## User 接口

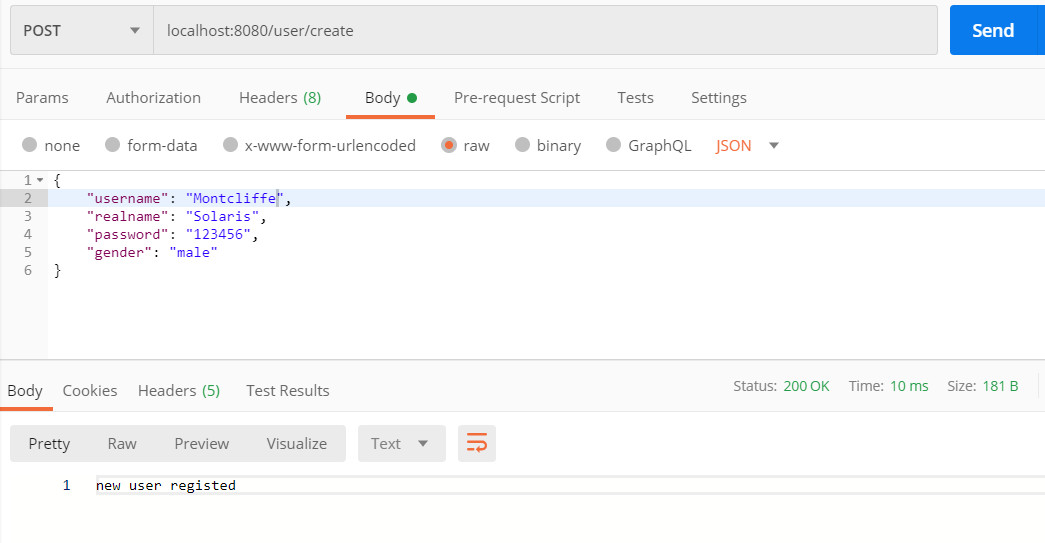
### 创建用户

* 请求方法：POST
* 路径：localhost:8080/user/create
* 参数：在body中传入username, realname, password, gender
* 返回结果：

如果成功，返回"new user registered"；

如果用户名已存在，返回"already existed username"；

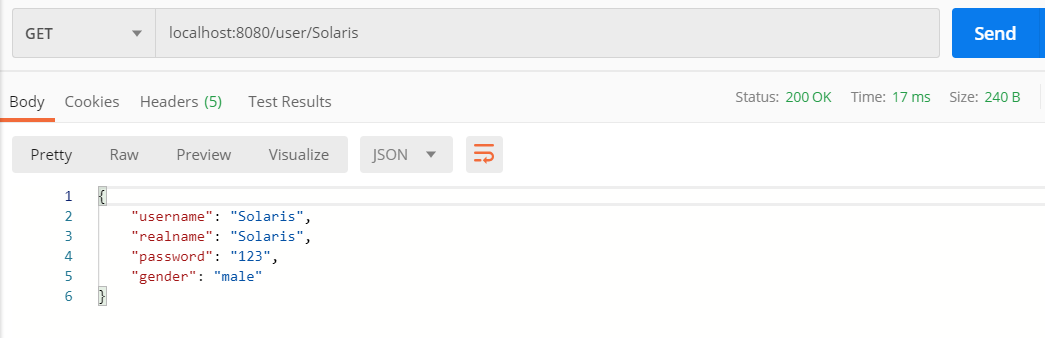
示例：



### 根据 username 查询用户

* 请求方法： GET
* 路径：localhost:8080/user/{username}
* 参数： 在URL添加username参数
* 返回结果：返回全部user信息；如果无匹配结果，返回空值

