
百度爬虫项目技术文档

1. 软件功能介绍:

项目描述: 该爬虫项目旨在从百度图片中爬取指定关键词的图片, 并将其保存到本地文件夹中。该软件提供了一个图形用户界面 (GUI) 来接收用户的输入并展示爬取的结果并且在程序执行结束后提示用户图片存储在本地的位置, 方便用户对下载图片的管理, 提升用户使用体验

2. 爬虫模块主要使用的技术

1. 在爬虫模块中, 我们主要使用了以下技术:
2. *Request* 模块: 使用 *Request* 模块向目标网站发起请求, 并将获取到的数据使用 *JSON* 转换为字符串。我们从数据中提取出包含下载链接和图片名字的信息。
3. *UserAgent* 模块: 使用 *UserAgent* 模块设置请求头中的 *User-Agent* 信息。通过随机生成代理请求, 我们可以有效地应对网站的反爬机制。
4. *Value* 模块: 在程序中设定了一个打印的提示信息, 最开始在选用 *Value* 方法之前是设定了一个全局的计数器, 并且使用 *global* 在函数内声明全局计数。然而, 我们最终选择使用 *Value* 模块, 主要考虑到程序运行时的安全性。

因为 `Value` 可以在多个线程之间同时访问和修改, 而不会导致数据的竞争和其他问题。在函数内使用 `Value` 之前需要使用 `with` 语句获取我们设定的 `Value`, 以保证计数器的线程安全, 在对 `Value` 及计数器进行操作。

5. `Os` 模块: 使用 `Os` 模块来创建用户在下载图片完成后创建的指定文件名, 并且加入判断, 如果用户创建的文件不存在则使用 `os.makedirs` 创建保存图片的文件, 方便用户对下载的文件进行统一的管理
 6. 。
-

3.Pyqt5 主要使用得技术

1. `QtDesigner`: 使用 `QtDesigner` 进行 `GUI` 界面的设计。我们可以通过拖拽式的设计快速地创建一个完美的界面。在 `GUI` 界面中, 我们设定了用户的输入操作和超链接的添加, 并且设定了背景图片。最后, 我们将下载的结果显示在 `GUI` 界面中并提示用户文件保存的位置。
 2. 界面背景设定: 使用 `QTextEdit` 组件设定界面的背景, 并且在 `QTextEdit` 中添加了背景图片。
 3. 鼠标悬停闪现消息方法: 使用 `QTextEdit` 中的 `setToolTip()` 方法设定鼠标悬停后显示此文本框使用方法的提示信息
 4. 。
-

4.项目过程遇到难点

第一个难点是：在程序制作过程中，GUI 界面运行程序后出现卡死。为了应对这个卡死的 bug，在程序中开辟了一条新线程用于爬虫程序的运行。主要使用 PyQt 的信号与槽机制，将爬虫程序运行结果发送到主线程，避免 GUI 界面的卡死出现。

第二个难点是：用户在 GUI 界面输入的数据传入到新的线程中，并且启动爬虫程序。最开始的方法是在槽函数内直接与 GUI 界面的文本框对接，并且在槽函数内创建新线程的实例，将参数传入。但是这个方法并没有解决 GUI 界面卡死的问题。在查找方法后，我对爬虫模块进行重新设定，在爬虫模块的对象中加入形参并设定值为空，在创建对应的实参时，就可以将用户输入的数据传入到新线程中，并且运行爬虫程序，最终解决了 GUI 界面卡死的问题。

。

5.总结

本项目实现了一个简单的百度图片爬虫程序，采用了 Python 语言和相关的第三方库来实现爬虫模块，使用 PyQt5 实现了用户交互界面。在项目开发的过程中，我遇到了不少问题，但通过学习相关资料和持续不断的尝试和调试，最终解决了这些问题，并完成了项目。

然而，本项目还有许多可以改进和扩展的地方。例如，可以增加更多的爬虫功能，如按照关键词爬取图片，或者在下载过程中实现断点续传等。另外，还可以考虑

使用更高级的技术来提高程序的性能和稳定性，如使用多线程、多进程等并行处理方式来加速下载和提高程序的响应速度。

总的来说，本项目是一个不错的练手项目，可以帮助初学者深入了解 Python 语言、爬虫技术和 GUI 界面的开发，同时也可以提高编程能力和解决问题的能力。在未来的学习和工作中，我将继续深入研究和应用相关技术，不断提高自己的能力和水平。

