书单管理子系统：

包含功能：个人新建书单、个人修改书单、书单定向（广播）推荐、个人所有书单查看、单个书单详细信息查看

书单信息包含：一个可以修改的书单名、一个可以修改的针对本书单的描述、书单创建时间、书单最后修改时间、一个可以修改（增加、删除书籍）的书籍列表。

书单信息定义（readings）：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段 | 数据类型 | IS NULL | 默认值 | 备注 |
| id | varchar(36) | N | 自增 | 书单ID（UUID），主键自增 |
| name | varchar(36) | N | 无 | 书单名 |
| description | varchar(255) | N | 无 | 书单描述 |
| reader\_id | bigint | N | 无 | 书单创建者ID |
| create\_time | datetime | N | getdate() | 书单创建时间 |
| update\_time | datetime | Y |  | 修改书单名/书单描述/增删书籍时更新此字段 |

书单里的书籍定义（readings\_books）：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段 | 数据类型 | IS NULL | 默认值 | 备注 |
| id | bigint | N | 自增 | 主键自增 |
| readings\_id | varchar(36) | N | 无 | 书单编号 |
| book\_id | varchar(36) | N |  | 书籍编号 |
| create\_time | datetime | N | getdate() | 书籍编入书单时间 |
| deleted | tinyint | N | 0 | 删除标志：  0：未删除  1：已删除 |
| delete\_time | datetime | Y | 无 | deleted=1时，该字段有效 |

书单推荐（readings\_recommand）：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段 | 数据类型 | IS NULL | 默认值 | 备注 |
| id | bigint | N | 自增 | 主键自增 |
| readings\_id | varchar(36) | N | 无 | 书单编号 |
| reader\_id | bigint | N | 无 | 读者编号 |
| create\_time | datetime | N | getdate() | 该推荐创建时间 |
| deleted | tinyint | N | 0 | 0：未删除  1：已删除 |
| delete\_time | datetime | Y | 无 | deleted=1时，该字段有效 |

**新建书单**

**前台设计**

**前后台接口**（JSON）→

Request:

{

“name”:”policy\_angle”, //书单名

“description”:”政治经济”, //书单描述

“reader\_id”:”112345”, //书单创建者ID

“bookList”:[ //书单里包含的书籍列表；该字段可能为空

“book1”,

“book2”

]

}

Response:

{

“code”:”0”, //返回码：0为创建书单成功；其他为创建书单失败

“message”:”codeDescription” //返回码描述

}

**后台处理**

1. 使用struts2的参数传递特性，获取到对应的书单信息
2. 检索数据库里的数据，查看reader下有没有同名书单
3. 如果有同名书单，更新该记录
4. 如果没有同名书单，插入数据到数据库
5. 封装响应消息，返回给前台

**修改书单**

**前台设计**

**前后台接口**（JSON）→

Request：

{

“readings\_id”:”readings\_1”, //书单ID

“name”:”policy\_angle”, //书单名

“description”:”政治经济”, //书单描述

“reader”:”112345”, //书单创建者ID

“bookList”:[ //书单里包含的书籍列表；该字段可能为空

“book1”,

“book2”

]

}

Response：

{

“code”:”0”, //返回码：0为创建书单成功；其他为创建书单失败

“message”:”codeDescription” //返回码描述

}

**后台处理**

1. 使用struts2的参数传递特性，获取到对应的书单信息
2. 检索数据库里的数据，查看reader下有没有该书单
3. 如果有同名书单，更新该记录；bookList的书籍如果已存在于该书单中，不处理，如果没有存在于该书单中，加入到该书单。
4. 如果没有同名书单，返回错误
5. 封装响应消息，返回给前台

**书单定向（广播）推荐**

**前台设计**

**前后台接口**（JSON）→

Request：

{

“readings\_id”:”readings\_1”, //书单ID

“type”:”0”, //0：定向 1：广播

“reader\_id”:”1234” //当type为0时，该字段有效；type为1时，该字段不存在

}

Response：

{

“code”:”0”, //返回码

“message”:”codeDescription” //返回码描述

}

**后台处理**

1. 使用struts2的参数传递特性，获取到对应的书单广播信息
2. 校验readings\_id有效性，查看数据库中是否存在该书单
3. 如果type=0，校验reader有效性，查看数据库中是否存在该读者
4. 插入数据到readings\_recommand中，如果type=0，写入reader\_id字段；如果type=1，reader\_id字段为空
5. 封装返回信息，返回给前台

**书单列表查看**

**前台设计**

**前后台接口**（JSON）→

Request：

{

“reader\_id”:”123”

}

Response：

{

“code”:”0”, //返回码：如果返回码为0，readings\_list字段才有效

“message”:”codeDescription”, //返回码描述

“readings\_list”:[

{

“name”:”test1234”,

“description”:”test123456”,

“create\_time”:”2016-01-22 12:42:00”,

“update\_time”:”2016-01-22 12:42:00”

},

{

“name”:”test11234”,

“description”:”test1123456”,

“create\_time”:”2016-01-21 12:42:00”,

“update\_time”:”2016-01-21 12:42:00”

}

]

}

**后台处理**

1. 使用struts2的参数传递特性，获取到对应的读者信息
2. 校验reader\_id有效性，查看数据库中是否存在该读者
3. 如果存在该读者，查询数据库中该读者下的所有书单
4. 根据查询到的书单，封装返回信息，返回给前台。

**书单信息查看**

**前台设计**

**前后台接口**（JSON）→

Request：

{

“readings\_id”:”readings\_12345”

}

Response：

{

“bookList”:[ //书籍列表

{

“book\_id”:””,

“name”:””

}

]

}

**后台处理**

1. 使用struts2的参数传递特性，获取到对应的书单信息
2. 校验readings\_id有效性，查看数据库中是否存在该书单
3. 如果存在该书单，查询数据库中该书单下的所有书籍
4. 根据查询到的书籍列表，封装返回信息，返回给前台