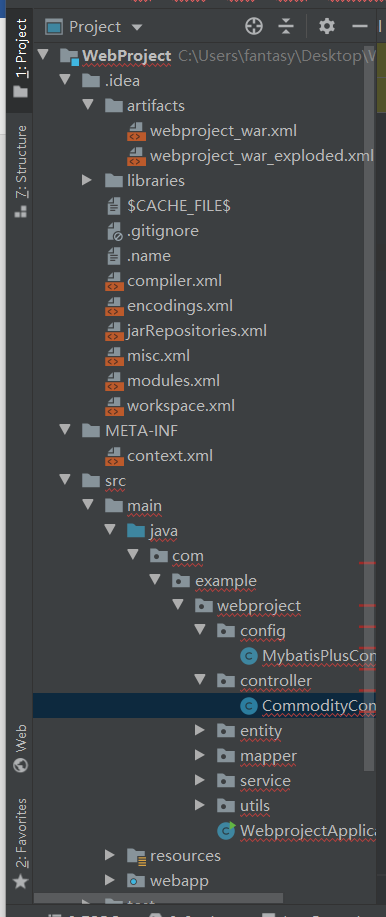
Select \* from 表名 limit pagesize,offset startrow

(Pagesize为每页显示的记录条数)

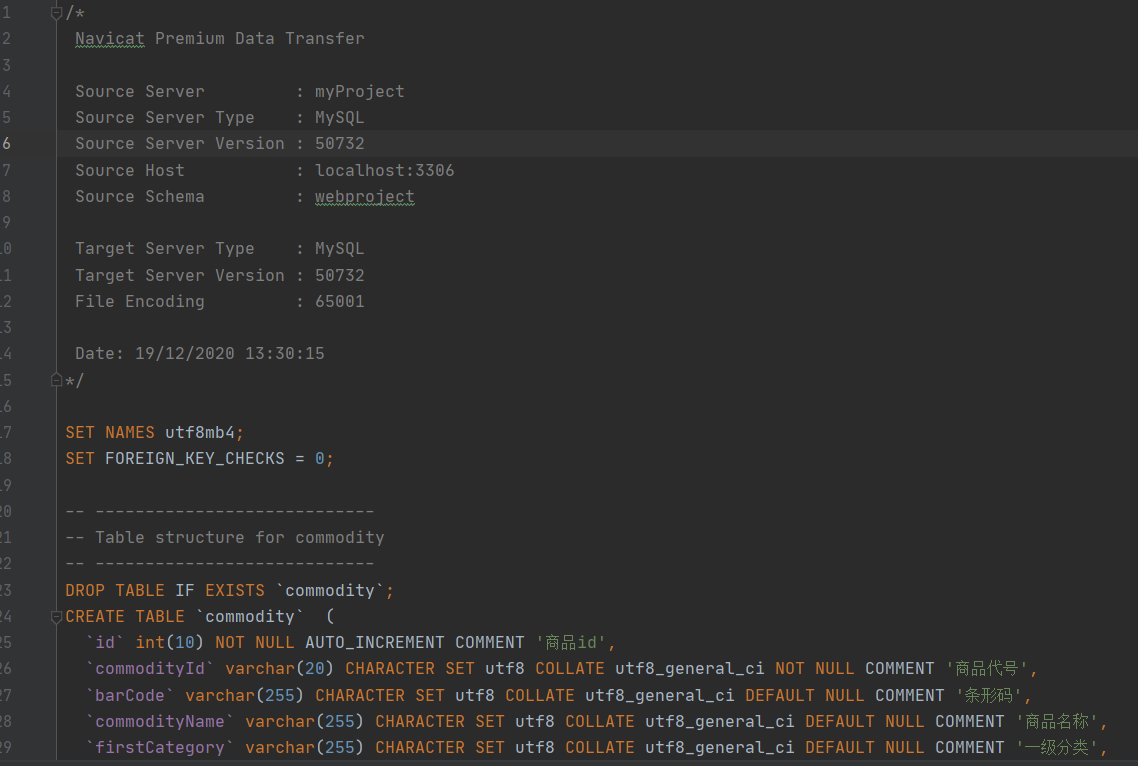
[复制代码](javascript:void(0);)

