HTTP协议常见的返回状态码：

200：服务器成功返回网页

404：请求的网页不存在

503：服务器超时

500：代码逻辑错误（也是服务器的原因）

502：服务器响应超时（服务器的服务没有启动起来）

504:504 Gateway Timeout 是一种HTTP协议的服务器端错误状态代码，表示扮演网关或者代理的服务器无法在规定的时间内获得想要的响应。Gateway （网关）在计算机网络体系中可以指代不同的设备，504 错误通常不是在客户端可以修复的，而是需要由途径的Web服务器或者代理服务器对其进行修复。

504

事件描述：dns查询过程超时，返回504；摸不着头脑，不管访问什么网站，都报504这个错误

问题原因：nginx或者后端配置不正确

解决办法：上网查nginx或后端的配置参数是否正确或者合理

解释： 实际上504很少会遇到，通常这个错误是由于nginx配置不当引起的，比如你将你的nginx的超时时间设置为300，那么如果此次请求的响应时间超过了300，你就会看到504这个报错。明白了吧。官方说法：**请求超时**

503

灾难事件：**临时的服务器维护/过载**，服务器当前无法处理请求，报503

问题原因：请求用户量太多，服务器为了保护自己不挂掉，机智的拒绝某些用户的访问，这些用户就会收到503这个错误

解决办法： 等一会儿仔访问该网站或者尝试强刷新页面，问题一般就能够解决了。

301：（永久移动）请求的网页已永久移动到新位置，服务器返回此响应时，会自动将请求者转到新位置。您应使用此代码告诉用户已永久移动到新位置。

302：（临时移动）服务器目前从不同位置的网页响应请求，但请求者应继续使用原有位置来响应以后的请求。此代码与响应请求的301代码类似，会自动将请求者转到不同的位置，但不应该告诉网页已经移动。

在ajax请求后台数据时有时会报 HTTP 400 错误 - 请求无效 (Bad request);出现这个请求无效报错说明请求没有进入到后台服务里;

原因：1）前端提交数据的字段名称或者是字段类型和后台的实体类不一致，导致无法封装;

2）前端提交的到后台的数据应该是json字符串类型，而前端没有将对象转化为字符串类型;

解决方案:

1）对照字段名称，类型保证一致性

2）使用stringify将前端传递的对象转化为字符串　data: JSON.stringify(param) ;

403是指没有权限访问这个服务

```
You, a few seconds ago | 1 author (You)
class SchedulePunchCardEmployeesResource(Resource):
    def post(self):
        employee_key = cas.username
        employee = Employee.get_by_key(employee_key)
        employee_detail = EmployeeDetail.get(employee.id)
        if not employee_detail:
            return dict(error='employee not complete'), 400

        employee_role_id = employee_detail.role_id
        has_permission = auth_helper.has_schedule_manager_permission(employee_role_id)
        if not has_permission:
            return dict(error='您没有足够的权限进行此项操作！'), 403        You, 2 minutes ago • Uncom

        form = request.get_json(True, True)
        if not form:
            logging.info("post schedule employees fail: form is null")
            return dict(error='form is null'), 400

        schedule_id = form.get('schedule_id')
        if not schedule_id:
            logging.info("get schedule rules fail: schedule id is null")
            return dict(error='schedule id is null'), 400

        employee_ids = list()
        employee_ids = form.get('employee_ids')
        if not employee_ids:
            logging.info("get employee_ids rules fail: schedule id is null")
            return dict(error='employee ids is null'), 400

        is_trial = form.get('is_trial')
        if is_trial is None:
            is_trial = int(False)
        else:
            is_trial = int(is_trial)

        memberIDs = list()
```

405：

比如原来需要的方法是post，但是你发起的是get请求，会报405，检查请求方法是否正确