CSC 123 GitHub Classroom User Manual

Nathan Kennedy

Computer Engineering Undergraduate

Cal Poly San Luis Obispo

June 2024

Senior Project

Student: Nathan Kennedy

Computer Engineering Undergraduate

Advisor: Ayaan Kazerouni

Client: Ayaan Kazerouni

Cal Poly San Luis Obispo

June 2024

Table of Contents

Overview

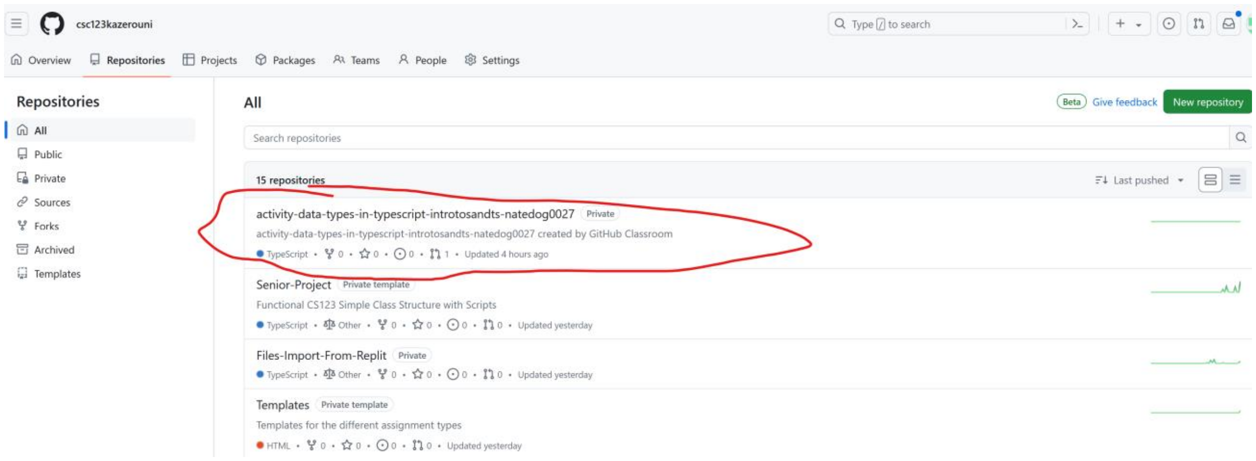
CSC 123 was originally taught using a free online development environment called Replit. This tool provided a simplistic and universally accessible location to learn to program in HTML, JavaScript, and TypeScript without the overhead of understanding a complicated programming environment. It was located entirely in the browser for accessibility to anyone with a computer. The students simply had to write their intended program and then click the run button, which would handle the compilation of their code and run it on a server so they can view the outcome of their work. However, the educational section of Replit, on which this classroom was run, is being phased out. Therefore, a new solution had to be found or created to maintain simplicity, ease of access, and functionality of this service in order to continue to provide introductory programming teachings to students with limited practice in programming.

The solution that was found involved GitHub Codespaces and GitHub Classroom. Codespaces provided an accessible and free online programming environment and GitHub Classroom provided the structure for which an instructor could assign projects to students, take submissions, and give feedback. To make this setup functional, Codespaces had to be configured to be as simplistic as possible while still being able to run all of the assignments that were previously on Replit. Specific scripts and configurations were written and set up to run the assignments that were transferred over from Replit. The finished project is a repository of functional templates and assignments, as well as a GitHub Classroom of student assignments to be used in an instructional manner. In this manual, I will elaborate on how to access, use, and navigate the CSC 123 GitHub setup for instructors and TA's.

Items and Resources

This section outlines the project structure and the location of various useful resources.

- *GitHub Organization:* This is the organization that holds all of the resources created for this project. This organization is called "csc123kazerouni".
  - Repositories: In the repository section, the various repositories for the different parts of this project are kept. Currently, there are 14 repositories.
  - Note: When actually running the future classroom, I suggest creating a new Organization for each class. When assignments are created and students start accepting and running their own Codespaces, the offshoot repositories that they make are added to the Organization that is associated with the Classroom. This can convolute the Organization with many new repositories that will make it difficult to navigate through the original resources and create new assignments. If you make a new Organization and Classroom setup for each class, then those repositories will be specific to each class, and you can hold and access the original resources yourself in a different area. Here is an image showing that GitHub adds student assignment repositories to the repository section of the organization:



15 repositories

activity-data-types-in-typescript-introtosandts-natedog0027 Private
activity-data-types-in-typescript-introtosandts-natedog0027 created by GitHub Classroom
TypeScript • 0 • 0 • 0 • 1 • Updated 4 hours ago

Senior-Project Private template
Functional CS123 Simple Class Structure with Scripts
TypeScript • Other • 0 • 0 • 0 • 0 • Updated yesterday

Files-Import-From-Replit Private
TypeScript • Other • 0 • 0 • 0 • 0 • Updated yesterday

Templates Private template
Templates for the different assignment types
HTML • 0 • 0 • 0 • 0 • Updated yesterday

- *Repositories:* These repositories are all held in the GitHub Organization mentioned above. They are currently private and only accessible to Nathan Kennedy and Ayaan Kazerouni. There are 14 in total:
  - File-Import-From-Replit: This repo holds the original Replit files that were downloaded and transferred from Replit. They are organized but unedited. This is a backup of the data from Replit if needed but can be deleted if not useful anymore.
  - Senior-Project: This repo holds all of the same assignments as the File-Import-From-Replit repository holds, but they are configured and functional.

When creating new assignments, I would clone this repository, remove the unnecessary files, and then edit the files that I needed for the specific assignments.

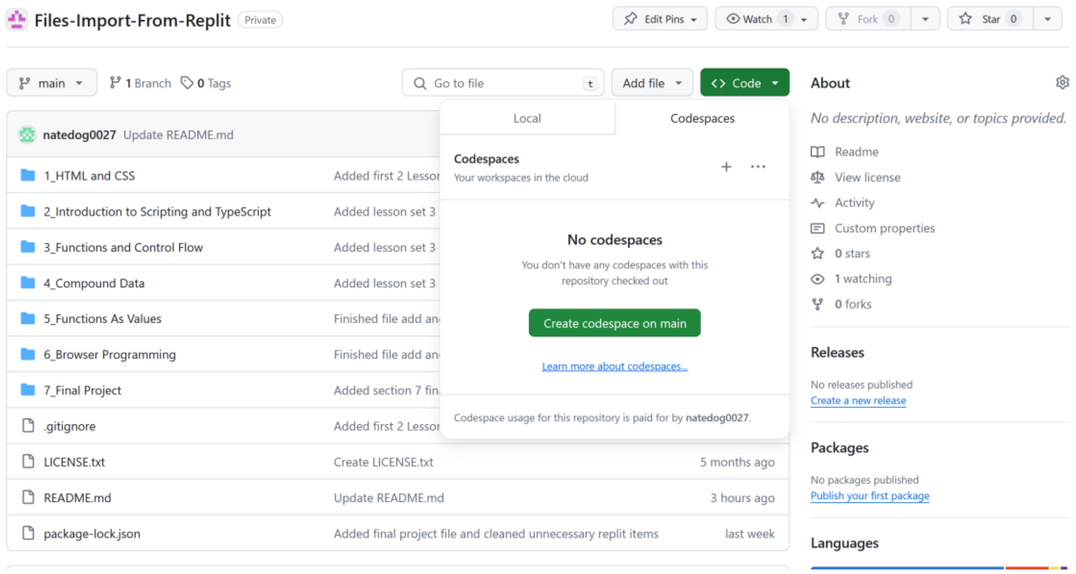
  - Templates: This repo holds 4 templates for the different assignment structures used for this project. The explanation of each template is in the associated readme.
  - The remaining repositories are the assignments that were configured for the classroom. Each of them is functional and usable. They have been uploaded to the classroom as assignments and are able to be assigned to students. There are 11 total:
    - Lab2Part1-HTMLandCSS
    - Lab2Part2-HTMLandCSS
    - Activity-DataTypesinTypescript-IntrotoScriptingandTypescript
    - Lab 3-IntrotoScriptingandTypescript
    - Activity-FunctionsandControlFlow
    - Activity-Loop Patterns- FunctionsandControlFlow
    - Lab4- FunctionsandControlFlow
    - Lab 5-CompoundData
    - Lab 6-BrowserProgramming
    - Practice-EventDrivenProgramming-BrowserProgramming
    - FinalGroupProject

- *ReadMe's:* There is extensive documentation in each repository for their specific uses that expands on the data in this manual.
  - File-Import-From-Replit: The readme simply goes over what this repo is used for.
  - Senior-Project: The main README.md explains what this repository is used for and includes all of the information provided for each of its sub-readmes. If you want all of the readme information that is reused for each assignment in one place, then read this file. The sub-readmes are also in this repo and are specific to each of the assignment template types.
  - Templates: The main README.md is an overview of what this repository is used for. The OverallREADME.md file is a copy of the readme from the "Senior-Project" repo. Within each template file, there are the specific assignment template readmes with their useful information.
  - The remaining repositories are the assignments that were configured for the classroom. Each of them has a readme file that is specific to the assignment and its use. While most of the information is the same as previously provided, there is occasionally different information included, such as the way each of the specific npm script buttons work.

