Angular week 2 IP

Due 7 Feb by 18:00 **Points** 22 **Submitting** a website url
Available 13 Jan at 0:00 - 7 Feb at 18:00 26 days

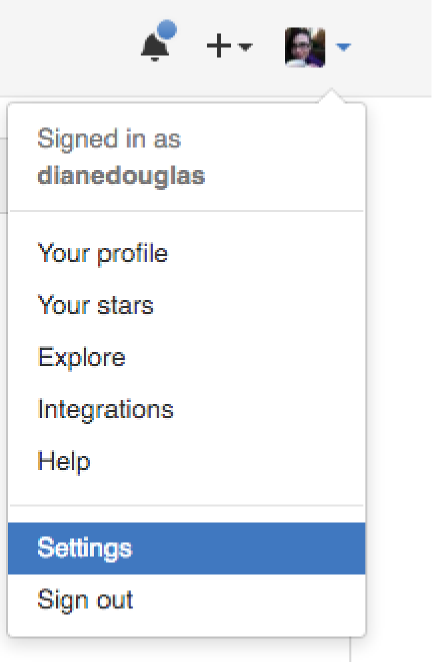
This assignment was locked 7 Feb at 18:00.

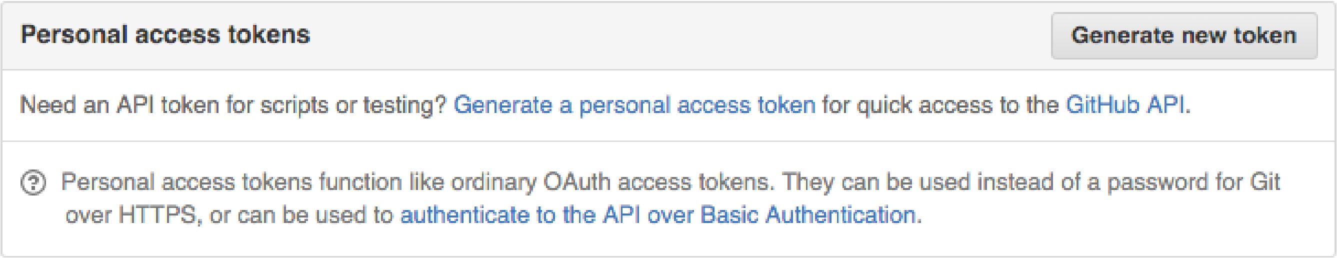
Github Search

Create a website where users may enter a GitHub username into a form, submit it, and see names and descriptions of that person's public repositories. A person can also look for repositories

Use the **GitHub API** (https://developer.github.com/v3/) to retrieve this information. This API allows 5,000 requests per hour with an API key, but only 60 requests per hour _without_ one. Everyone is therefore required to use their own unique key. GitHub refers to these keys as "Personal Access Tokens".

