

Python 大作业

PB18020694 刘洪辰

1 任务说明

1. 学习利用 API 文档进行 API 相关的 python 编程.
2. 学习 GUI 相关的 python 编程
3. 对<https://gank.io/> API 编程和 GUI 展示

2 实验细节

2.1 API 相关的说明

2.1.1 首页 banner 轮播

<https://gank.io/api/v2/banners> 请求方式: GET 注: 返回首页 banner 轮播的数据

2.1.2 分类 API

https://gank.io/api/v2/categories/<category_type> 请求方式: GET
注: 获取所有分类具体子分类 [types] 数据

1. category_type 可接受参数 Article |GanHuo |Girl
2. Article: 专题分类、GanHuo: 干货分类、Girl: 图片

2.1.3 分类数据 API(子分类具体数据)

`https://gank.io/api/v2/data/category/<category>/type/<type>/page/<page>/count/<count>` 请求方式: GET

1. category 可接受参数 All(所有分类) | Article | GanHuo | Girl
2. type 可接受参数 All(全部类型) | Android | iOS | Flutter | Girl | app | frontend 等, 即分类 API 返回的类型数据
3. count: [10, 50]
4. page: ≥ 1

2.1.4 随机 API

`https://gank.io/api/v2/random/category/<category>/type/<type>/count/<count>` 请求方式: GET

1. category 可接受参数 Article | GanHuo | Girl
2. type 可接受参数 Android | iOS | Flutter | Girl, 即分类 API 返回的类型数据
3. count: [1, 50]

2.1.5 文章详情 API

`https://gank.io/api/v2/post/<post_id>` 请求方式: GET

1. post_id 可接受参数文章 id[分类数据 API 返回的 _id 字段]

2.1.6 本周最热 API

`https://gank.io/api/v2/hot/<hot_type>/category/<category>/count/<count>` 请求方式: GET

1. hot_type 可接受参数 views (浏览数) | likes (点赞数) | comments (评论数)

2. category 可接受参数 Article | GanHuo | Girl

3. count: [1, 20]

2.1.7 文章评论获取 API

`https://gank.io/api/v2/post/comments/<post_id>` 请求方式: GET

1. post_id 可接受参数文章 Id

2.1.8 搜索 API

`https://gank.io/api/v2/search/<search>/category/<category>/type/<type>/page/<page>/count/<count>` 请求方式: GET

1. search 可接受参数要搜索的内容

2. category 可接受参数 All[所有分类] | Article | GanHuo

3. type 可接受参数 Android | iOS | Flutter ..., 即分类 API 返回的类型数据

4. count: [10, 50]

5. page: >=1

2.2 设计的类

设计的类包含 *Ui_Dialog*, *MainDialog*, *NewWin*, 其中 *ClassMainDialog* 最为复杂, 所以仅介绍此部分. *MainDialog* 是 *QDialog* 的一个子类.

```
1 def __init__(self):
2     super().__init__()
3     self.ui = Ui_Dialog()
4     self.ui.setupUi(self)
```

这部分通过 *Ui_Dialog()* 定义了可视化的基本布局. 包含按钮位置, 文本框位置, 选择按钮位置.

```
1 def queryArticle(self):
2     Category = self.ui.comboBox1.currentText()
3     Type = self.ui.comboBox2.currentText()
4     CategoryCode = self.get_CategoryCode(Category)
5     TypeCode = self.get_TypeCode(Type)
6     r = requests.get("https://gank.io/api/v2/data/
        category/{}/type/{}/page/1/count/10".format(
            CategoryCode, TypeCode))
```

这一部分相当于调用接口,Category 和 Type 为用户选择的选项的反馈,并通过 get_CategoryCode,get_TypeCode 两个函数变为对应的 API 代码.其中 ge_CategoryCode 代码如下,get_TypeCode 代码与之类似遂省略.

```
1 def get_CategoryCode(self, Category):
2     CategoryDict = {"Article": "Article",
3                     "Skill": "GanHuo",
4                     "Girl": "Girl"}
5     return CategoryDict.get(Category)
```

```
1 def searchText(self):
2     item = self.ui.textEdit.text()
3     r = requests.get("https://gank.io/api/v2/search
        /{}/category/All/type/All/page/1/count/10".
        format(item))
4     ...
5     self.ui.textEdit2.setText(Msg2)
```

这一部分为利用查找 API 进行查找,输入通过 textEit.text 进行读取后进行查找.进行反馈后,通过 setttext 进行展现.

```
1 def clearText(self):
2     self.ui.textEdit.clear()
```

这一部分为点击按钮后对搜索框的内容进行清除.

```
1 def detialText1(self):
2     global idd
3     idd = self.r.json()['data'][0]['_id']
4     self.child_win = NewWinGirl()
5     self.child_win.show()
6     self.child_win.exec_()
```

这里遇到的难点为如何将这个窗口内 request 的结果传递给下一个窗口, 我采用了设置全局变量的方法, 传递文章对应的 id. 此外如何创建新的子窗口也为本次作业中卡的时间较长的部分. 在 class NewWin(QDialog) 中:

```
1 self.lb = QLabel(self)
2 self.lb.setText("<A href='{ }'>The Url</a>".format(self
    .r.json()['data']['url']))
3 self.lb.setOpenExternalLinks(True)
```

这里为利用 QLabel 创建超链接, 直接打开网页.

此外, 编写代码的过程中通过 API 获取数据后, 对 json 文件的操作, 可以堪称字典和列表的套娃, 进行操作起来比较方便.

3 界面布局

界面布局采用了 QGridLayout 和 setGeometry(QtCore.QRect()) 相结合的方法. 此外, 利用 QGroupBox 进行分块有利于界面的布局的美观. 主窗口和子窗口具体布局可以参见视频

4 实验总结

本次编程的结果虽有些地方想要改进但未找到具体可行的办法, 但总体来说较为满意, 运行结果符合预期. 本次实验最大的收获是了解了 API 和 GUI, 通过自己阅读 API 文档进行编程, 到设计 GUI 进行成果展现也提高了自己的编程能力.