- *GitHub Classroom:* This is the classroom where the functional assignments are located. All of the 11 assignment repositories previously spoken about have been set up in this classroom.

The Tools

Codespaces: Codespaces is used to run VSCode in the browser in order for students to access these assignments. To start a Codespace from a repository, click the green code button in the top right, select Codespaces, and click “Create Codespace on Main”. Wait for the setup to occur, and then the Codespace will be already configured and ready to run the programs.


Files-Import-From-Replit Private

Edit Pins · Watch 1 · Fork 0 · Star 0

main · 1 Branch · 0 Tags · Go to file · Add file · Code

natedog0027 Update README.md

1_HTML and CSS	Added first 2 Lessor	
2_Introduction to Scripting and TypeScript	Added lesson set 3	
3_Functions and Control Flow	Added lesson set 3	
4_Compound Data	Added lesson set 3	
5_Functions As Values	Finished file add an	
6_Browser Programming	Finished file add an	
7_Final Project	Added section 7 fin	
.gitignore	Added first 2 Lessor	
LICENSE.txt	Create LICENSE.txt	5 months ago
README.md	Update README.md	3 hours ago
package-lock.json	Added final project file and cleaned unnecessary replit items	last week

Local · Codespaces

Codespaces
Your workspaces in the cloud

No codespaces
You don't have any codespaces with this repository checked out

Create codespace on main

Learn more about codespaces...

Codespace usage for this repository is paid for by natedog0027.

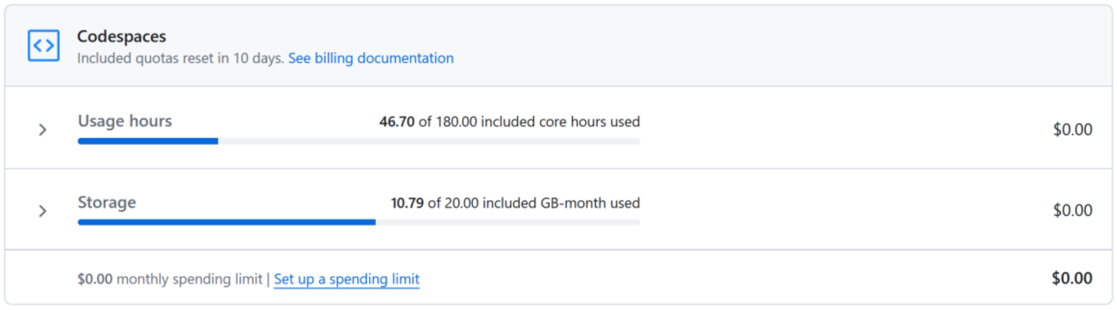
About
No description, website, or topics provided.
Readme
View license
Activity
Custom properties
0 stars
1 watching
0 forks

Releases
No releases published
Create a new release

Packages
No packages published
Publish your first package

Languages

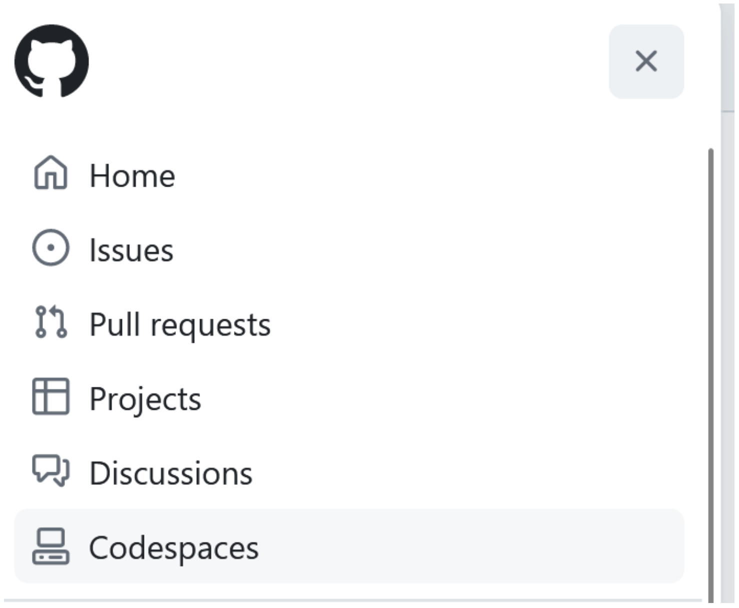
Codespaces have a limited amount of hours that can be used before you have to pay for usage. As of now, the limit is 60 hours. To view your usage, go to GitHub setting, Billing and Plans, and then scroll down for your usage data. It will look like this:


Codespaces
Included quotas reset in 10 days. See billing documentation

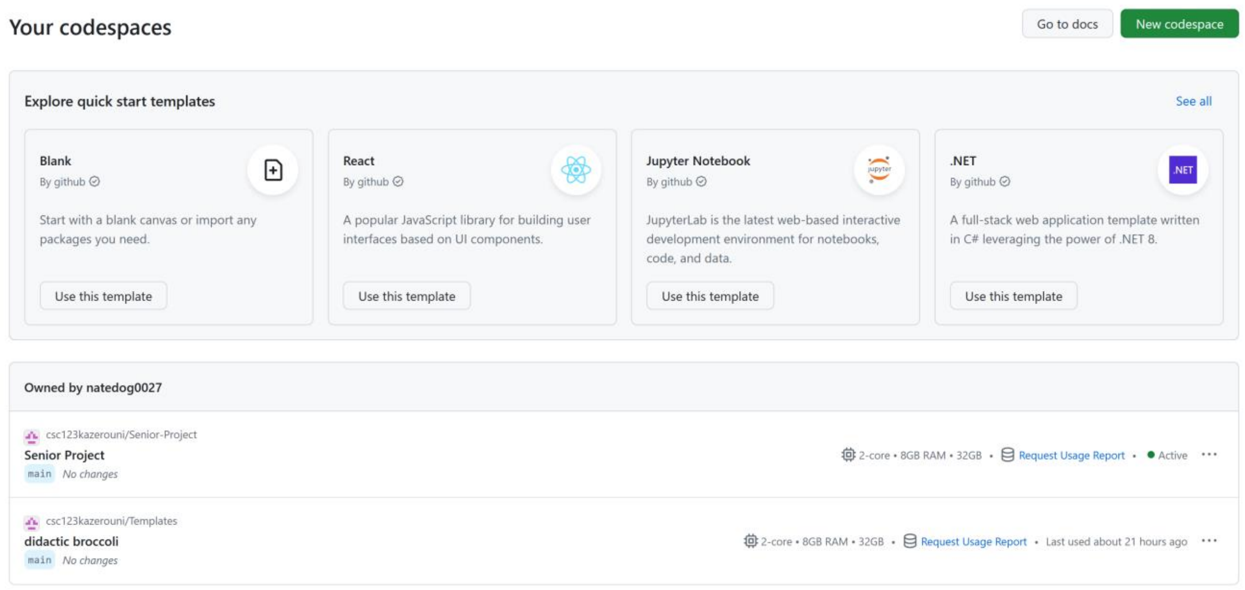
Usage hours	46.70 of 180.00 included core hours used	$0.00
Storage	10.79 of 20.00 included GB-month used	$0.00
$0.00 monthly spending limit | Set up a spending limit		$0.00

On the bright side, students and instructors can sign up for GitHub pro, which will allow them to increase their Codespace hours to 180 total per month. Ensure that students, instructors, and TA’s go to this link and set up their educational benefits: https://education.github.com/discount_requests/application. Note: Each hour is considered a core hour. Ensure that students are creating 2-core Codespace instances. This will allot up to 90 (180/2) hours of usage per month, which should be ideal.