Creating Personal Access Tokens

- Visit the _Settings_ area of your GitHub account like this
- 

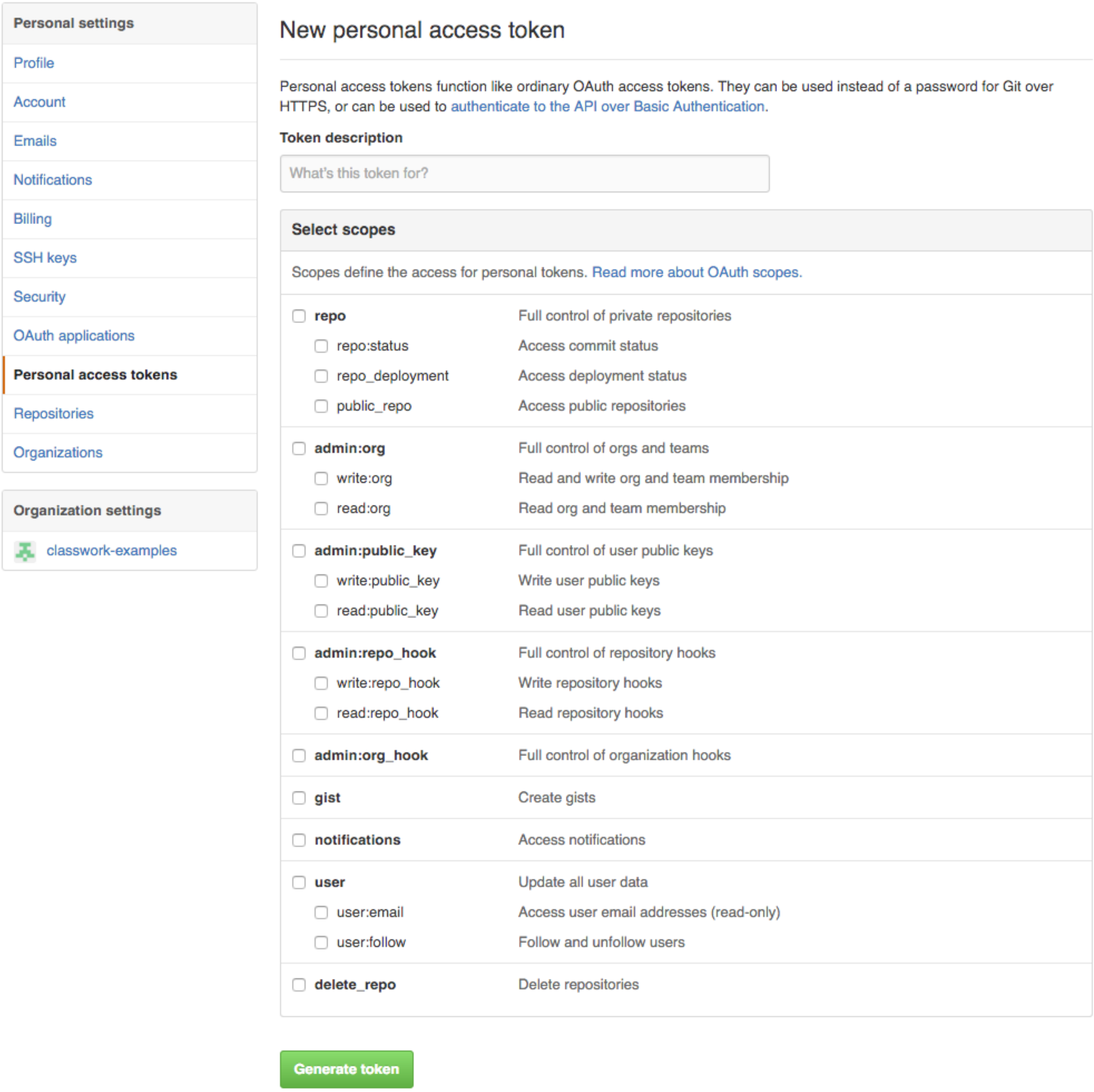
- select Personal Access Tokens from the sidebar, and hit Generate New Token
- 

Personal access tokens

Need an API token for scripts or testing? Generate a personal access token for quick access to the GitHub API.

Personal access tokens function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to authenticate to the API over Basic Authentication.

- GitHub will offer a list of options **Do not select any**. These grants read/write permissions and access to personal data. Finally, select **Generate Token.**

- 

Personal settings

Profile

Account

Emails

Notifications

Billing

SSH keys

Security

OAuth applications

Personal access tokens

Repositories

Organizations

Organization settings

classwork-examples

New personal access token

Personal access tokens function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to authenticate to the API over Basic Authentication.

Token description

What's this token for?

Select scopes

Scopes define the access for personal tokens. Read more about OAuth scopes.

