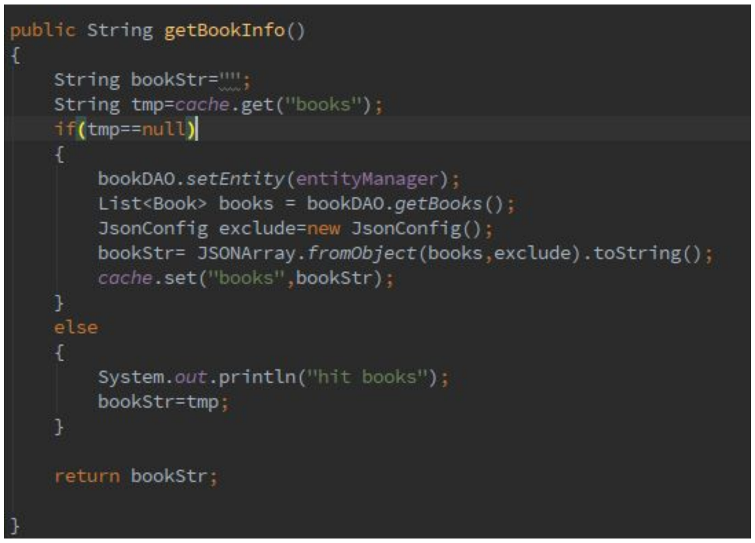
第三次迭代

孙 梦 5130379078

1. Applying MemCached or Redis in your application

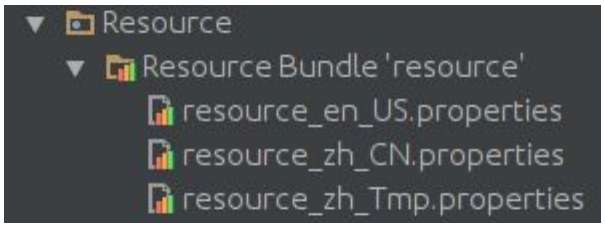
Solution: 采用 cache/cacheManager.java 中 cacheManager 类包装了 jedis 的 set 和 get，考虑到我的书店系统中在进入书店界面时，展示的上架书籍的简要信息数据是相对变化少的，所以在这一块应用了缓存设置。



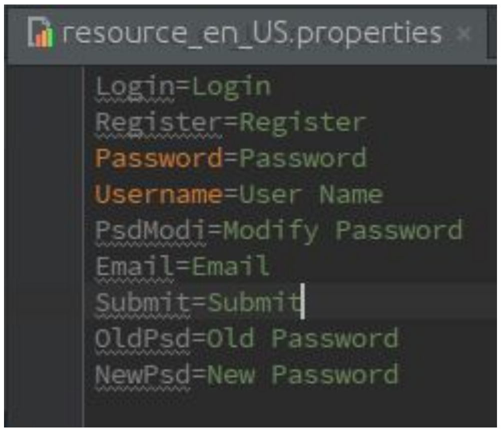
```
public String getBookInfo()
{
    String bookStr="";
    String tmp=cache.get("books");
    if(tmp==null)
    {
        bookDAO.setEntity(entityManager);
        List<Book> books = bookDAO.getBooks();
        JsonConfig exclude=new JsonConfig();
        bookStr= JSONArray.fromObject(books,exclude).toString();
        cache.set("books",bookStr);
    }
    else
    {
        System.out.println("hit books");
        bookStr=tmp;
    }

    return bookStr;

}
```

2. Refactoring your application to support internationalization

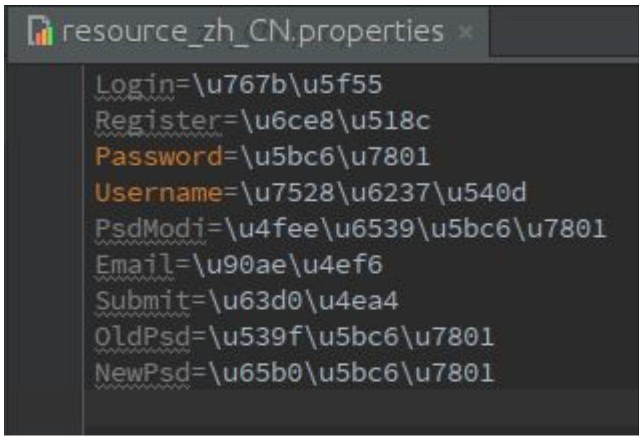
Solution： 设置了登录界面的中英文切换；添加了中英文对应的资源文件（其中 resource_zh_Tmp.properties 用于 native2ascii 工具转换的临时文件）



```
▼ Resource
    ▼ Resource Bundle 'resource'
        resource_en_US.properties
        resource_zh_CN.properties
        resource_zh_Tmp.properties
```

