Intermediate APIs: Takeaways

by Dataquest Labs, Inc. - All rights reserved © 2019

Syntax

- Passing in a token to authorize API access:

```
{"Authorization": "token 1f36137fbbe1602f779300dad26e4c1b7fbab631"}
```

- Defining pagination query parameters:

```
{"per_page": 50, "page": 1}
```

- Using a POST request to create a repository:

```
payload = {"name": "test"}

requests.post("https://api.github.com/user/repos", json=payload)
```

- Using a PATCH request to change a repository:

```
payload = {"description": "The best repository ever!", "name": "test"}

response = requests.patch("https://api.github.com/repos/VikParuchuri/test", json=payload)
```

- Using a DELETE request to delete a repository:

```
response = requests.delete("https://api.github.com/repos/VikParuchuri/learning-about-apis"
```

Concepts

- APIs use authentication to perform rate limiting. Rate limiting ensures the user can't overload the API server by making too many requests too fast, which allows the API server to be available and responsive for all users.
- Rate limiting ensures the user can't overload the API server by making too many requests too fast.
- APIs requiring authentication use an access token. An access token is a string the API can read and associate with your account. Tokens are preferable to a username and password for the following security reasons:
    - Typically someone accesses an API from a script. If you put your username and password in a script and someone manages to get their hands on it, they can take over your account. If someone manages to get their hands on the access token, you can revoke the access token.
    - Access tokens can also have scopes in specific permissions. You can generate multiple tokens that give different permissions to give you more control over security.

- Pagination allows for a certain number of records per page.
- Different API endpoints choose what types of requests they will accept.
    - We use POST requests to send information and to create objects on the API's server. POST requests almost always includes data so the server can create the new object. A successful POST request will return a 201 status code.
    - A successful PATCH request will return a 200 status code.
    - A PUT request will send the object we're revising as a replacement for the server's existing version.
    - A DELETE request removes objects from the server. A successful DELETE request will return a 204 status code.

Resources

- Github API documentation
- Understanding REST

Takeaways by Dataquest Labs, Inc. - All rights reserved © 2019