To view, open, and delete active Codespaces, go to the menu button in the top left of GitHub. Then, select Codespaces. The menu looks like this:



It will open a webpage thats showcase active and running Codespaces, like this:



Your codespaces

Go to docs | New codespace

Explore quick start templates — See all

Blank
By github
Start with a blank canvas or import any packages you need.
Use this template

React
By github
A popular JavaScript library for building user interfaces based on UI components.
Use this template

Jupyter Notebook
By github
JupyterLab is the latest web-based interactive development environment for notebooks, code, and data.
Use this template

.NET
By github
A full-stack web application template written in C# leveraging the power of .NET 8.
Use this template

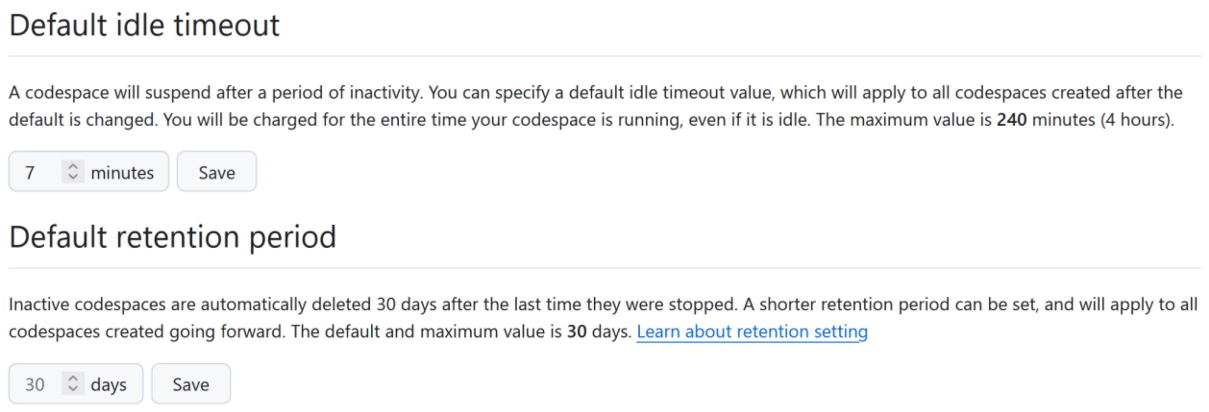
Owned by natedog0027

csc123kazerouni/Senior-Project
Senior Project
main *No changes*
2-core • 8GB RAM • 32GB • Request Usage Report • Active

csc123kazerouni/Templates
didactic broccoli
main *No changes*
2-core • 8GB RAM • 32GB • Request Usage Report • Last used about 21 hours ago

In this menu, you can close running Codespaces and delete unnecessary Codespaces to save resources.

Note: Codespaces is set to automatically close after 30 minutes and delete after 30 days. It is ideal to set the default timeout to a short amount of time so that when you are inactive, the Codespace will stop running and you can save core hours. The default retention period maximum is 30 days. Be sure to set it to the maximum. If you do not want the Codespace to close after 30 days, simply run it and the timer will reset. To access these settings, go to GitHub settings, Codespaces, and then scroll down and set these items:


Default idle timeout

A codespace will suspend after a period of inactivity. You can specify a default idle timeout value, which will apply to all codespaces created after the default is changed. You will be charged for the entire time your codespace is running, even if it is idle. The maximum value is **240** minutes (4 hours).

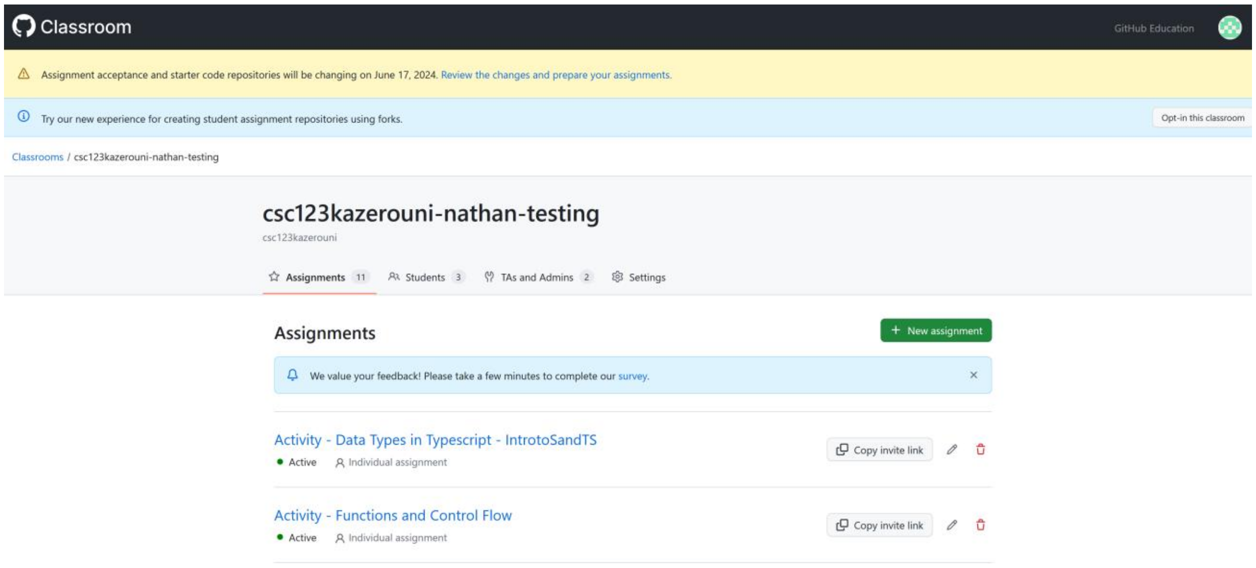
7 minutes Save

Default retention period

Inactive codespaces are automatically deleted 30 days after the last time they were stopped. A shorter retention period can be set, and will apply to all codespaces created going forward. The default and maximum value is **30** days. Learn about retention setting

30 days Save

Classroom: GitHub Classroom is used to host the class structure and assign assignments. In the classroom webpage, you can create new assignments, add students, and check student activity.



Assignment acceptance and starter code repositories will be changing on June 17, 2024. Review the changes and prepare your assignments.

Try our new experience for creating student assignment repositories using forks.

csc123kazerouni-nathan-testing
csc123kazerouni

Assignments

We value your feedback! Please take a few minutes to complete our survey.

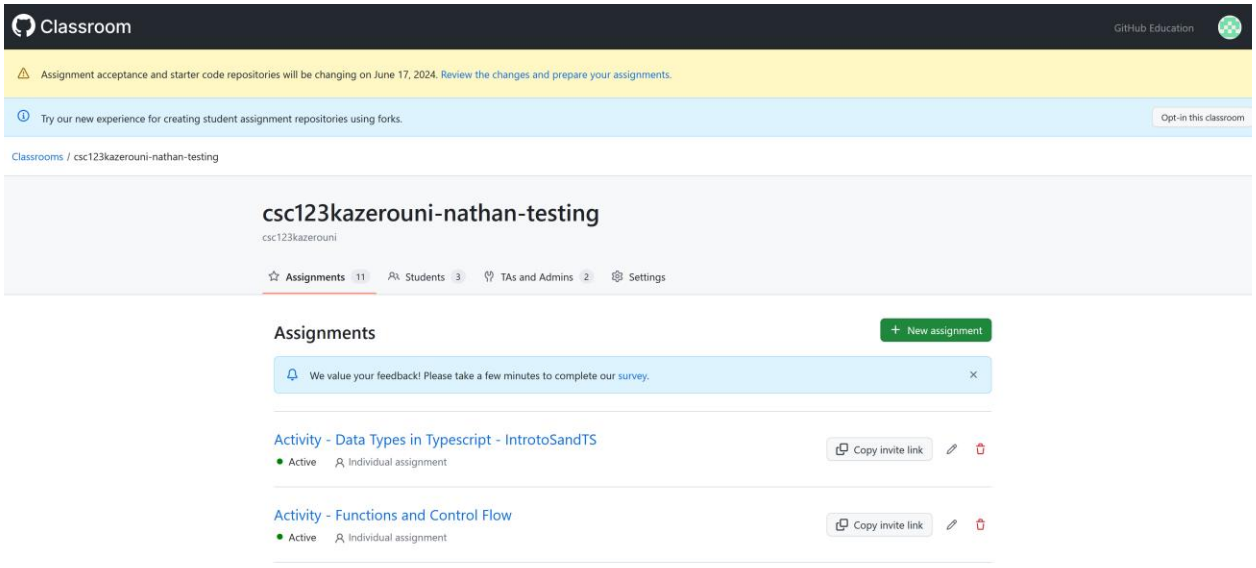
Activity - Data Types in Typescript - IntrotoSandTS
Active · Individual assignment

Activity - Functions and Control Flow
Active · Individual assignment

LiveShare: LiveShare is the VSCode extension used for any group projects. Usage documentation is included in the readmes under “Setup for Group Assignments”. Note: When using LiveShare, only the host uses Codespace hours.

Quick Start

Instructor and TA Setup: Access the GitHub Organization and GitHub classroom. In the organization, you can create and edit the assignment repositories. In the classroom, you can manage your class here:



Assignment acceptance and starter code repositories will be changing on June 17, 2024. Review the changes and prepare your assignments.

Try our new experience for creating student assignment repositories using forks.