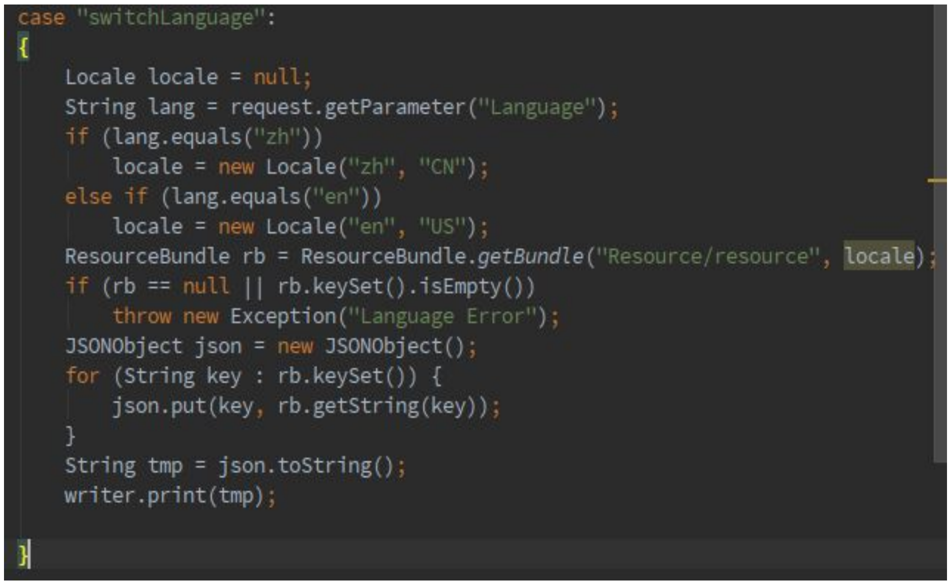

resource_en_US.properties

```
Login=Login
Register=Register
Password=Password
Username=User Name
PsdModi=Modify Password
Email=Email
Submit=Submit
OldPsd=Old Password
NewPsd=New Password
```

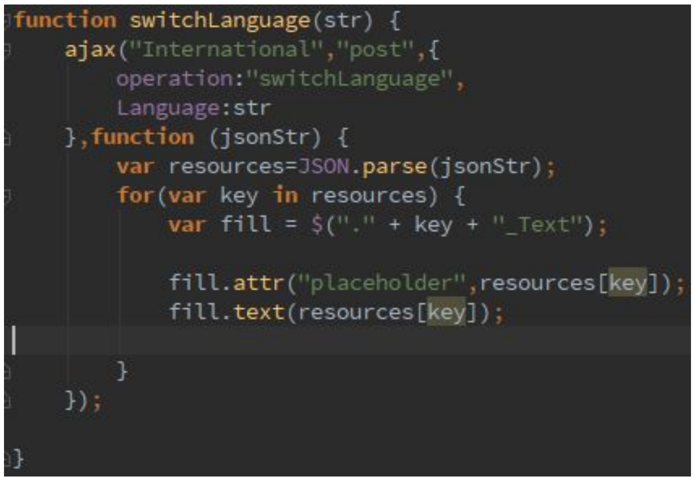

resource_zh_CN.properties

```
Login=\u767b\u5f55
Register=\u6ce8\u518c
Password=\u5bc6\u7801
Username=\u7528\u6237\u540d
PsdModi=\u4fee\u6539\u5bc6\u7801
Email=\u90ae\u4ef6
Submit=\u63d0\u4ea4
OldPsd=\u539f\u5bc6\u7801
NewPsd=\u65b0\u5bc6\u7801
```

