<https://www.youtube.com/watch?v=dD1dDf__RK0&list=PLCLjnatUzPGXrmrv1lXN--PcIVM9afOQE&ab_channel=SEASResearchLab>

GithHub имеет свой собственный API версия 3

<https://github.com/orgs/PyGithub/repositories> - Typed interactions with the GitHub API v3

доступен для всех разработчиков.

Где доступна Информация для участников, для интеграторов, информация обо всем.

Но изначально, когда Вы начинаете работать в этой области, использование этого API немного сложно, поэтому я предлагаю первоначально использовать библиотеку, которая разработана на основе guthub rest api версии <https://pygithub.readthedocs.io/en/latest/introduction.html>

Теперь, это официальный веб-сайт библиотеки pi github, поэтому я советую ознакомиться и прочитать эту документацию. В этом руководстве мы практически начнем реализацию, используя библиотеку pi github.

Начнем работу с github и извлечем данные из github, переходим в pycharm,

проверим есть ли возможность для создания виртуальной среды и установки в нее различных библиотек Python – anaconda?

Anaconda – используется для создания в ней виртуальной среды

PS F:\HTML\Python\students\preply\olga\_st\pyGithub\_api> pip install PyGithub

После установки импортируем ее

from github import Github

# First create a Github instance:

Создадим переменную

# using an access token

access\_token = ""

Токен доступа – позволяет мне создать переменную с именем, и как только вы захотите взаимодействовать с github, Вам нужно будет создать для этого класс, для которого будет метод по аутенсификации на github, также для отправки запроса Вам нужна аутенфицикация

Есть три типа аутенфикации (базовая – передача вашего имени пользователя и пароля)

g = Github("username", "password")

Вы можете напрямую вставить username и password, но это не считается безопасным, т.к. как никто не хочет делиться своим паролем. Другой метод сохранить Ваше имя пользователя в системной среде и получить к ним доступ, но все же тоже не рекомендуется. Второй метод – это художественная аутентификация, в которой используется стороннее ПО, третий тип – использование токена личного доступа. Сейчас мі будем использовать токен личного доступа.

Что такое токен личного доступа и как его создать?

Для этого у Вас должен быть аккаунт в репозитории github, при регистрации Вы прописываете Ваш username, password, e-mail, у Вас появляется собственная учетная запись

Переходим в Github🡪Settings🡪Developer settings🡪Personal access tokens

В этой среде я могу создать свой собственный токе личного доступа

Вводим пароль \*\*\*\*\*\*\*\*

Предоставленный вами пароль находится в списке паролей, обычно используемых на других веб-сайтах. Для повышения безопасности вам необходимо обновить свой пароль. После 21 апреля 2022 г. мы автоматически сбросим ваш пароль. Смените пароль на странице настроек.

**Note: yurii\_token**

При предоставлении следующей области **Select scopes** будьте осторожны, т.к. это может быть опасно, а именно когда Вы проверяете это право и удаляете пакет, например человек у которого есть этот токен доступа может выполнять такие операции в Вашем репозитории, поэтому на основе Ваших требований проверьте эти области.

Выбираем:



Нажимаем Generate token

И сейчас копируем текст, т.к. Вы только один раз будете видеть свой токен, сохраните его и вставьте в файл yurii\_token.txt. Далее вставляем token в access\_token. Мы использовали личный токен доступа аутенфикации.

g = Github(access\_token)

Проверим, что мы создали объект класса, который передает Ваш токен доступа

print(type(g))

<class 'github.MainClass.Github'>

Например мне нужно получить доступ к текущему пользователю этой учетной записи, которая является, потому что она моя, поэтому сейчас я есть текущий пользователь.

current\_user = g.get\_user()

get\_user() – означает получить мою информацию

Распечатаем имя текущего пользователя, вся информация связанная с текущим пользователем будет доступна здесь сейчас и мой биодид, Вы также можете изучить остальную информацию, запускаем

print(current\_user.name)  
print(current\_user.bio)  
print(current\_user.company)  
print(current\_user.location)

Эту информацию Вы можете найти в github - Your profile.

Например я хочю звлечь свои репозитории, потомучто я текущий пользователь, извлечем эти репозитории. Напечатаю имя этих репозиториев, но позже мы можем извлечь другую инфформацию.

repos = g.get\_user().get\_repos()  
  
for repo in repos:  
 print(repo.name)

Теперь у нас есть доступ к дереву репозиториев

<https://github.com/>

Сейчас мы получили доступ, только к извлеченным именам этих репозиториев, но позже мы рассмотрим как получить остальные данные, принадлежащие этим репозиториям.

Например нам нужно получить проекты github репозитория, которые написаны на каком-то языке например java, python.

Вы можете искать и извлекать эти проекты, информацию из этих проектов

java\_repos = g.search\_repositories(query="language:java")  
for repo in java\_repos:  
 print(repo.name)

<https://docs.github.com/en/search-github>

<https://docs.github.com/en/search-github/searching-on-github>

<https://docs.github.com/en/search-github/searching-on-github/searching-code>

<https://docs.github.com/en/search-github/getting-started-with-searching-on-github/sorting-search-results>

<https://docs.github.com/en/search-github/getting-started-with-searching-on-github/about-searching-on-github>