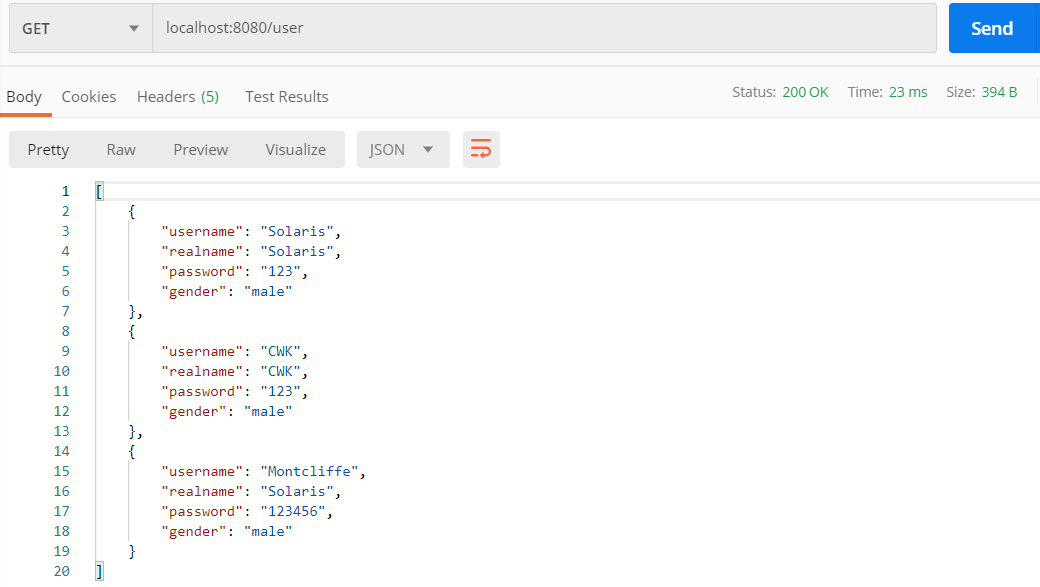
示例：



### 查询全部 user

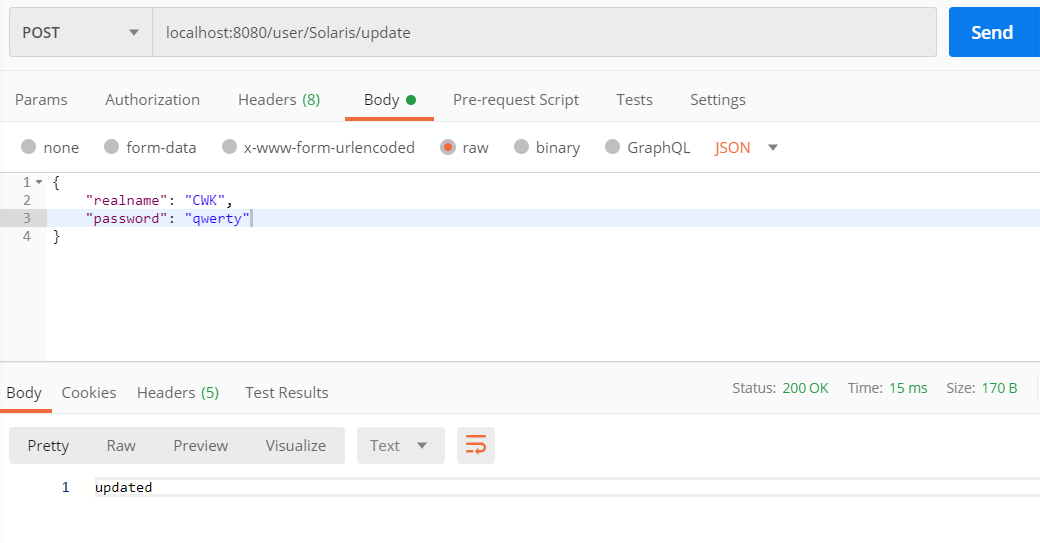
* 请求方法：GET
* 路径： localhost:8080/user
* 参数： 无
* 返回结果： 全部 user 信息

示例：



### 更新用户信息

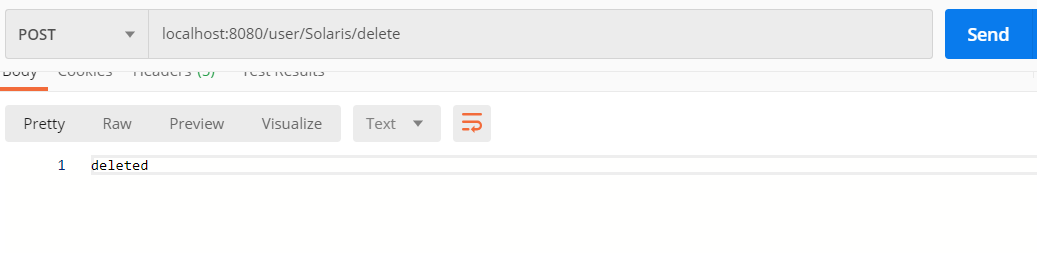
* 请求方法： POST
* 路径： localhost:8080/user/{username} /update
* 参数： 在路径中添加username 参数；在body中加入要修改的数据
* 返回结果：如果成功，返回“updated’; 如果没有匹配的用户，返回”no match”



### 删除用户

* 请求方法： POST
* 路径： localhost:8080/user/{username} /delete
* 参数： 在路径中添加username 参数；在body中加入要修改的数据
* 返回结果：如果成功，返回“deleted; 如果没有匹配的用户，返回”no match”