初始设置页面语言为英文， ajax 响应切换。



```
case "switchLanguage":
{
    Locale locale = null;
    String lang = request.getParameter("Language");
    if (lang.equals("zh"))
        locale = new Locale("zh", "CN");
    else if (lang.equals("en"))
        locale = new Locale("en", "US");
    ResourceBundle rb = ResourceBundle.getBundle("Resource/resource", locale);
    if (rb == null || rb.keySet().isEmpty())
        throw new Exception("Language Error");
    JSONObject json = new JSONObject();
    for (String key : rb.keySet()) {
        json.put(key, rb.getString(key));
    }
    String tmp = json.toString();
    writer.print(tmp);

}
```

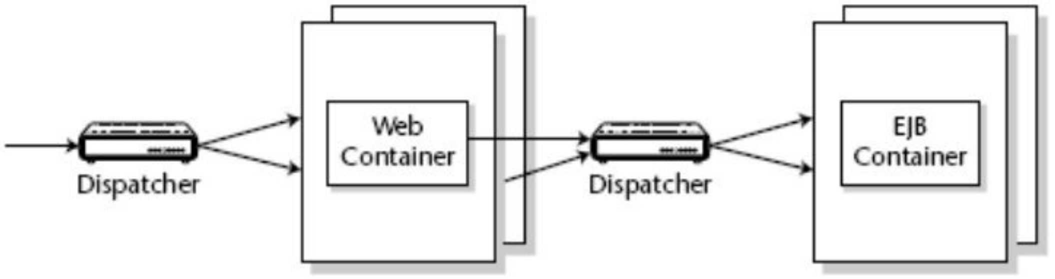


```
function switchLanguage(str) {
    ajax("International","post",{
        operation:"switchLanguage",
        Language:str
    },function (jsonStr) {
        var resources=JSON.parse(jsonStr);
        for(var key in resources) {
            var fill = $("." + key + "_Text");

            fill.attr("placeholder",resources[key]);
            fill.text(resources[key]);

        }
    });

}
```

3. Writing a design doc to describe the clustering solution you will adopt to scale

up your Book Store

Solution：采用 Distributed architecture



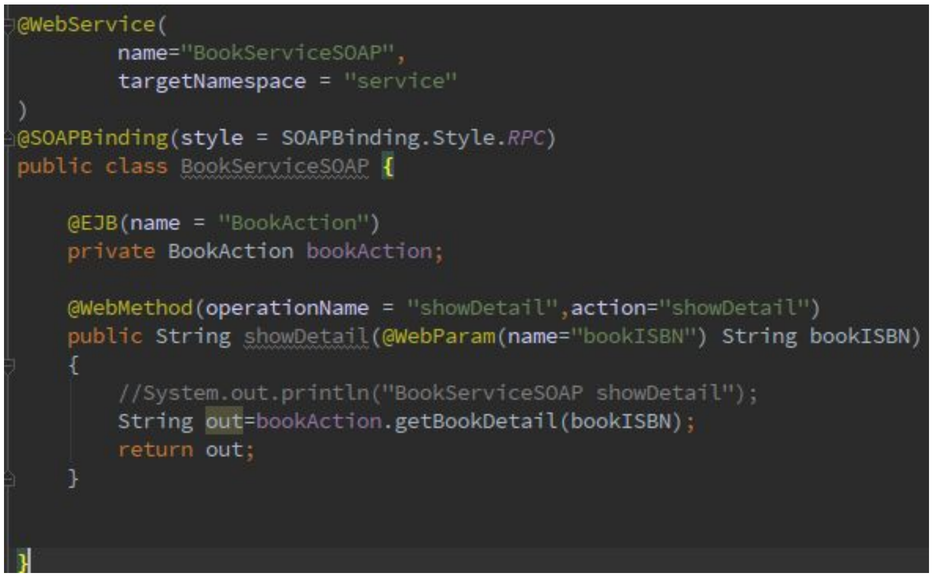
使用 nginx 实现负载均衡。对于物理机配置不同，可以采用对配置高的物理机分配多些的 tomcat 或 jboss 实例；如果所有物理机均匀分配，可以对配置高的物理机所在的 tomcat 或 jboss 实例设置相应比重的权重，在 dispatcher 分配的时候优先选择，这样可以提高 cpu 或是内存的利用率。