Classrooms / csc123kazerouni-nathan-testing

csc123kazerouni-nathan-testing
csc123kazerouni

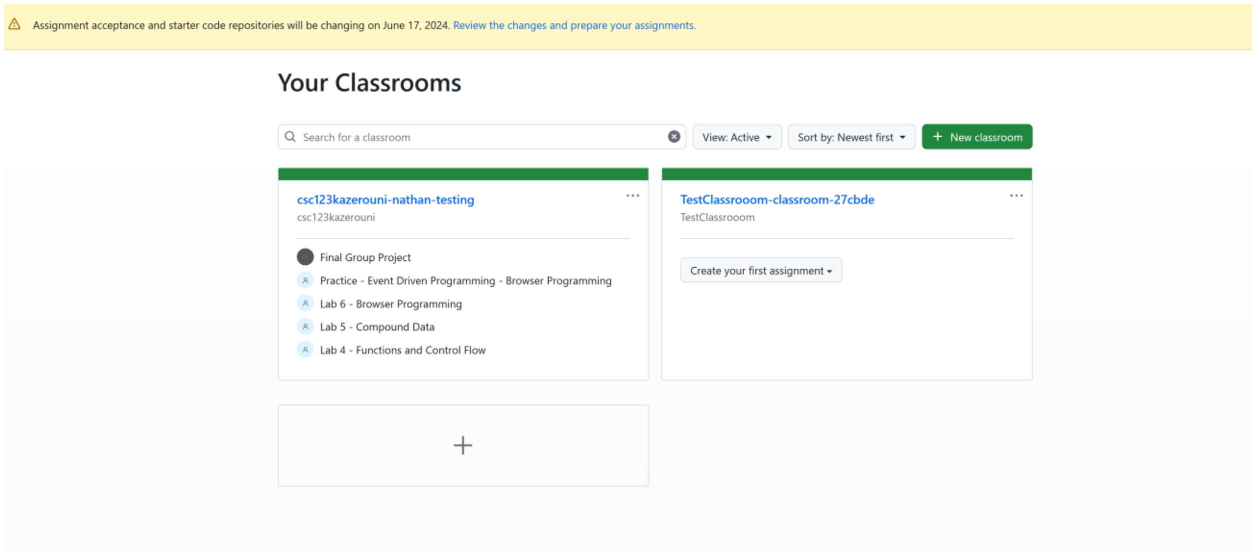
Assignments

We value your feedback! Please take a few minutes to complete our survey.

Activity - Data Types in Typescript - IntrotoSandTS
● Active   Individual assignment

Activity - Functions and Control Flow
● Active   Individual assignment

You can manage multiple classrooms within the same organization or within multiple organizations, as shown here:



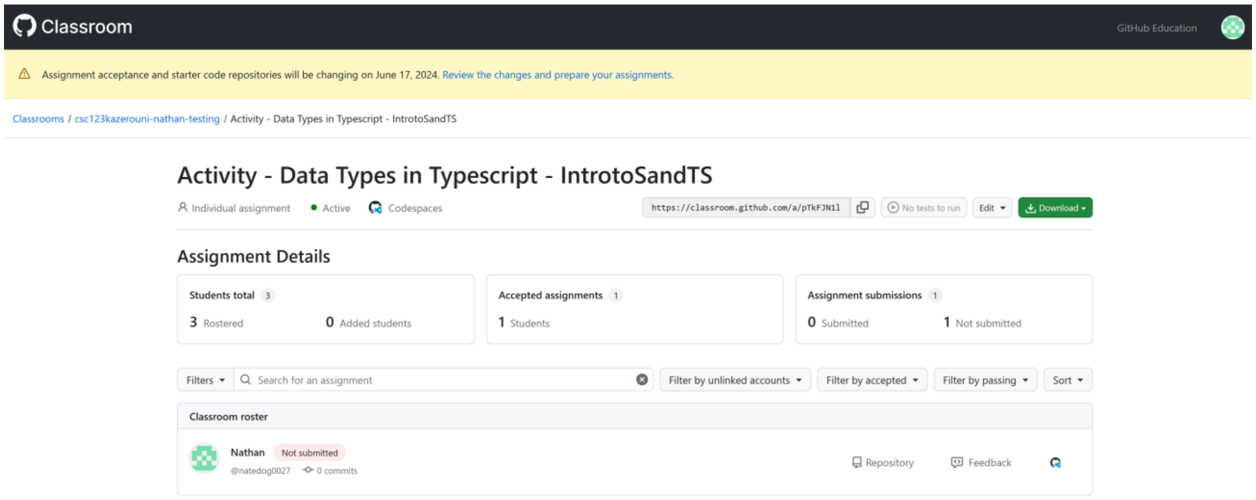
Assignment acceptance and starter code repositories will be changing on June 17, 2024. Review the changes and prepare your assignments.

Your Classrooms

csc123kazerouni-nathan-testing
csc123kazerouni
- Final Group Project
- Practice - Event Driven Programming - Browser Programming
- Lab 6 - Browser Programming
- Lab 5 - Compound Data
- Lab 4 - Functions and Control Flow

TestClassrooom-classroom-27cbde
TestClassrooom

You can click the new assignment button to create a new assignment based on a pre-built repository. If you select an assignment, you can invite students using the “copy invite link” button and send it to whoever you would like to join the classroom. In the assignment page, you can view the student’s activity and submissions:


Classroom

GitHub Education

Assignment acceptance and starter code repositories will be changing on June 17, 2024. Review the changes and prepare your assignments.

Classrooms / csc123kazerouni-nathan-testing / Activity - Data Types in Typescript - IntrotoSandTS

Activity - Data Types in Typescript - IntrotoSandTS

Individual assignment · Active · Codespaces · https://classroom.github.com/a/pTkFJN1l · No tests to run · Edit · Download

Assignment Details

Students total 3 — 3 Rostered, 0 Added students

Accepted assignments 1 — 1 Students

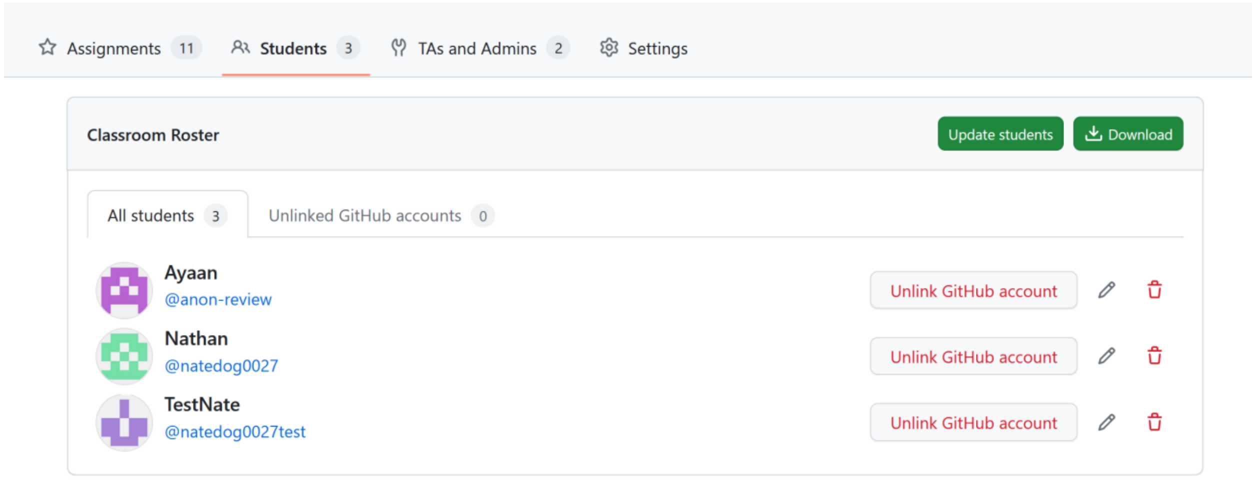
Assignment submissions 1 — 0 Submitted, 1 Not submitted

Classroom roster

Nathan — Not submitted — @natedog0027 — 0 commits — Repository — Feedback

If you click “repository”, you can view the students repository. If you select “feedback”, you can create comments on the students work or respond to request for assistance.