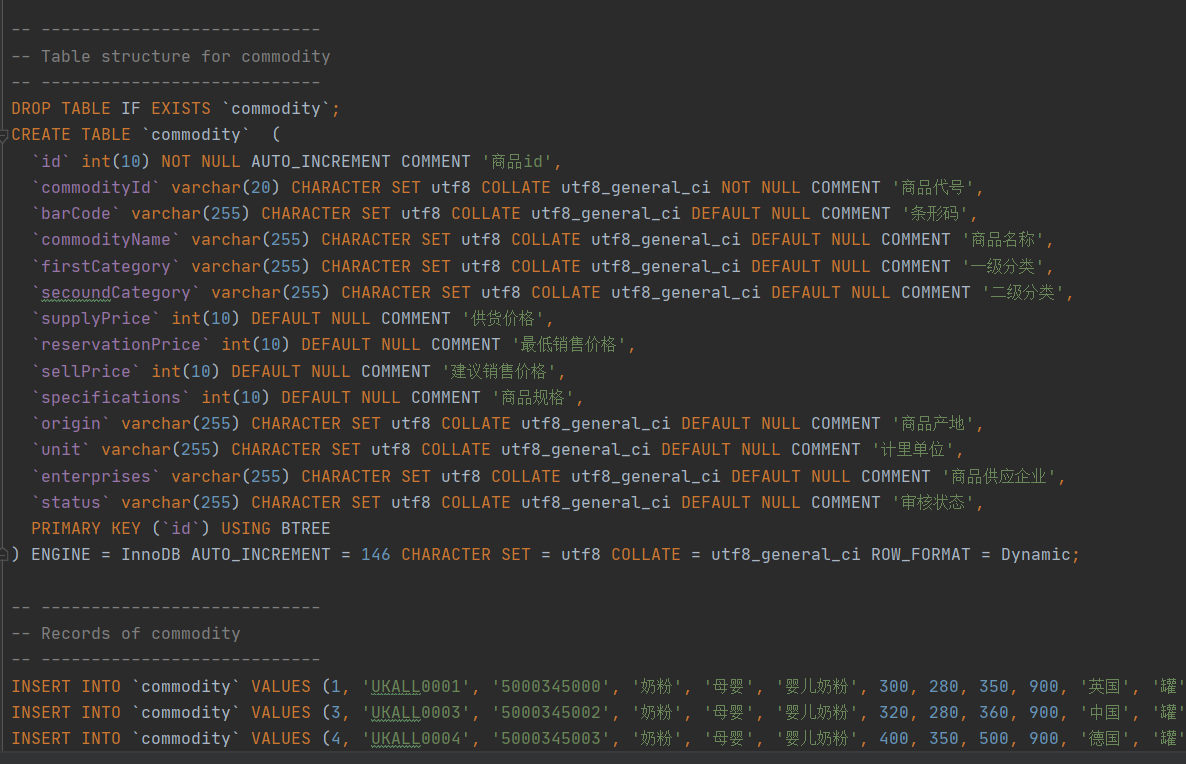
# 系统实现

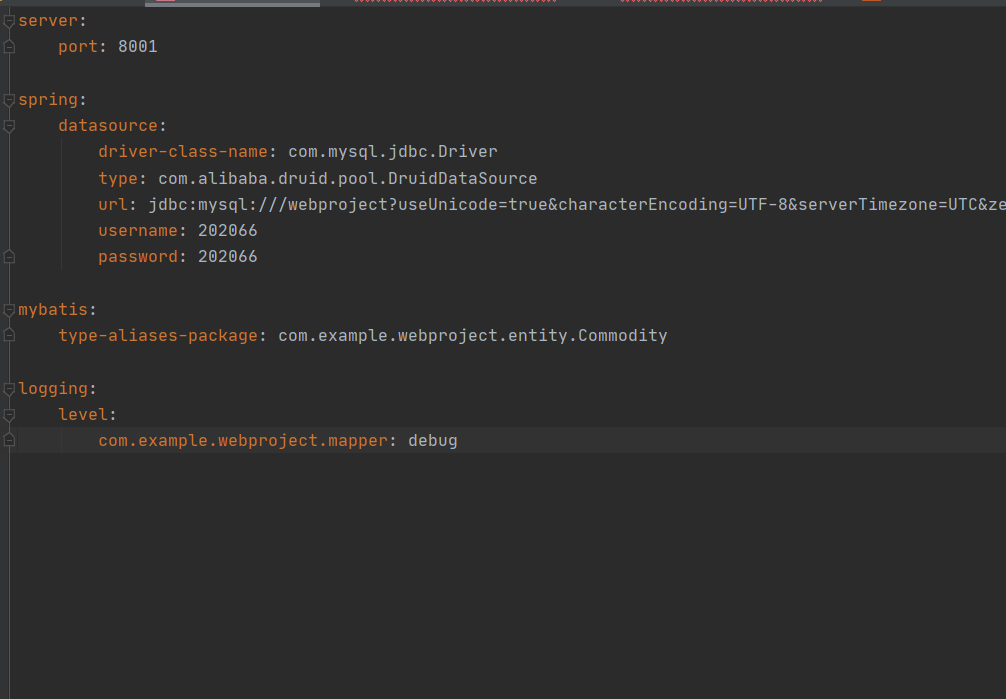
## 3.1 项目结构



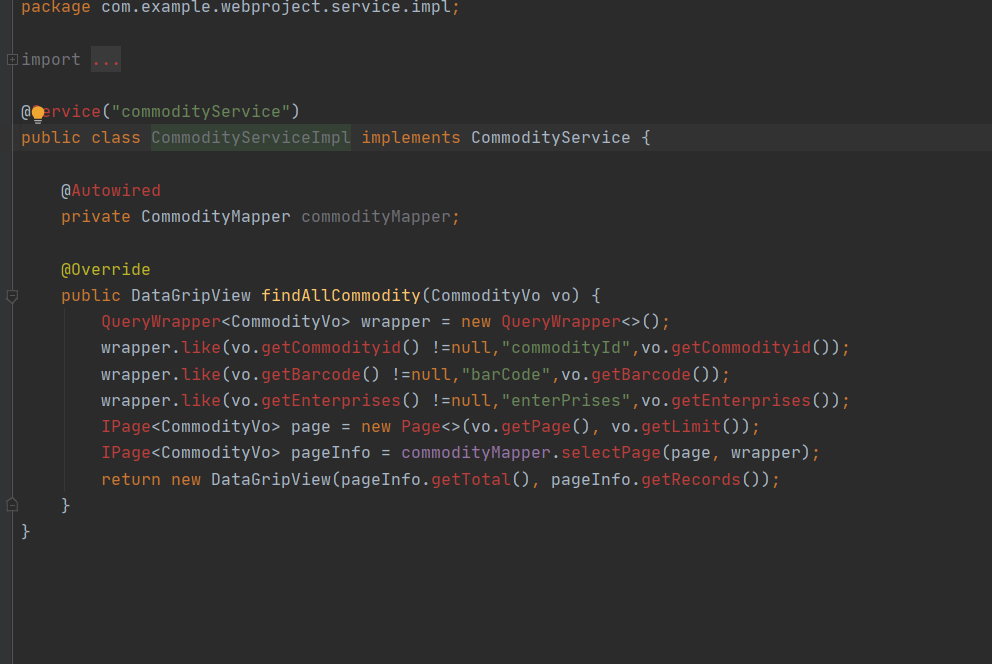
## 3.2 配置文件



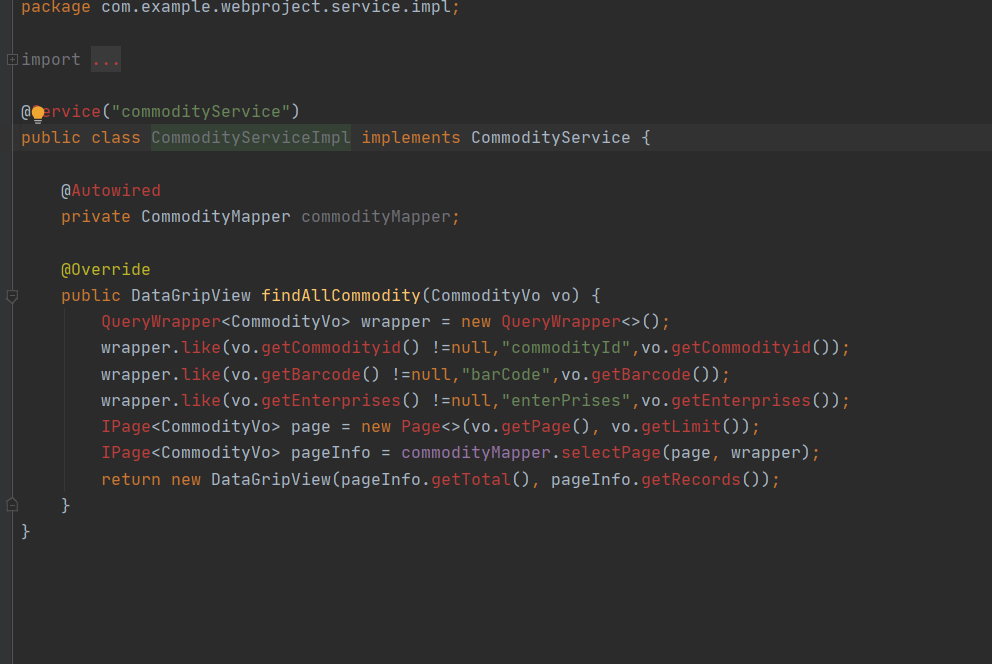