4. Developing an Aspect to implement logging

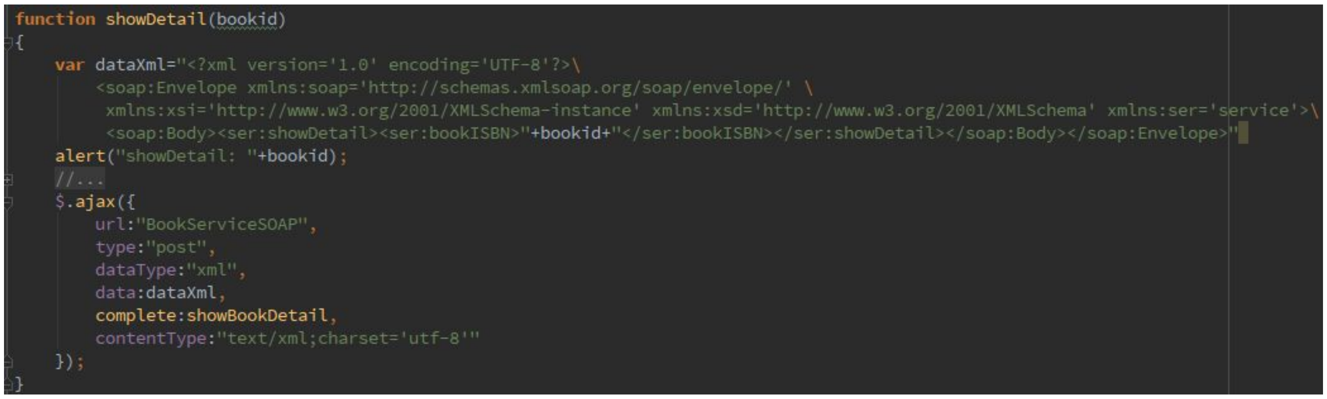
Solution：添加 aspectj/Log.aj
对每个 action 设置 pointcut，在执行前输出所在代码的位置和执行的函数名；在完成后输出返回结果。（代码详见 Log.aj）

5. Respectively developing a SOAP and a REST web service for a query

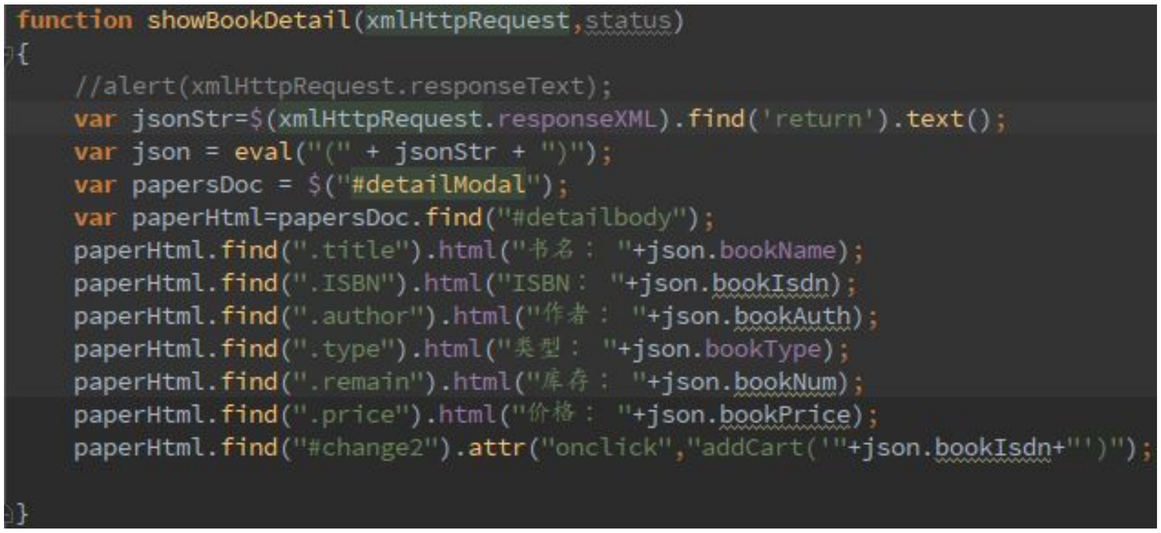
Solution：
1）SOAP：对书籍详情的浏览的响应采用 SOAP web service



```java
@WebService(
        name="BookServiceSOAP",
        targetNamespace = "service"
)
@SOAPBinding(style = SOAPBinding.Style.RPC)
public class BookServiceSOAP {

    @EJB(name = "BookAction")
    private BookAction bookAction;

    @WebMethod(operationName = "showDetail",action="showDetail")
    public String showDetail(@WebParam(name="bookISBN") String bookISBN)
    {
        //System.out.println("BookServiceSOAP showDetail");
        String out=bookAction.getBookDetail(bookISBN);
        return out;
    }


}
```

