1 后台的

try catch finally 有什么作用，划分为这三部分有什么好处？

在代码执行时，什么时候会用到try，什么时候会用到catch？finally呢？

```
try{

}
catch{

}
finally{

}
```

2

后台和前端交互信息时使用的代码实现是什么？

3

```
response.setContentType("text/html;charset=utf-8");
```

这行代码有什么作用？去掉可以吗，为什么？

```
PrintWriter out=response.getWriter();
```

这行代码呢？有什么作用？

4

如果进行搜索的话，使用了模糊搜索吗？是怎样使用的？

```
String name=request.getParameter("name");
        List<Play> result=null;
        if(name != null && name.length() > 0)
            result=new PlaySrv().Fetch(name);
        else
            result=new PlaySrv().FetchAll();
```

5

后台将数据传送给前端，使用的是什么格式？为什么要进行json到字符串的转化？不转化可以吗？为什么？

6 后台

后台有很多包，这些包的层次划分是什么？谁由谁封装？最上面的一层是什么？

7 前端的

这部分代码有什么作用or功能？

```
function init(){
      search();
      //不同页面修改这里默认选中样式
      $("#linkticket").css("background-color","#a2c05c");
      $("#sub1").attr("class","collapse panel-collapse in");
    }
```

这部分呢？

```
if (window.XMLHttpRequest)
          req = new XMLHttpRequest();
      else if (window.ActiveXObject)
          req = new ActiveXObject("Microsoft.XMLHTTP");
```

用于适配浏览器

在JS里，window是最顶级对象（除了Object，Function...那些之外），它代表了一个窗体。而window.XMLHttpRequest代表的是window的一个属性。这个是用来区分浏览器的，因为在firefox,opera,safiar,IE7.0,IE8.0（我所知道的window对象有这个属性的浏览器）这些浏览器中，window是有XMLHttpRequest这个属性的，而IE6.0，5.5都是没有的，IE6.0或5.5是没有这个属性的，使用window.ActiveXObject替代。当然前者和后者的XMLHttpRequest对象生成方式也是不一样的。

4 200

为什么是4 200？有什么含义？

```
function delComplete() {
    if (req.readyState == 4 && req.status == 200) {
      if(req.responseText==1)
        search();
      else
        alert("数据删除失败，请重试");
    }
  }
```

4：响应已完成；您可以获取并使用服务器的响应了。

200——交易成功

xmlhttp.readyState的值及解释： 0：请求未初始化（还没有调用 open()）。 1：请求已经建立，但是还没有发送（还没有调用 send()）。 2：请求已发送，正在处理中（通常现在可以从响应中获取内容头）。 3：请求在处理中；通常响应中已有部分数据可用了，但是服务器还没有完成响应的生成。 4：响应已完成；您可以获取并使用服务器的响应了。

xmlhttp.status的值及解释： 100——客户必须继续发出请求 101——客户要求服务器根据请求转换HTTP协议版本

200——交易成功 201——提示知道新文件的URL 202——接受和处理、但处理未完成 203——返回信息不确定或不完整 204——请求收到，但返回信息为空 205——服务器完成了请求，用户代理必须复位当前已经浏览过的文件 206——服务器已经完成了部分用户的GET请求

300——请求的资源可在多处得到 301——删除请求数据 302——在其他地址发现了请求数据 303——建议客户访问其他URL或访问方式 304——客户端已经执行了GET，但文件未变化 305——请求的资源必须从服务器指定的地址得到 306——前一版本HTTP中使用的代码，现行版本中不再使用 307——申明请求的资源临时性删除

400——错误请求，如语法错误 401——请求授权失败 402——保留有效ChargeTo头响应 403——请求不允许 404——没有发现文件、查询或URl 405——用户在Request-Line字段定义的方法不允许 406——根据用户发送的Accept拖，请求资源不可访问 407——类似401，用户必须首先在代理服务器上得到授权 408——客户端没有在用户指定的饿时间内完成请求 409——对当前资源状态，请求不能完成 410——服务器上不再有此资源且无进一步的参考地址 411——服务器拒绝用户定义的Content-Length属性请求 412——一个或多个请求头字段在当前请求中错误 413——请求的资源大于服务器允许的大小 414——请求的资源URL长于服务器允许的长度 415——请求资源不支持请求项目格式 416——请求中包含Range请求头字段，在当前请求资源范围内没有range指示值，请求也不包含If-Range请求头字段 417——服务器不满足请求Expect头字段指定的期望值，如果是代理服务器，可能是下一级服务器不能满足请求

合起来

500——服务器产生内部错误 501——服务器不支持请求的函数 502——服务器暂时不可用，有时是为了防止发生系统过载 503——服务器过载或暂停维修 504——关口过载，服务器使用另一个关口或服务来响应用户，等待时间设定值较长 505——服务器不支持或拒绝支请求头中指定的HTTP版本

1xx:信息响应类，表示接收到请求并且继续处理 2xx:处理成功响应类，表示动作被成功接收、理解和接受 3xx:重定向响应类，为了完成指定的动作，必须接受进一步处理 4xx:客户端错误，客户请求包含语法错误或者是不能正确执行 5xx:服务端错误，服务器不能正确执行一个正确的请求

7 同步与异步

同步的优点是：同步是**按照顺序**一个一个来，不会乱掉，更不会出现上面代码没有执行完就执行下面的代码， 缺点：是解析的速度没有异步的快；

异步的优点是：异步是接取一个任务，**直接给后台**，在接下一个任务，一直一直这样，谁的先读取完先执行谁的， 缺点：没有顺序 ， 谁先读取完先执行谁的 ， 会出现上面的代码还没出来下面的就已经出来了，会报错；