To add students to the classroom, simply invite them to an assignment by sending them an assignment link. You can view and edit the classroom roster under the students section:



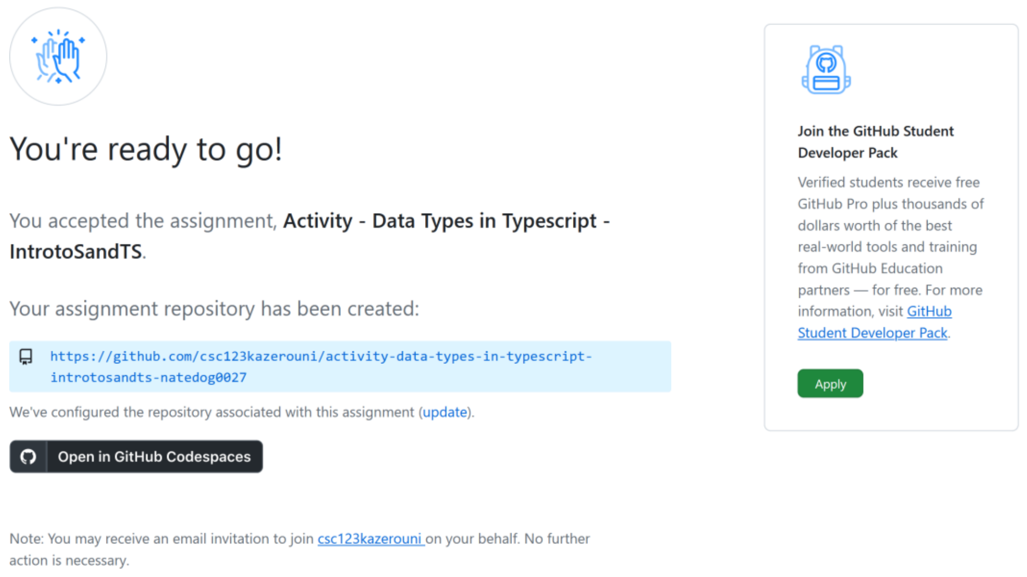
Classroom Roster

Ayaan @anon-review	Unlink GitHub account
Nathan @natedog0027	Unlink GitHub account
TestNate @natedog0027test	Unlink GitHub account

You need to be sure to create a list of student names in your class under the “update students” button. Then, when students accept the link to the classroom, they can link their GitHub account to their name and be set up in the roster.

Student Setup: At the very beginning of the quarter and as soon as possible, students should create a GitHub account and request educational benefits using this link: https://education.github.com/discount_requests/application. After they are approved, then they will have access to GitHub pro and have up to 180 core hours of GitHub Codespaces available per month. It is also advisable to get students acquainted with GitHub as a tool and instruct them ASAP on how to access assignments, setup Codespaces, and submit their work.

For assignments, students will receive a link sent to them by the instructor or TA. They will attach their GitHub account on the class roster and then they can get started on their assignment. Here is an image of what a student would see when clicking on an assignment link:



You're ready to go!

You accepted the assignment, **Activity - Data Types in Typescript - IntrotoSandTS**.

Your assignment repository has been created:

`https://github.com/csc123kazerouni/activity-data-types-in-typescript-introtosandts-natedog0027`

We've configured the repository associated with this assignment (update).

Open in GitHub Codespaces

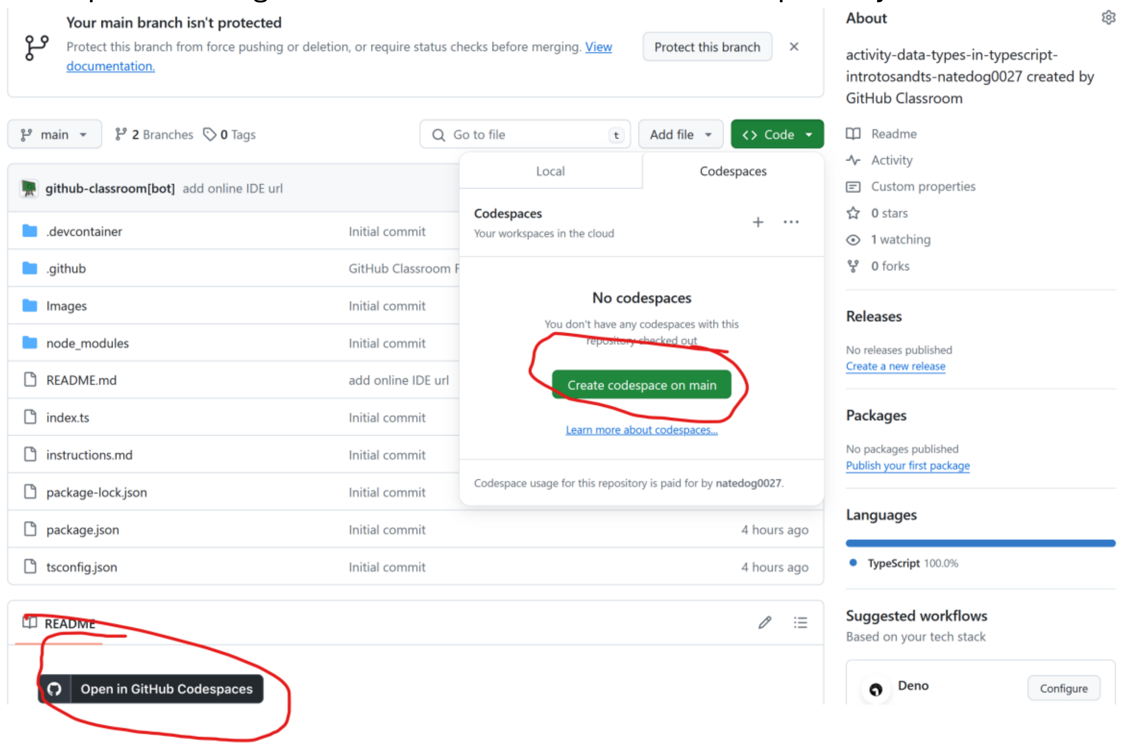
Note: You may receive an email invitation to join csc123kazerouni on your behalf. No further action is necessary.

Join the GitHub Student Developer Pack

Verified students receive free GitHub Pro plus thousands of dollars worth of the best real-world tools and training from GitHub Education partners — for free. For more information, visit GitHub Student Developer Pack.

Apply

They simply click the link to their repository and they are all set up. Next, they can open a Codespace clicking either of these two buttons in their repository:



Your main branch isn't protected

Protect this branch from force pushing or deletion, or require status checks before merging. View documentation.

Protect this branch

main · 2 Branches · 0 Tags · Go to file · Add file · Code

github-classroom[bot] add online IDE url

.devcontainer	Initial commit	
.github	GitHub Classroom F	
Images	Initial commit	
node_modules	Initial commit	
README.md	add online IDE url	
index.ts	Initial commit	
instructions.md	Initial commit	
package-lock.json	Initial commit	
package.json	Initial commit	4 hours ago
tsconfig.json	Initial commit	4 hours ago

Local · Codespaces

Codespaces
Your workspaces in the cloud

No codespaces
You don't have any codespaces with this repository checked out

Create codespace on main

Learn more about codespaces...

Codespace usage for this repository is paid for by natedog0027.

README

Open in GitHub Codespaces

About

activity-data-types-in-typescript-introtosandts-natedog0027 created by GitHub Classroom

Readme
Activity
Custom properties
0 stars
1 watching
0 forks

Releases
No releases published
Create a new release

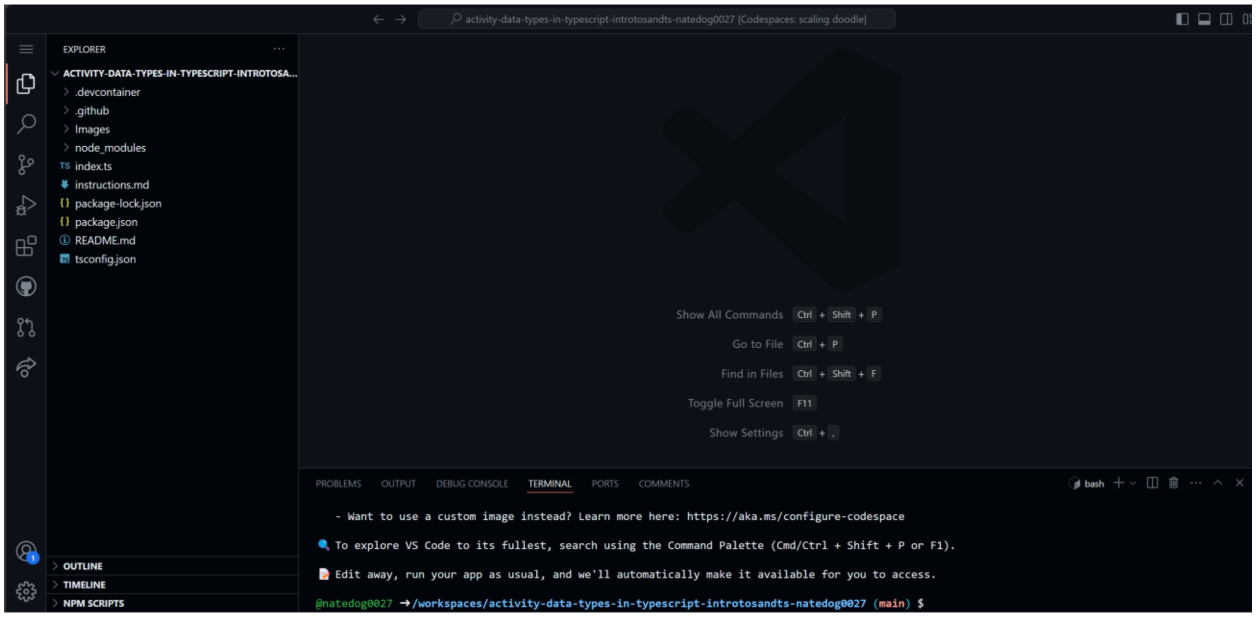
Packages
No packages published
Publish your first package