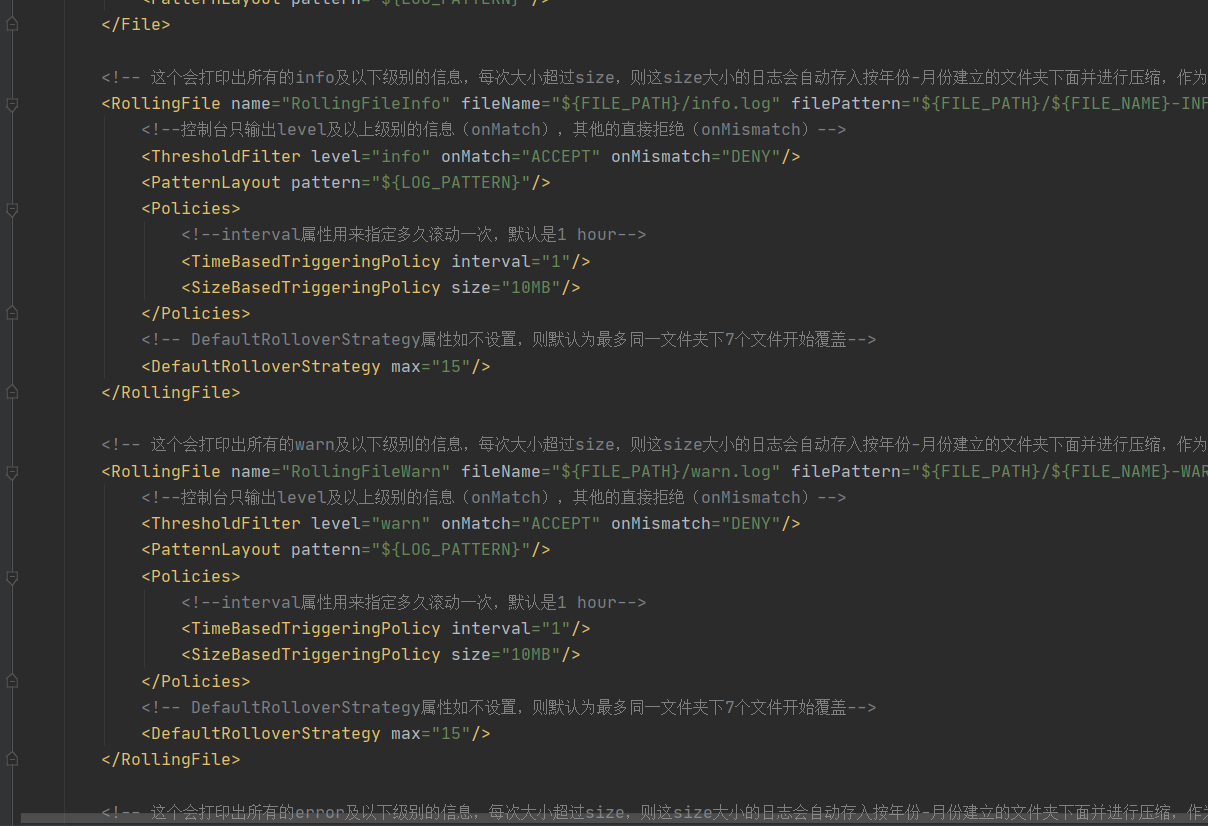


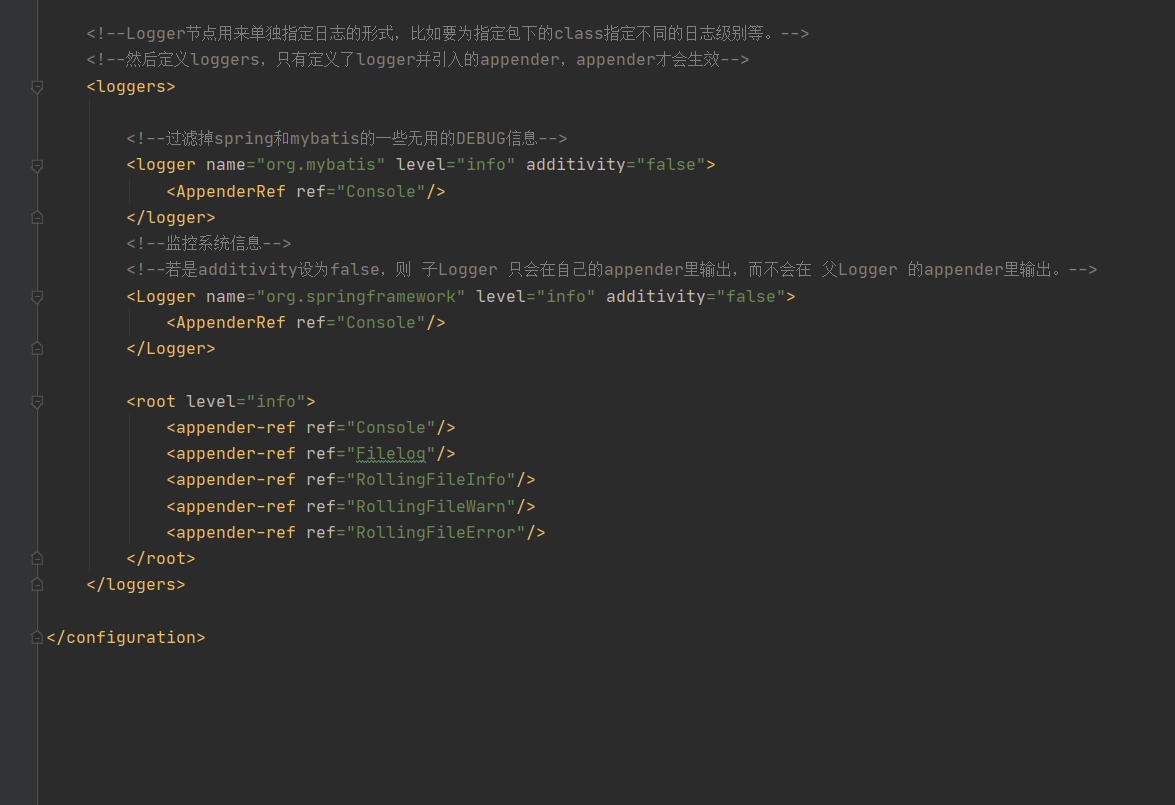


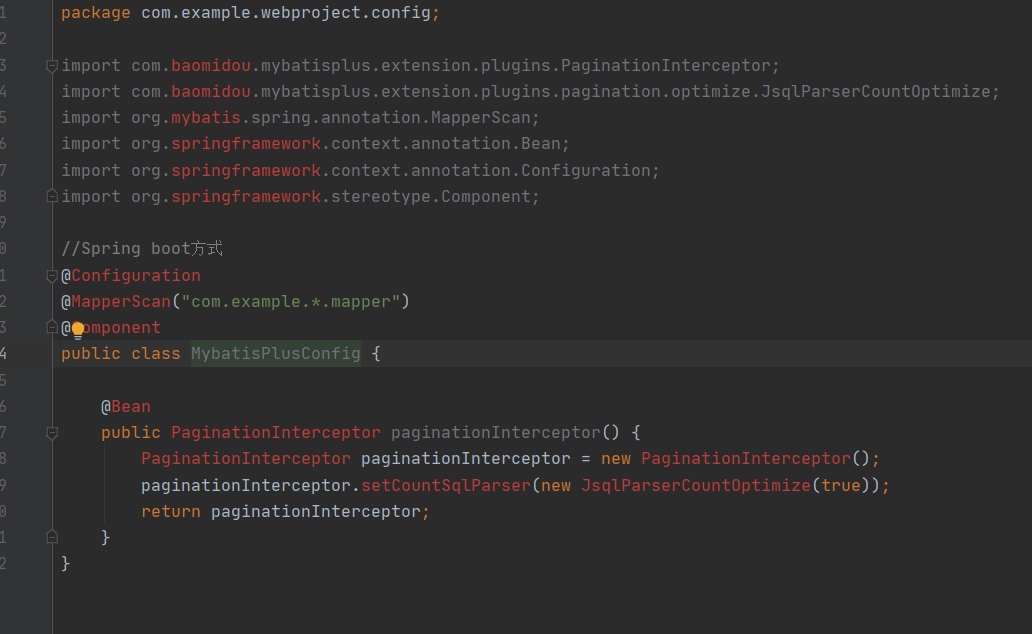












5总结

Web程序的开发我认为大约分4个层次:

1.表现层 (represent layer)

2.控制层 (logic control layer)

3.业务逻辑层 (service layer)

4.数据存储层 (persistent layer)

一个标准的系统大致就是做3件事,I(Input)P(Process)O(Output),也就是输入,处理,输出.

由于开发,部署,移植,性能和代码可重用性的考虑,Web开发将IPO分为了若干层次.:

1.表现层:

此层的主要作用是:向用户展示信息,并且得到用户输入数据和向用户展示处理后的反馈.

2.控制层:

此层的主要作用是:为了让开发人员和维护人员方便控制Web页面的流向,一目了然的对其走向进行控制.同时此层也可以进行一些简单的预处理,使业务逻辑避开本不该它们触碰的外部检测.此层的大部分任务是程序走向的控制,小部分任务是一般预处理和检测功能.

3.业务逻辑层:

此层的主要作用是:进行用户所要关心的业务逻辑,进行整个程序的核心业务处理,此层一般会使用从表现层传入的数据并调用数据存储层的接口来进行相应的查询和更新删除保存功能.并将最终处理结果反馈给控制层,由控制层根据处理结果去寻找表现给用户的路径.

4.数据存储层:

此层的主要作用是:进行数据的查询和持久化过程.