示例：

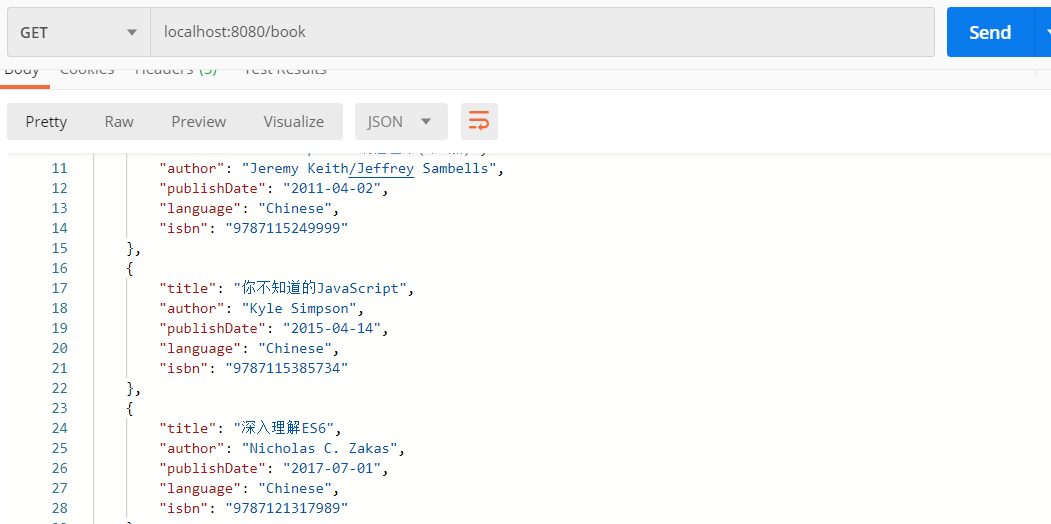


## Book接口

### 查看全部书籍

* 请求方法：GET
* 路径： localhost:8080/book
* 参数： 无
* 返回结果： 全部书籍信息

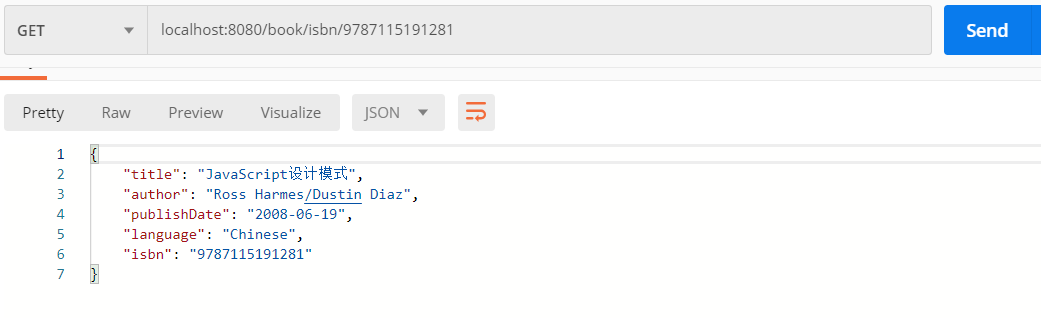
示例：



### 根据 ISBN 查询书籍

* 请求方法： GET
* 路径： localhost:8080/book/isbn/{ISBN}
* 参数：在路径中添加 ISBN
* 返回结果：书籍信息；如果没有匹配结果，返回空值