Languages
TypeScript 100.0%

Suggested workflows
Based on your tech stack

Deno · Configure

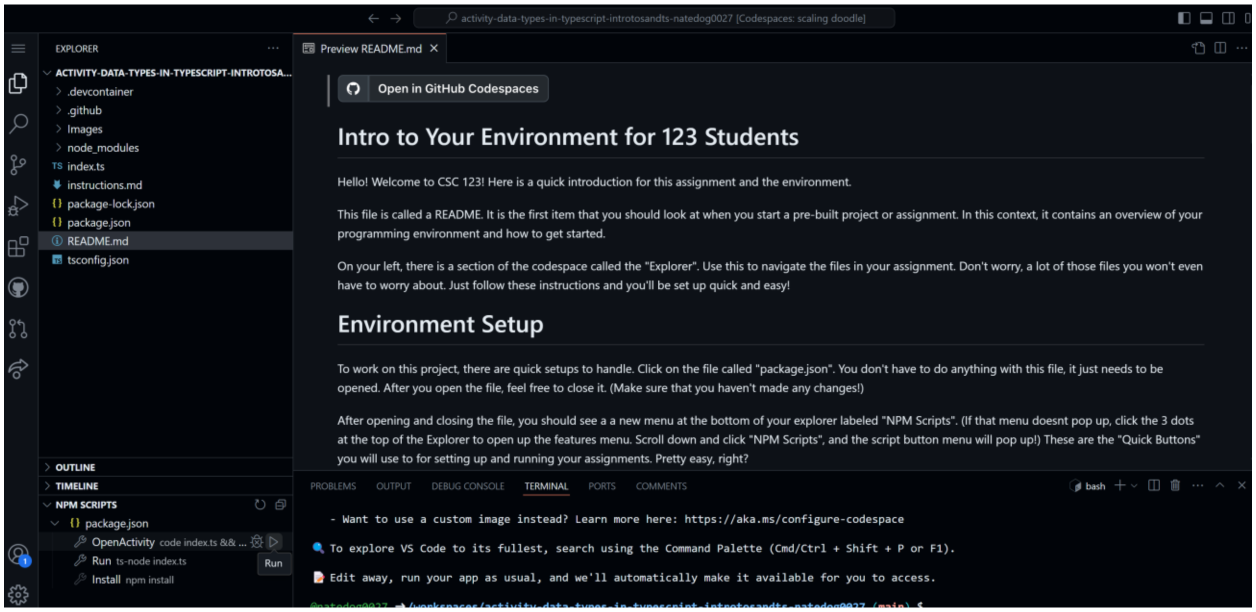
Their Codespace will look like this:



```
  - Want to use a custom image instead? Learn more here: https://aka.ms/configure-codespace

🔍 To explore VS Code to its fullest, search using the Command Palette (Cmd/Ctrl + Shift + P or F1).

📝 Edit away, run your app as usual, and we'll automatically make it available for you to access.

@natedog0027 ➜ /workspaces/activity-data-types-in-typescript-introtosandts-natedog0027 (main) $
```

The first thing that the student should open is the README.md in preview mode. They simply right click on the file and select “Open Preview”. This will open up information about the environment, how to start their assignment, and any other information necessary for them to access their assignment. Note: As of right now, the npm scripts sidebar automatically shows up in the bottom left because the package.json is in the root folder. Occasionally, updates remove this functionality. See the readme to check out alternative ways of getting the npm scripts sidebar to appear.

In the bottom left, the students will see the npm scripts sidebar. They simply expand it and then they will have the necessary scripts to run their assignment:



Intro to Your Environment for 123 Students

Hello! Welcome to CSC 123! Here is a quick introduction for this assignment and the environment.

This file is called a README. It is the first item that you should look at when you start a pre-built project or assignment. In this context, it contains an overview of your programming environment and how to get started.

On your left, there is a section of the codespace called the "Explorer". Use this to navigate the files in your assignment. Don't worry, a lot of those files you won't even have to worry about. Just follow these instructions and you'll be set up quick and easy!

Environment Setup

To work on this project, there are quick setups to handle. Click on the file called "package.json". You don't have to do anything with this file, it just needs to be opened. After you open the file, feel free to close it. (Make sure that you haven't made any changes!)

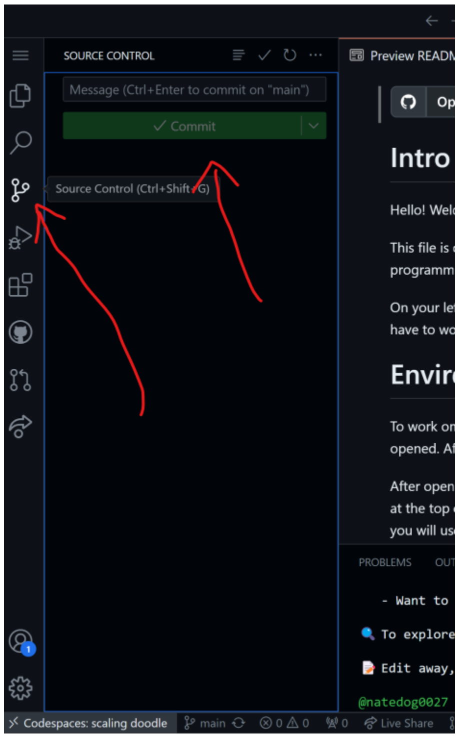
After opening and closing the file, you should see a a new menu at the bottom of your explorer labeled "NPM Scripts". (If that menu doesnt pop up, click the 3 dots at the top of the Explorer to open up the features menu. Scroll down and click "NPM Scripts", and the script button menu will pop up!) These are the "Quick Buttons" you will use to for setting up and running your assignments. Pretty easy, right?

```
  - Want to use a custom image instead? Learn more here: https://aka.ms/configure-codespace

🔍 To explore VS Code to its fullest, search using the Command Palette (Cmd/Ctrl + Shift + P or F1).

📝 Edit away, run your app as usual, and we'll automatically make it available for you to access.

@natedog0027 ➜ /workspaces/activity-data-types-in-typescript-introtosandts-natedog0027 (main) $
```

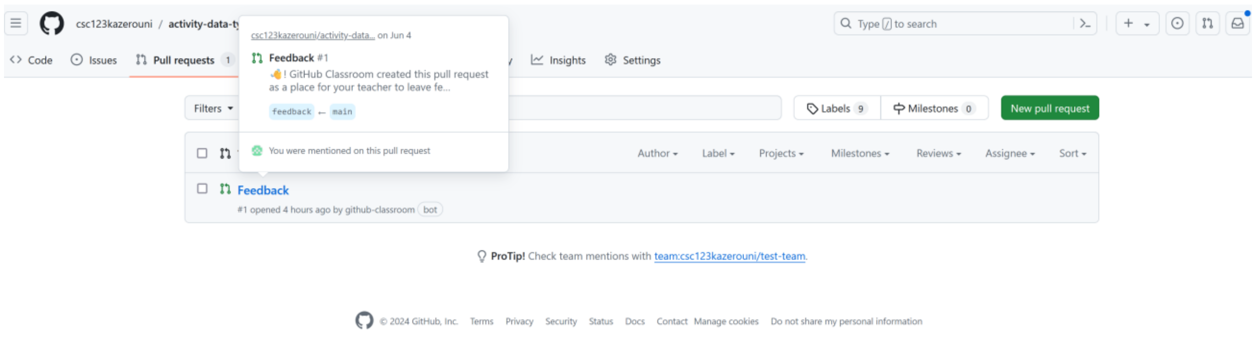
The scripts are buttons, and the students only have to click the “play” symbol next to their desired script to run it. Most assignments have 3 scripts labeled as “OpenAssignment”, “Run”, and “Install”. The TSServer assignments have a 4th script called “setup”. The “open” button opens up all of the necessary files to finish the assignment. It also opens up an instructions.md file, in which the TA or professor can provide instructions specific to the assignment. The “run” button runs the assignment or sets up the server in the browser to view their work. If the “setup” button is present, it must be pressed before the “run” button to ensure proper setup occurs for the TSServer assignments. The “Install” script installs the necessary dependencies for each assignment. There are instructions and image tutorials on how to use these in every assignment readme for the students.

When students press the run script, it sets up a server in their browser that auto updates when they make changes to the .html and .ts files within their program so that they can see their changes instantly.

