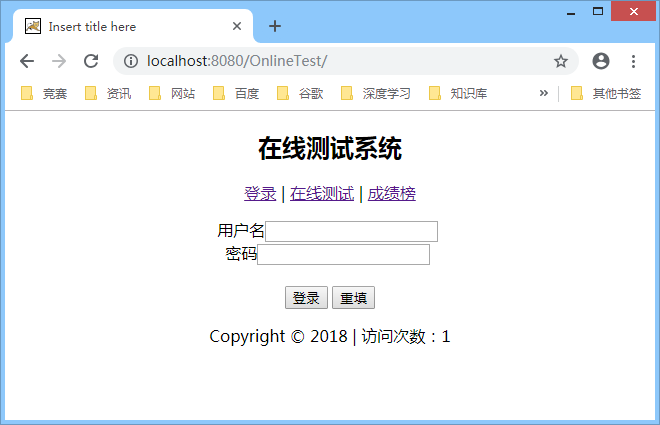
准备工作：新建Maven项目（不升级），添加index.jsp

PPT添加：header.jsp、footer.jsp适当修改



**1、index.jsp：登录页面**

表单代码：

=========================================

<form action="check.jsp" method="get">

用户名<input type="text" name="username" />

<br>

密码<input type="password" name="psd" />

<br><br>

<button type="submit">登录</button>

<button type="reset">重填</button>

</form>

=========================================

**2、check.jsp:检测页面**

(1)基本代码

=======================================

<%

String username=request.getParameter("username");

String psd=request.getParameter("psd");

if(username!=null&&username.equals("zz")&&psd!=null&&psd.equals("888")){

out.print("欢迎"+username);

out.print("&nbsp;&nbsp;<a href='test.jsp'>开始测试</a>");

}

else{

out.print("登录失败，3秒之后重新登录");

response.setHeader("refresh", "3;url='index.jsp'");

}

%>

=======================================

(2)添加Map存储用户信息：（代替数据库存储）

====================================

<%!

Map<String,String> user\_list=new HashMap<String,String>();

%>

<%

user\_list.put("zz", "888");

user\_list.put("yy", "666");

user\_list.put("qq", "222");

%>

====================================

(3)添加检测函数：

====================================

<%!

boolean check(String username, String psd) {

if (user\_list.containsKey(username))

if (user\_list.get(username).equals(psd)) {

return true;

}

return false;

}%>

====================================

**3、test.jsp：做题页面**

（1）题目代码：

====================================

<h3>在线测试题</h3>

<form action="submit.jsp" onsubmit="return confirm('确定提交吗？')">

第1题：湖北省会是

<input type="text" name="q1" />

<br><br>

第2题：宋朝开国皇帝是

<br>

<input type="radio" value="赵匡胤" name="q2">

赵匡胤

<input type="radio" value="朱元璋" name="q2">

朱元璋

<input type="radio" value="李渊" name="q2">

李渊

<br><br>

第3题：四大名著有

<br>

<input type="checkbox" value="红楼梦" name="q3">

红楼梦

<input type="checkbox" value="水浒传" name="q3">

水浒传

<input type="checkbox" value="J2EE编程技术" name="q3">

J2EE编程技术

<br><br>

<button type="submit">提交</button>

</form>

====================================

（2）添加session控制：解决2个问题（传值和页面访问控制）

先在check.jsp验证完成后添加： session.setAttribute("username", username);

====================================

<%

String username=(String)session.getAttribute("username");

if(username==null){

out.print("你还没有登录,3秒钟之后重新登录");

response.setHeader("refresh", "3;url='index.jsp'");

}

else{ //登录之后才能做题

%>

考生：<%=username %>

....//测试题目

<%}%>

====================================

4、submit.jsp：提交答题

（1）基本代码：

====================================

<%

int score=0;

String q1=request.getParameter("q1");

String q2=request.getParameter("q2");

String[] q3=request.getParameterValues("q3");

if(q1!=null&&q1.equals("武汉")){ score+=10; }

if(q2!=null&&q2.equals("赵匡胤")){ score+=10; }

if(q3!=null&&q3.length==2&&q3[0].equals("红楼梦")&&q3[1].equals("水浒传")){

score+=10; }

//out.print("<h2>你的成绩=" + score + "</h2>");

%>

====================================

改进：不显示成绩改为计分后退出：

response.sendRedirect("logout.jsp");

====================================

**添加logout.jsp页面：**代码

<%

session.invalidate() ; //注销 session

response.sendRedirect("index.jsp");

%>

====================================

（2）添加application对象存储所有考生成绩，给成绩榜listscore.jsp显示

====================================

<%!

Map<String, Integer> score\_list = new HashMap<String, Integer>(); //存放用户名+成绩

%>

之后：

score\_list.put((String) session.getAttribute("username"), score);

application.setAttribute("score\_list", score\_list);

====================================

**5、scorelist.jsp:**

====================================

（1）基本代码：遍历application

====================================

<h2>成绩榜</h2>

<%

if(application.getAttribute("score\_list")==null){

out.print("<h3>没有成绩</h3>");

}

else{ //遍历显示所有成绩（Map遍历）

Map<String, Integer> score\_list= (Map<String, Integer>)application.getAttribute("score\_list");

Set s=score\_list.keySet();

Iterator it=s.iterator();

while(it.hasNext()){

String username=(String)it.next();

int score=score\_list.get(username);

out.print("<h3>"+username+":"+score+"</h3>");

}

}

%>

====================================

（2）补充：给Map添加排序功能（降序）

定义函数：

====================================

<%!

//降序排序

public <K, V extends Comparable<? super V>> Map<K, V> sortByValueDescending(Map<K, V> map)

{

List<Map.Entry<K, V>> list = new LinkedList<Map.Entry<K, V>>(map.entrySet());

Collections.sort(list, new Comparator<Map.Entry<K, V>>()

{

public int compare(Map.Entry<K, V> o1, Map.Entry<K, V> o2)

{

int compare = (o1.getValue()).compareTo(o2.getValue());

return -compare;

}

});

Map<K, V> result = new LinkedHashMap<K, V>();

for (Map.Entry<K, V> entry : list) {

result.put(entry.getKey(), entry.getValue());

}

return result;

}

%>

score\_list遍历前调用一下即可：

score\_list = sortByValueDescending(score\_list);

====================================