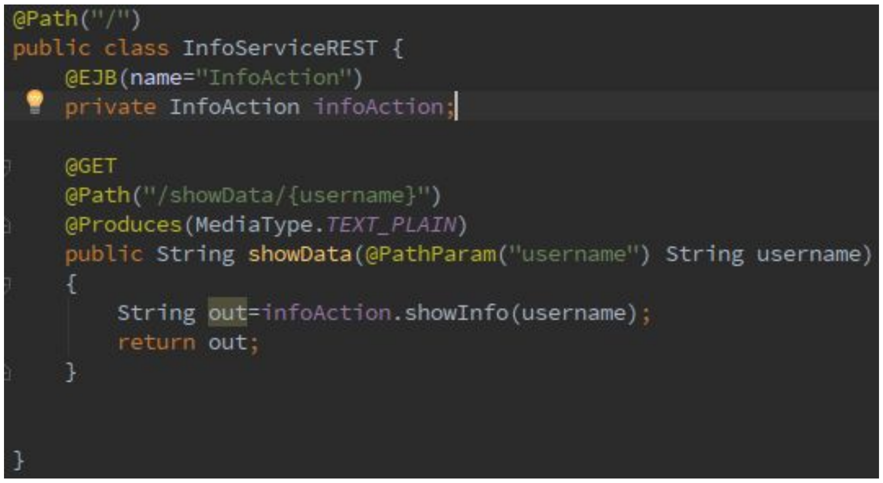


```javascript
function showDetail(bookid)
{
    var dataXml="<?xml version='1.0' encoding='UTF-8'?>\
        <soap:Envelope xmlns:soap='http://schemas.xmlsoap.org/soap/envelope/' \
         xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance' xmlns:xsd='http://www.w3.org/2001/XMLSchema' xmlns:ser='service'>\
         <soap:Body><ser:showDetail><ser:bookISBN>"+bookid+"</ser:bookISBN></ser:showDetail></soap:Body></soap:Envelope>"
    alert("showDetail: "+bookid);
    //...
    $.ajax({
        url:"BookServiceSOAP",
        type:"post",
        dataType:"xml",
        data:dataXml,
        complete:showBookDetail,
        contentType:"text/xml;charset='utf-8'"
    });
}
```

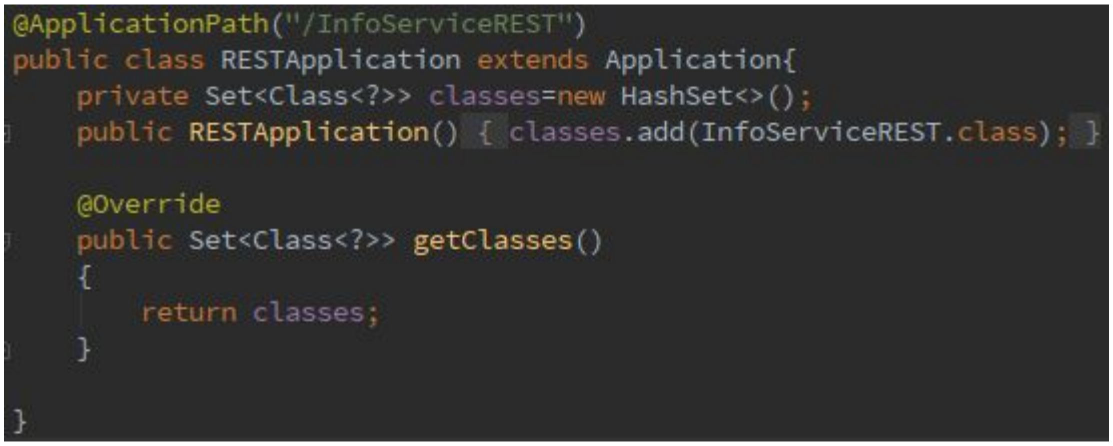


```javascript
function showBookDetail(xmlHttpRequest,status)
{
    //alert(xmlHttpRequest.responseText);
    var jsonStr=$(xmlHttpRequest.responseXML).find('return').text();
    var json = eval("(" + jsonStr + ")");
    var papersDoc = $("#detailModal");
    var paperHtml=papersDoc.find("#detailbody");
    paperHtml.find(".title").html("书名： "+json.bookName);
    paperHtml.find(".ISBN").html("ISBN： "+json.bookIsdn);
    paperHtml.find(".author").html("作者： "+json.bookAuth);
    paperHtml.find(".type").html("类型： "+json.bookType);
    paperHtml.find(".remain").html("库存： "+json.bookNum);
    paperHtml.find(".price").html("价格： "+json.bookPrice);
    paperHtml.find("#change2").attr("onclick","addCart('"+json.bookIsdn+"')");

}
```