To submit, students will go to “source control” in their sidebar menu, type in their commit message, and commit and then upload their assignments:


SOURCE CONTROL
Message (Ctrl+Enter to commit on "main")
✓ Commit
Source Control (Ctrl+Shift+G)
Preview READM
Intro
Envir
PROBLEMS
Codespaces: scaling doodle
main
Live Share

Once a commit has been submitted, the student’s assignment will show as submitted in the classroom and you can provide feedback. Students can view their feedback and ask questions in the “pull request” section of their repository under “Feedback”:



csc123kazerouni/activity-data... on Jun 4

Feedback #1

👋! GitHub Classroom created this pull request as a place for your teacher to leave fe...

feedback ← main

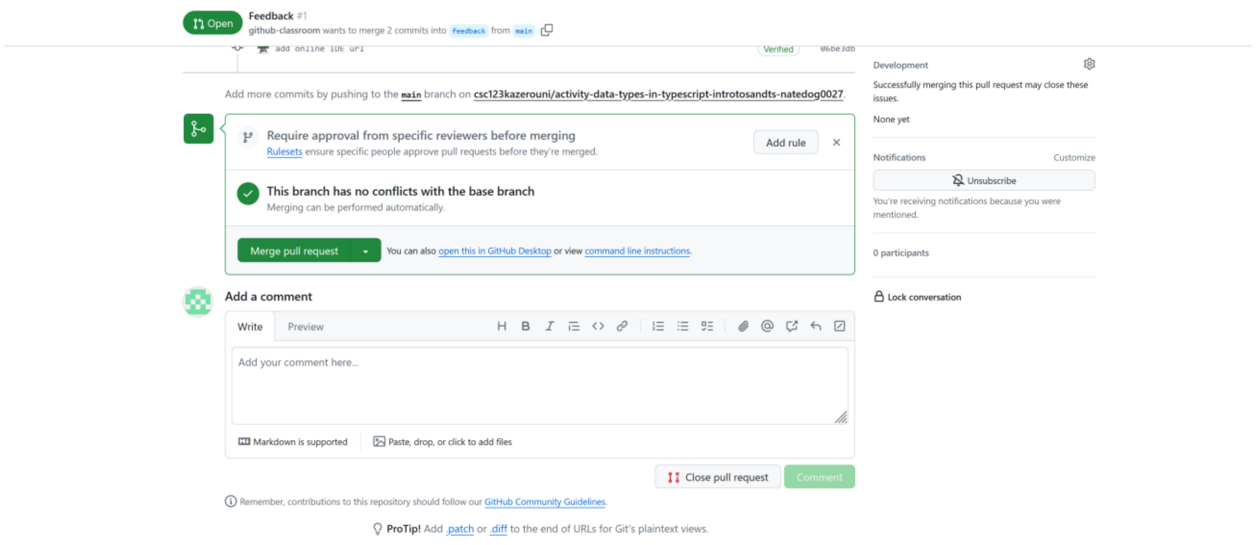
You were mentioned on this pull request

Feedback

#1 opened 4 hours ago by github-classroom bot

ProTip! Check team mentions with team:csc123kazerouni/test-team.

© 2024 GitHub, Inc. Terms Privacy Security Status Docs Contact Manage cookies Do not share my personal information



Feedback #1

github-classroom wants to merge 2 commits into feedback from main

add online IDE url

Add more commits by pushing to the main branch on csc123kazerouni/activity-data-types-in-typescript-introtosandts-natedog0027.

Require approval from specific reviewers before merging

Rulesets ensure specific people approve pull requests before they're merged.

This branch has no conflicts with the base branch

Merging can be performed automatically.

You can also open this in GitHub Desktop or view command line instructions.

Add a comment

Add your comment here...

Markdown is supported

Paste, drop, or click to add files

Remember, contributions to this repository should follow our GitHub Community Guidelines.

ProTip! Add .patch or .diff to the end of URLs for Git's plaintext views.

Development

Successfully merging this pull request may close these issues.

None yet

Notifications

You're receiving notifications because you were mentioned.

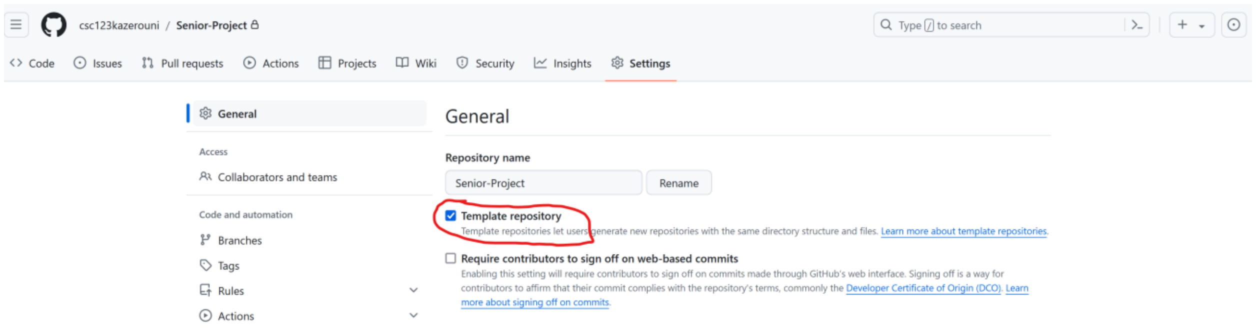
0 participants

Lock conversation

To summarize, students open their assignment, create a Codespace, read the readme, use the 3-4 scripts buttons to access their assignment, and submit using source control. While it seems like a lot to handle, it is a very simple and intuitive setup that they will quickly get acclimated to.

Overall Usage

Creating New Assignments: To create a new assignment, you can create a new repository using either the Template repo or the Senior-Project repo as the basis. In this new repository, you must go to repository settings and ensure that it is a template repository so that it can be used as a template for GitHub Classroom assignments:



General

Repository name
Senior-Project
Rename

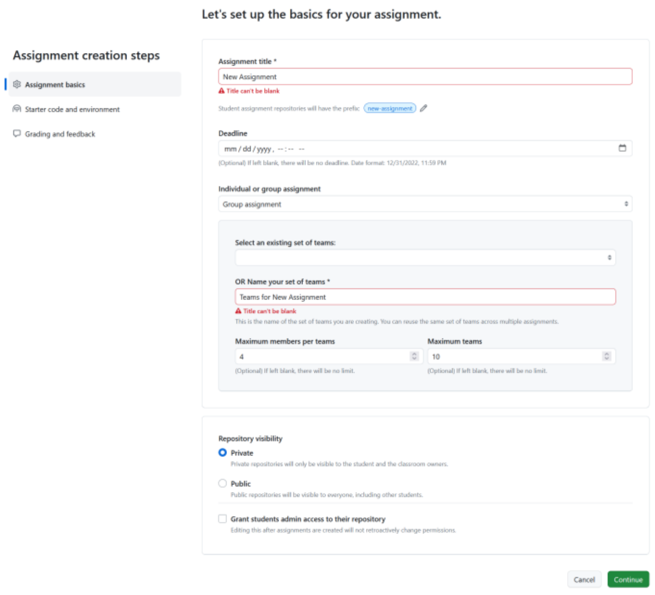
- [x] Template repository
  Template repositories let users generate new repositories with the same directory structure and files. Learn more about template repositories.
- [ ] Require contributors to sign off on web-based commits
  Enabling this setting will require contributors to sign off on commits made through GitHub's web interface. Signing off is a way for contributors to affirm that their commit complies with the repository's terms, commonly the Developer Certificate of Origin (DCO). Learn more about signing off on commits.

Then, you can create your new setup!

Uploading to GitHub Classroom: When creating a new assignment, you can set various attributes such as the deadline, whether it is individual or group (group assignments need to have a configured set of student teams), whether it is public or private (I have set all to private), adding admin access to students (I never select this option), a template repository as a basis (you must change the setting of the template repository to allow it to be a template), a supported editor (GitHub Codespaces), and whether to enable pull requests (Yes, for feedback).

Here are a few images of my typical configuration for assignment creation:

Page 1:



Let's set up the basics for your assignment.

Assignment title *
New Assignment
Title can't be blank
Student assignment repositories will have the prefix: new-assignment

Deadline
mm / dd / yyyy , -- : -- --
(Optional) If left blank, there will be no deadline. Date format: 12/31/2022, 11:59 PM

Individual or group assignment
Group assignment

Select an existing set of teams:

OR Name your set of teams *
Teams for New Assignment
Title can't be blank
This is the name of the set of teams you are creating. You can reuse the same set of teams across multiple assignments.

