

示例

- GET /students: 获取所有学生
- POST /students: 新建学生
- GET /students/id: 获取某一个学生
- PUT /students/id: 更新某个学生的信息 (需要提供学生的全部信息)
- PATCH /students/id: 更新某个学生的信息 (需要提供学生变更部分信息)
- DELETE /students/id: 删除某个学生

过滤信息 (Filtering)

如果记录数量过多, 服务器不可能将它们返回给用户。API应该提供参数, 过滤返回结果。

- ?limit=10
- ?offset=10
- ?page=2&per_page=20
- ?sortby=name&order=desc
- ?student_id=id

参数的设计允许存在冗余, 即允许API路径和URL参数偶尔有重复, 比如 GET /students/id 和 ? student_id=id

状态码

服务器向用户返回的状态码和提示信息, 常见的有以下一些地方

- 200 OK - [GET]: 服务器成功返回用户请求的数据
- 201 CREATED -[POST/PUT/PATCH]: 用户新建或修改数据成功
- 202 Accepted - [*]: 表示一个请求已经进入后台排队 (异步任务)
- 204 NO CONTENT - [DELETE]: 表示数据删除成功
- 400 INVALID REQUEST - [POST/PUT/PATCH]: 用户发出的请求有错误
- 401 Unauthorized - [*]: 表示用户没有权限 (令牌, 用户名, 密码错误)
- 403 Forbidden - [*]: 表示用户得到授权, 但是访问是被禁止的
- 404 NOT FOUND - [*]: 用户发出的请求针对的是不存在的记录
- 406 Not Acceptable - [*]: 用户请求格式不可得
- 410 Gone - [GET]: 用户请求的资源被永久移除, 且不会再得到的
- 422 Unprocesable entity -[POST/PUT/PATCH]: 当创建一个对象时, 发生一个验证错误
- 500 INTERNAL SERVER ERROR - [*]: 服务器内部发生错误

错误处理

如果状态码是4xx, 就应该向用户返回出错信息。一般来说, 返回的信息中将error做为键名

返回结果

针对不同操作，服务器想用户返回的结果应该符合以下规范

- GET /collection: 返回资源对象的列表（数组，集合）
- GET /collection/id: 返回单个资源对象
- POST /collection: 返回新生成的资源对象
- PUT /collection/id: 返回完整的资源对象
- PATCH /collection/id: 返回完整的资源对象
- DELETE /collection/id: 返回一个空文档

使用链接关联资源

RESTful API最好做到Hypermedia，即返回结果中提供链接，连向其他API方法，使得用户不查文档，也知道下一步应该做什么。

```
{
  "link": {
    "rel": "collection https://www.rock.com/zoostudents",
    "href": "https://api.rock.com/students",
    "title": "List of students",
    "type": "application/vnd.yourformat+json"
  }
}
```

- rel: 表示这个API与当前网址的关系
- href: 表示API的路径
- title: 表示API的标题
- type: 表示返回的类型

其它

- 服务器返回的数据格式，应该尽量使用JSON
- API的身份认证应该使用OAuth2.0框架



```
data : BOOK.to_dict()
}
return JsonResponse(data=data, status=201)
```

把真正的 status 改掉