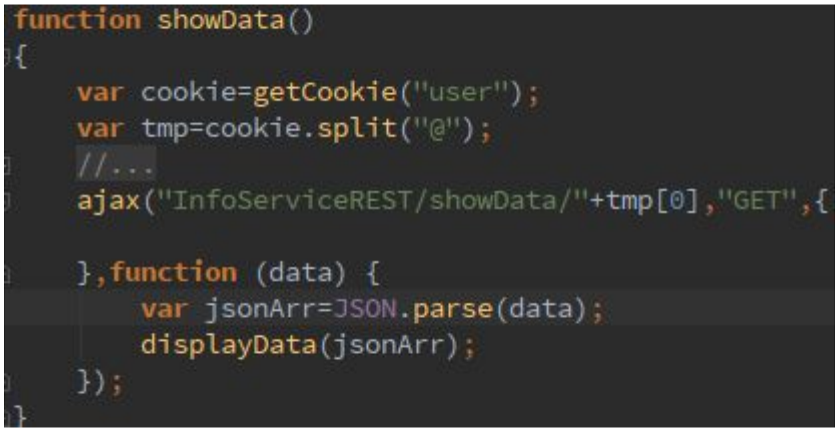
2）REST：对购物记录展示的响应采用 REST web service



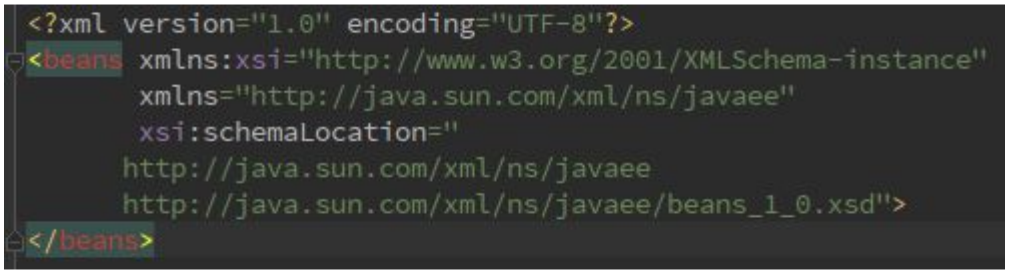
```java
@Path("/")
public class InfoServiceREST {
    @EJB(name="InfoAction")
    private InfoAction infoAction;

    @GET
    @Path("/showData/{username}")
    @Produces(MediaType.TEXT_PLAIN)
    public String showData(@PathParam("username") String username)
    {
        String out=infoAction.showInfo(username);
        return out;
    }


}
```



```java
@ApplicationPath("/InfoServiceREST")
public class RESTApplication extends Application{
    private Set<Class<?>> classes=new HashSet<>();
    public RESTApplication() { classes.add(InfoServiceREST.class); }

    @Override
    public Set<Class<?>> getClasses()
    {
        return classes;
    }

}
```



```javascript
function showData()
{
    var cookie=getCookie("user");
    var tmp=cookie.split("@");
    //...
    ajax("InfoServiceREST/showData/"+tmp[0],"GET",{

    },function (data) {
        var jsonArr=JSON.parse(data);
        displayData(jsonArr);
    });
}
```



```xml
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xmlns="http://java.sun.com/xml/ns/javaee"
       xsi:schemaLocation="
      http://java.sun.com/xml/ns/javaee
      http://java.sun.com/xml/ns/javaee/beans_1_0.xsd">
</beans>
```

Docker: lyndocker/bookstore:3