

1、GITflow工作流

- 概念：企业在对使用git的分支管理和版本控制的方式。
- 流程：在win系统先把代码clone下来；
 - 1、查看分支：git branch
 - 2、查看本地分支和远程分支的映射关系：git branch -vv
 - 3、在本地master上创建本地dev，并让本地dev和远程dev映射：git checkout -b dev origin/dev
 - 每个同学，不能往dev分支推送代码，但是，在dev创建新分支，推送新分支代码；
 - 每个同学，先创建分支，再实现代码。
 - 把头条代码放到2个地方：
 - 一个地方每天只拉取老师的最新代码；
 - 一个地方只在今天拉取一次，后续不在拉取dev代码，可以创建新分支，每天不从dev拉取老师代码，可以推送到自己的分支上，自己练习代码用。
 - 4、对接好dev分支后，在dev分支上，新建功能分支，分支名建议f_login/f_orders，在新分支上实现代码，代码实现完成后，连分支和代码一起推送到远程仓库。

2、项目目录的理解

- 入口文件、启动文件，新的项目的快速理解建议直接找到入口，文件名main/app/manage
- 项目代码启动：
 - 1、代码上传到centOS，选择远程解释器
 - 2、右键运行main文件，编辑配置，以module-name方式运行flask run -h 0.0.0.0，配置FLASK_APP=toutiao.main，删除toutiao/
 - 3、在centOS系统heme/python/路径下，创建logs文件夹
 - 4、把common文件夹右键标记为source-root。
 - 5、启动成功
- 项目目录说明：
 - 1、**toutiao**文件夹：核心文件夹，main启动文件，__init__.py 核心初始化文件(工厂函数)，视图-蓝图
 - 2、**common**文件夹，项目的公用文件，cache、utils、models、settings等

3、调试方法

- 1、尝试分析错误信息，判断错误产生的大致位置，前后端；
- 2、状态码：4xx开头看前端，5xx开头看后端，前后端分离的情况下；
- 3、断点调试
- 4、查看日志：**tail**指令
 - tail flask.log看最后1条日志，

- `tail -f flask.log`查看实时日志,
- `tail -n 20 flask.log`查看多条日志

4、JWT认证

- JWT认证的本质：通过加密字符串，颁发给用户，用户请求时必须携带，区分用户身份。
- 加密字符串 = header + payload + signature
 - header：加密方式，编码，base64
 - payload：业务数据，过期时间；
 - signature：header.payload.secret_key签名字符串
- 编码？加密？签名？
 - 1、编码：可逆的，表示对计算机数据的表达方式，比如字符串编码(ascii/utf8/gbk)、音频、图片、文件，base64；
 - 2、加密：表示明文转成密文，可逆的，有密钥；sha1、sha256、sha512
 - 3、签名：不可逆的，表示对数据进行标记，md5
- 头条项目中token的生成使用pyjwt
- 需要安装：pip install pyjwt
 - 1、pyjwt.encode()表示生成token
 - 2、pyjwt.decode()表示校验token
- 需求：
 - token，设置有效期，但有效期不宜过长，需要刷新。
 - 如何解决刷新问题？
- token刷新的本质：
 - 1、生成2个token，token1、token2；
 - 2、token1用来请求业务接口，有效期2小时，每隔2小时，会更换一个新token1；
 - 3、token2用来请求更换token的凭证，有效期14天，只有在token1过期的时候需要携带；
- 登录代码说明：
 - 短信验证码校验？两种都行，校验手机号是你的。
 - 1、先删除、再比较；
 - 2、先比较、再删除；
 - 图片验证码校验？校验你是不是人？
 - 1、先删除、再比较：因为，只能get数据库中redis一次；

5、token刷新的实现

- 1、生成2个token；token1有效期2小时，token2有效期14天
 - 代码实现：toutiao/resources/user/passport.py
- 2、在请求钩子中，实现每次请求尝试获取用户token，校验token，提取用户id，赋值给g对象
 - 代码实现：在common/utis/middlewares.py

- 3、定义登录验证装饰器，从g对象中提取用户id，如果有用户id，允许进入视图，否则返回错误信息
 - 代码实现：在common/utils/decorators.py
- 4、定义刷新token接口，必须携带token2，校验token2，返回新的token1；
 - 代码实现：在toutiao/resources/user/passport.py

6、jwt禁用问题

- 需求：用户修改密码，封禁部分用户。