□ **repo**	Full control of private repositories
□ repo:status	Access commit status
□ repo_deployment	Access deployment status
□ public_repo	Access public repositories
□ **admin:org**	Full control of orgs and teams
□ write:org	Read and write org and team membership
□ read:org	Read org and team membership
□ **admin:public_key**	Full control of user public keys
□ write:public_key	Write user public keys
□ read:public_key	Read user public keys
□ **admin:repo_hook**	Full control of repository hooks
□ write:repo_hook	Write repository hooks
□ read:repo_hook	Read repository hooks
□ **admin:org_hook**	Full control of organization hooks
□ **gist**	Create gists
□ **notifications**	Access notifications
□ **user**	Update all user data
□ user:email	Access user email addresses (read-only)
□ user:follow	Follow and unfollow users
□ **delete_repo**	Delete repositories

Generate token

A sample request URL might look like this.

```
'https://api.github.com/users/daneden?access_token=' + apiKey
```

Technical Requirements

1. Your project must contain a well-designed landing page that displays your GitHub information such as your username, your profile photos and a list of your repositories.
2. You must place your access key inside the `environment.ts` file and place it inside the `gitignore` file.
3. You must create a HTTP service that uses a promise to make the requests to the GitHub API.
4. The HTTP request should be able to search for both GitHub users and GitHub repositories.
5. Your project must have two classes for the `user` and `repository`.
6. Your project must have a proper routing structure that links a GitHub username to the users GitHub repositories.
7. Your project must have a separate routing module.

5.Your project must contain a custom directive and a custom pipe.

8. Your Project must be well designed and visually appealing and of portfolio quality.
9. Your project must have a well-documented README file.
10. Your project must be deployed and the deployed link should be submitted.

Github Project Objectives

Criteria	Ratings		Pts
A well documented readme is the first documentation any developer should use	**1 Pts** **A well documented Readme file on Github** A project README that includes: - project or program name - author name - description of project - project setup instructions - link to live site on GitHub Pages - copyright and license information	**0 Pts** **This is the default readme that is generated by Github on new projects. Usually contains the name of the project and a description if one is provided**	1 pts
Project is in a polished, portfolio-quality state. Suggestions for what this can mean: Intuitive, easy to follow layout. Simple yet polished styling. Form field labels where appropriate. Form fields that are validated correctly, and get cleared after submitting. Detailed, well put together readme. No typos. And much more.	**2 Pts** **Went above and beyond in the visual aspect** Suggestions for what this can mean: Intuitive, easy to follow layout. Simple yet polished styling. Form field labels where appropriate. Form fields that are validated correctly, and get cleared after submitting. Detailed, well put together readme. No typos. And much more.	**0 Pts** **No effort to make it visually pleasant**	2 pts
Does the project work functionally achieving the expected objective?	**2 Pts** **Yes, the project works as expected** All objectives are functionally met in the deployed project	**0 Pts** **No, the project does not work as specified in the objectives**	2 pts
Commits are made regularly with clear messages associated with them	**2 Pts** **20 + commits in the project with well detailed commit messages**	**0 Pts** **Less than 10 Github commits present in the project**	2 pts
Project is link and description is provided on Github repository	**2 Pts** **Project description is present** This is used as a blurb of the project, explains what the project is bout and the linked to a deployed site if present	**0 Pts** **Did not make use of Github gh-pages**	2 pts
Page incorporates a custom-made stylesheet	**1 Pts** **Used custom CSS stylesheet** All objectives are functionally met in the deployed project	**0 Pts** **Used custom CSS stylesheet**	1 pts
Project contains a well-designed landing page.	**1 Pts** **Well designed landing page**	**0 Pts** **Default HTML**	1 pts
HTTP service that uses a promise to make the requests to the Github API	**3 Pts** **Use working HTTP service**	**0 Pts** **No service available**	3 pts

Criteria	Ratings		Pts
Two classes for the User and Repository	**2 Pts** **Has two separate classes that are in separate file**	**0 Pts** **Classes are all in the same file**	2 pts
Your project must have a separate routing module	**3 Pts** **Project has a routing module linked to the app module**	**0 Pts** **Routing is not done inside the app module**	3 pts
Contains a custom directive and a custom pipe	**3 Pts** **Custom pipe and directive**	**0 Pts** **No custom pipe or directive**	3 pts
			Total points: 22