需要redis，mysql，springboot，一个云储存服务器阿里OSS，用来存放图片，pdf资源

**前后端同一约定格式**

{

code：状态码：20000表示处理成功，20001表示处理失败，20002表示服务异常

message：一些补充信息，出现异常时携带，例如注册时用户名重复

data：{ 返回的数据集

}

}

当前端收到错误的状态码的时候，可以使用message打印错误信息

输入格式由前端校验，后端只需要接收

**用户相应操作模块：**

user表

{

uid：自增

userName:String

passWord:String,(MD5)加密

mail:String --前端校验是否为@结尾

phoneNumber:String --前端校验是否合法

}

提供用户登录，用户注册功能，其中密码找回采用邮件加验证码的形式

后端提供api：

**登录**

8001：/usercenter/login

post方式，请求格式：{

userName:xxx，

passWord:xxx

}

说明：该方法需要验证userName是否存在，验证passWord是否正确，成功之后，需要生成一个token，token内容是通过uid加密得到（这种加密一般可以使用JWT，参考https://blog.csdn.net/Mr\_Dracy/article/details/116738663），返回数据格式为

{

**. . .**

data:{

token:xxx

}

}

登录之后前端保留这个token，以cookie的形式记录下来，下一次访问就直接在url中附加这个token，后端每一次解密这个token从而获得用户id

登录成功后跳转到主页面

**注册**

8001: /usercenter/regist

post方式，请求格式：{

userName:

passWord:

mail:

phoneNumber:  
 }

后端验证注册时，要检查是否已经有重复用户名，如果有的话就终止操作，如果注册成功，就将信息放入数据库中，其中密码这一项需要经过MD5加密之后再放入数据库

前端获得返回结果，然后根据状态码进行注册判断，如果成功跳转至登录页面

**密码找回**

密码找回，首先是要通过邮件发送验证码，用户只有在输入正确验证码之后才能进行密码修改，验证码有效时长设置为5分钟，通过redis缓存存放。涉及到两个函数

8001：/email/sendcode --发送验证码

GET方式，后端随机生成一个六位数字，存入redis后，通过邮件发送到对方邮箱中

8001：/email/checkandchange --根据用户id从redis取出验证码验证后修改密码

POST方式，{

password:xxx

}

返回{

**....**

}

**查询用户个人信息**

8001：/usercenter/getuserinfo

GET

返回类型

userInfo:{

uid：

userName:String

mail:String

phoneNumber:String

}

根据url的token获得id之后查表获得完整的用户信息，注意，密码不需要

**图书服务模块：**

**表设计**

book{

bookID 自增

categoryID 分类ID

bookName：

author:

bookPrice： 默认0

createDate: 图书添加时间

isBorrowed: 是否借出 默认false

discreption：图书描述

bookNum：图书的数量 默认0

cover： 一个url，指向一个地址，存放图书封面的链接

pdf： 一个url，主要是用来存放图书pdf链接

}

借阅表

borrow{

userID

bookID

createDate 记录创建时间

return 是否归还，0未还，1已还

}

分类表：

category{

categoryID 自增

categoryName

parentID 父分类的categoryID

}

限制：分类最多两级

后端提供的api

**上传图书封面**

8002：/bookservice/uploadcover

post方法，接口参数为MultipartFile，接收前端上传的图片，然后将其上传到云端的储存器中，返回上传后得到的存储url，并发给前端

**上传图书pdf**

8002：/bookservice/uploadbookpdf

post方法，接口参数为MultipartFile，接收前端上传的pdf文件，然后将其上传到云端的储存器中，返回上传后得到的储存url，并发给前端

**添加分类**

8002：/bookservice/addcategoryinfo

post方法

请求格式：

{

categoryname:xxx,

parentname:xxx,

}

**获得分类信息**

8002：/bookservice/getbookcategoryinfo

GET

返回格式

{

code:xxx

message:xxx

data:{

parentCategorys:

[

{

categoryID:xxx

categoryName:xxx

children:[

{

categoryID:xxx

categoryName:xxx

},

{

categoryID:xxx

categoryName:xxx

},

]

}，

{

categoryID:xxx

categoryName:xxx

children:[

{

categoryID:xxx

categoryName:xxx

},

{

categoryID:xxx

categoryName:xxx

},

]

}，

…

]

}

}

**上传图书信息**

8002：/bookservice/addbookinfo

post 方法，参数

{

categoryID 分类ID

bookName：

bookPrice：

author:

discreption：图书描述

bookNum：图书的数量

bookRemain ：剩余数量

cover： 一个url，指向一个地址，存放图书封面的链接

pdf： 一个url，主要是用来存放图书pdf链接

}

储存时，如果图书名已存在，就终止并返回message

**修改图书信息**

8002: /bookservice/modifybookinfo

post方法，参数

{

bookID

categoryID 分类ID

bookName：

bookPrice：

author:

discreption：图书描述

bookNum：图书的数量

bookRemain ：剩余数量

cover： 一个url，指向一个地址，存放图书封面的链接

pdf： 一个url，主要是用来存放图书pdf链接

}

通过bookID找到对应记录行然后更改

-----------------------------------------------------------------------------

**通过图书id获得图书信息**

8002：/bookservice/getbookinfobyID/{bookID}

GET

{

bookID 自增

categoryID 分类ID

bookName：

author:

bookPrice： 默认0

createDate: 图书添加时间

isBorrowed: 是否借出 默认false

discreption：图书描述

bookNum：图书的数量 默认0

cover： 一个url，指向一个地址，存放图书封面的链接

pdf：

}

**获得图书信息,带分页功能**

8002：/bookservice/getbook/{page}/{limit} //page，查询第page页；limit，一页有limit条记录

GET

{

bookID 自增

categoryID 分类ID

bookName：

author:

bookPrice： 默认0

createDate: 图书添加时间

isBorrowed: 是否借出 默认false

discreption：图书描述

bookNum：图书的数量 默认0

cover： 一个url，指向一个地址，存放图书封面的链接

pdf：

}

**获得图书信息，通过搜索方式**

8002: /bookservice/booksearch

POST

{

searchName

categoryID

page

limit

}

后端通过传入数据做条件查询，如果其中某个为空，则不使用，searchName用模糊查询

-----------------------------------------------------------------------------

**借阅图书**

8002:/bookservice/borrowbook/{bookID}

GET

后端判断借阅的时候，首先检查图书是否还有库存，没有则终止，其次，检查借阅者是否已经借阅了五本书，是的话就不允许借阅，都检查过后才能借阅。借阅失败的话要在返回的message中添加失败的原因，成功了记得同步图书表对应的剩余数量，然后向借阅关系表中放一条记录

用户ID从url的token中获得

-----------------------------------------------------------------------------

**归还图书**

8002：/bookservice/returnbook/{bookID}

GET

归还的时候，根据url的token获得用户ID，然后记得同步图书表对应的剩余数量，借阅关系表中的记录不必移除，而是将return标记为1，这样可以让用户查看所有历史借阅记录

----------------------------------------------------------------------------

**查看某一用户所有借阅记录**

8002: /bookservice/getborrowedhistory

GET

{

bookName

author

price

createDate //创建这个记录的时间，实际上就是借阅的时间

}

根据url中的token获得用户ID