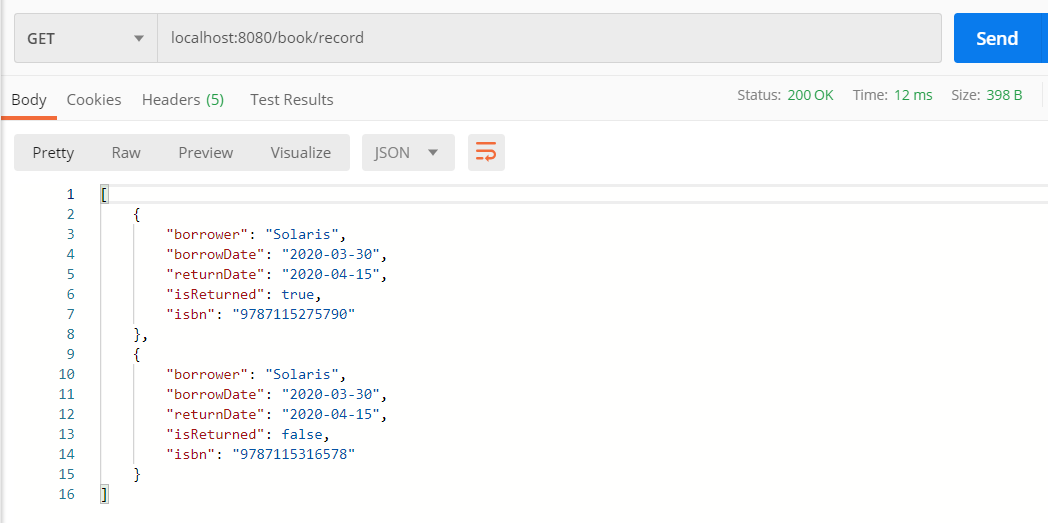
示例：



### 查询全部借阅记录

* 请求方法：GET
* 路径：localhost:8080/book/record
* 参数：无
* 返回结果：全部记录

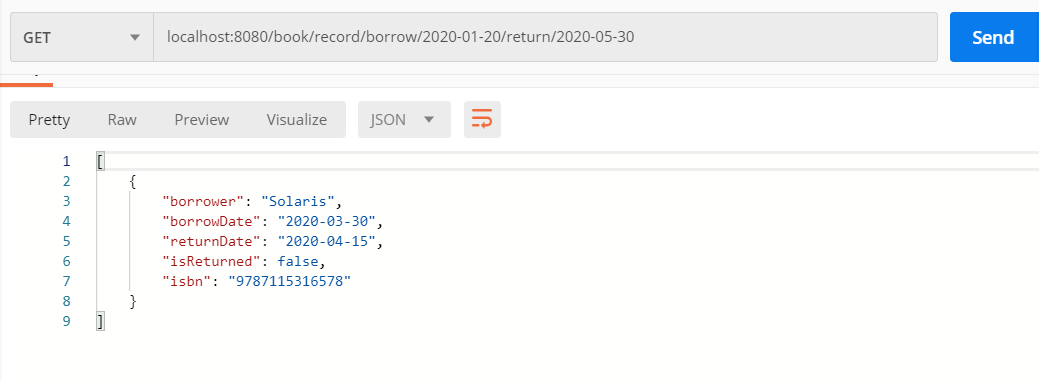
示例：



### 根据时间查询借书记录(YYYY-MM-DD)

* 请求方法：GET
* 路径：localhost:8080/book/record/borrow/{borrowDate}/return/{returnDate}
* 参数：在路径中添加borrowDate，returnDate
* 返回结果：返回全部记录里的日期区间在请求的日期区间里的结果

示例：



## 借书，还书接口

### 借书

* 请求方法： POST
* 路径： localhost:8080/book/{ISBN}/borrow
* 参数：在路径添加ISBN参数；在body中添加borrower， borrowDate, returnDate

（注意格式为“YYYY-MM-DD”）

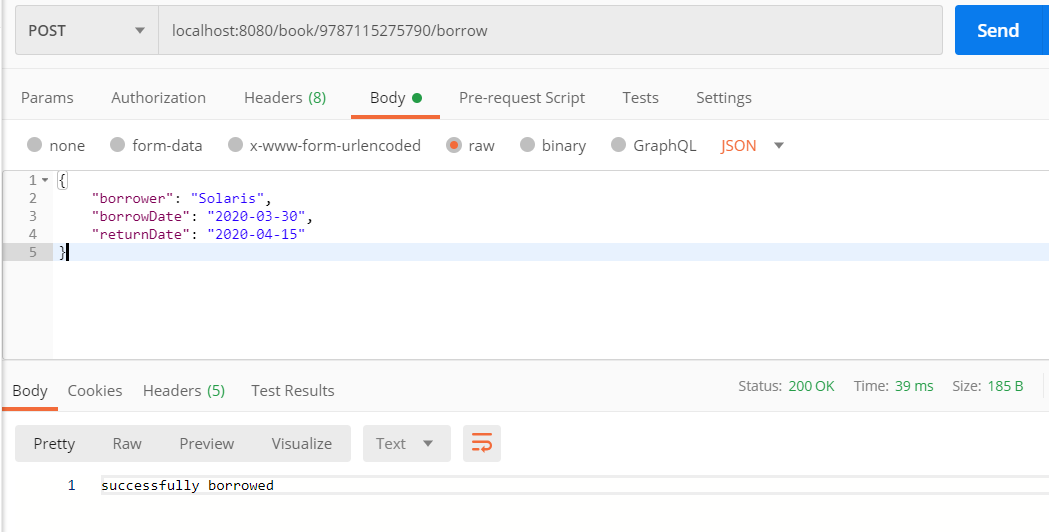
* 返回结果：

如果借阅成功，返回"successfully borrowed"；

如果书籍不存在，返回"book not exist"；

如果书籍已被借出，返回"book is already borrowed"

示例：



### 还书接口

* 请求方法：POST
* 路径： localhost:8080/book/{ISBN}/return/{username}
* 参数：在路径中添加ISBN， username
* 返回结果：

如果归还成功， 返回"successfully returned"；

如果书籍不存在，返回"book not exist"；

如果没有正确的记录，返回"wrong information"

示例：