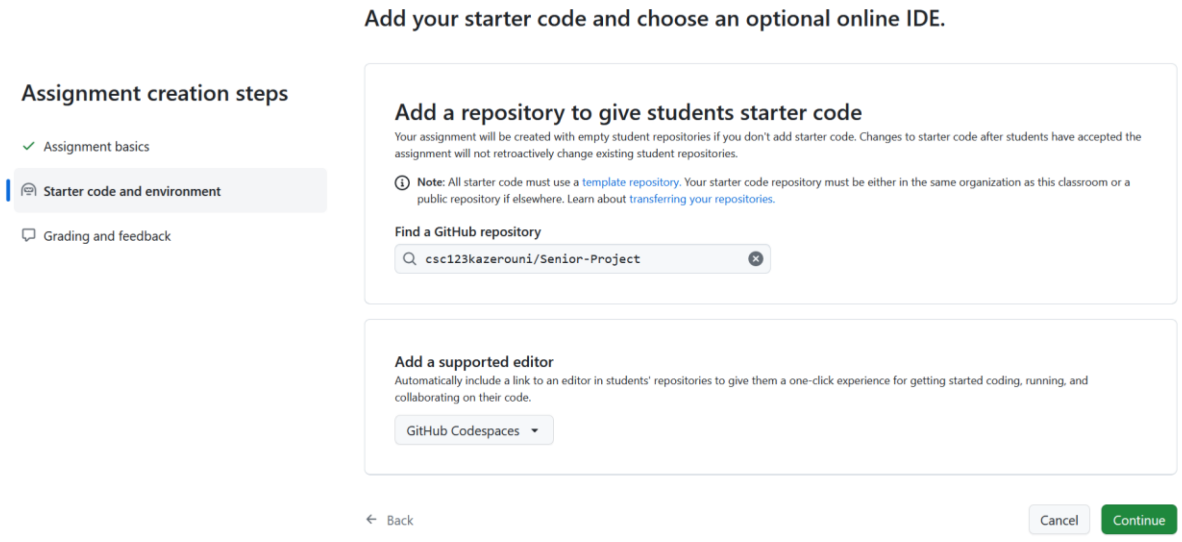
Maximum members per teams
4
(Optional) If left blank, there will be no limit.

Maximum teams
10
(Optional) If left blank, there will be no limit.

Repository visibility
[✓ Private]
Private repositories will only be visible to the student and the classroom owners.
Public
Public repositories will be visible to everyone, including other students.

- [ ] Grant students admin access to their repository
  Editing this after assignments are created will not retroactively change permissions.

Page 2:



Add your starter code and choose an optional online IDE.

Assignment creation steps

✓ Assignment basics

Starter code and environment

Grading and feedback

Add a repository to give students starter code

Your assignment will be created with empty student repositories if you don't add starter code. Changes to starter code after students have accepted the assignment will not retroactively change existing student repositories.

Note: All starter code must use a template repository. Your starter code repository must be either in the same organization as this classroom or a public repository if elsewhere. Learn about transferring your repositories.

Find a GitHub repository

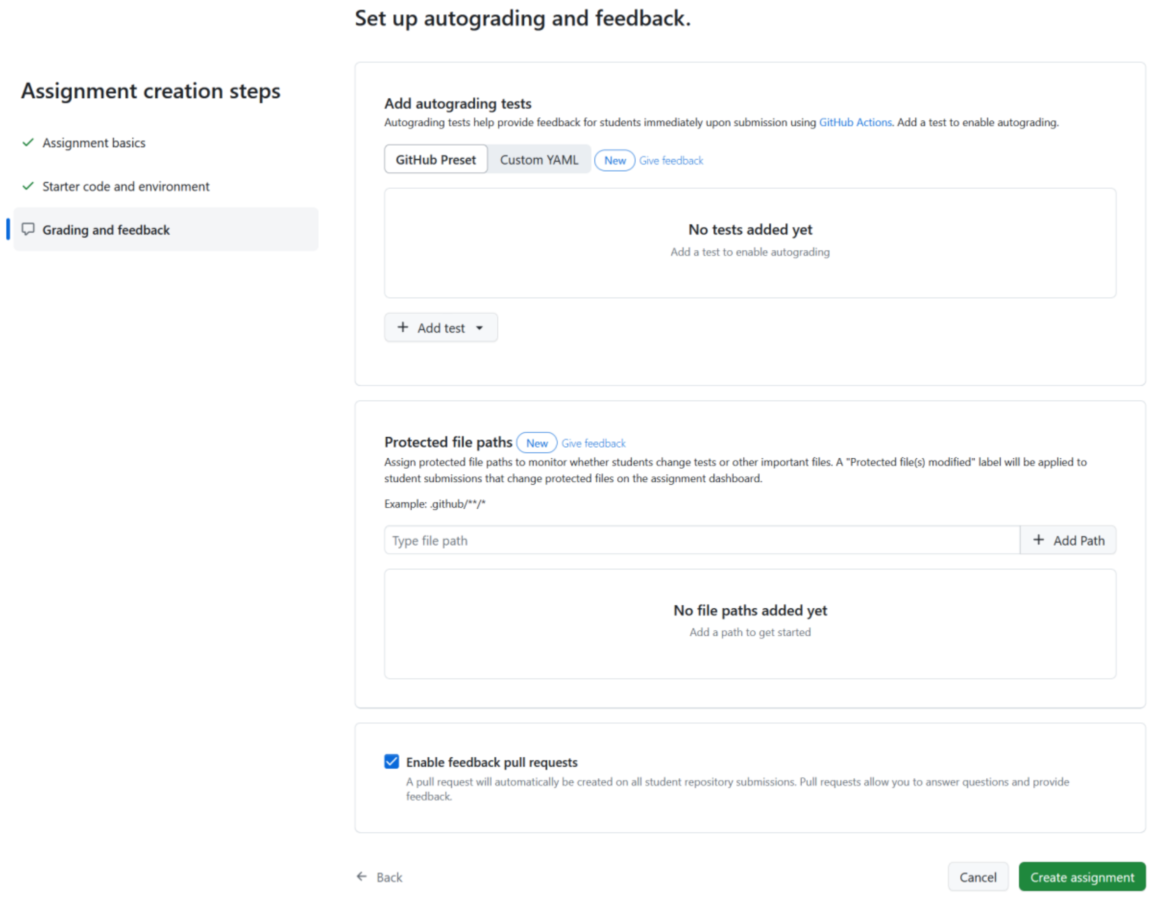
`csc123kazerouni/Senior-Project`

Add a supported editor

Automatically include a link to an editor in students' repositories to give them a one-click experience for getting started coding, running, and collaborating on their code.

GitHub Codespaces

Page 3:



Set up autograding and feedback.

Assignment creation steps

✓ Assignment basics

✓ Starter code and environment

Grading and feedback

Add autograding tests

Autograding tests help provide feedback for students immediately upon submission using GitHub Actions. Add a test to enable autograding.

No tests added yet

Add a test to enable autograding

Protected file paths

Assign protected file paths to monitor whether students change tests or other important files. A "Protected file(s) modified" label will be applied to student submissions that change protected files on the assignment dashboard.

Example: .github/**/*

No file paths added yet

Add a path to get started

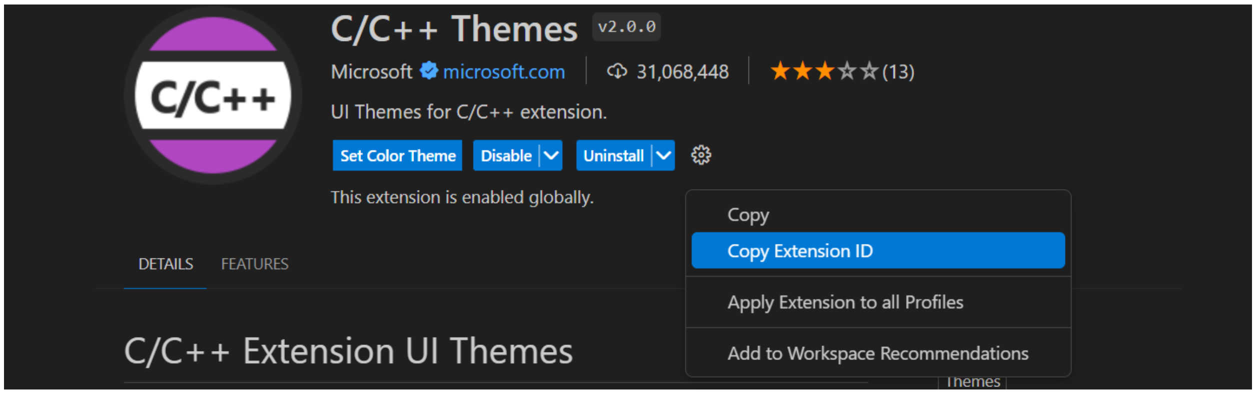
☑ Enable feedback pull requests

A pull request will automatically be created on all student repository submissions. Pull requests allow you to answer questions and provide feedback.

Devcontainer: Each repository has a .devcontainer file. This file has configurations for the Codespace so that students do not have to do any work on setup. As of right now, the devcontainer file only has configurations for auto-downloading LiveShare, a VSCode extension. I have left in the structure for opening files under "codespaces", but that setup has yet to work on Codespaces. Perhaps it will be useful in the future. Here is the devcontainer setup included in each repository:

```
{
    "name": "CSC123",
    "image": "mcr.microsoft.com/devcontainers/universal:2",

    "customizations": {
        "codespaces": {
            "openFiles": []
        },
        "vscode": {
            "extensions": [
                "ms-vsliveshare.vsliveshare"
            ]
        }
    }
```

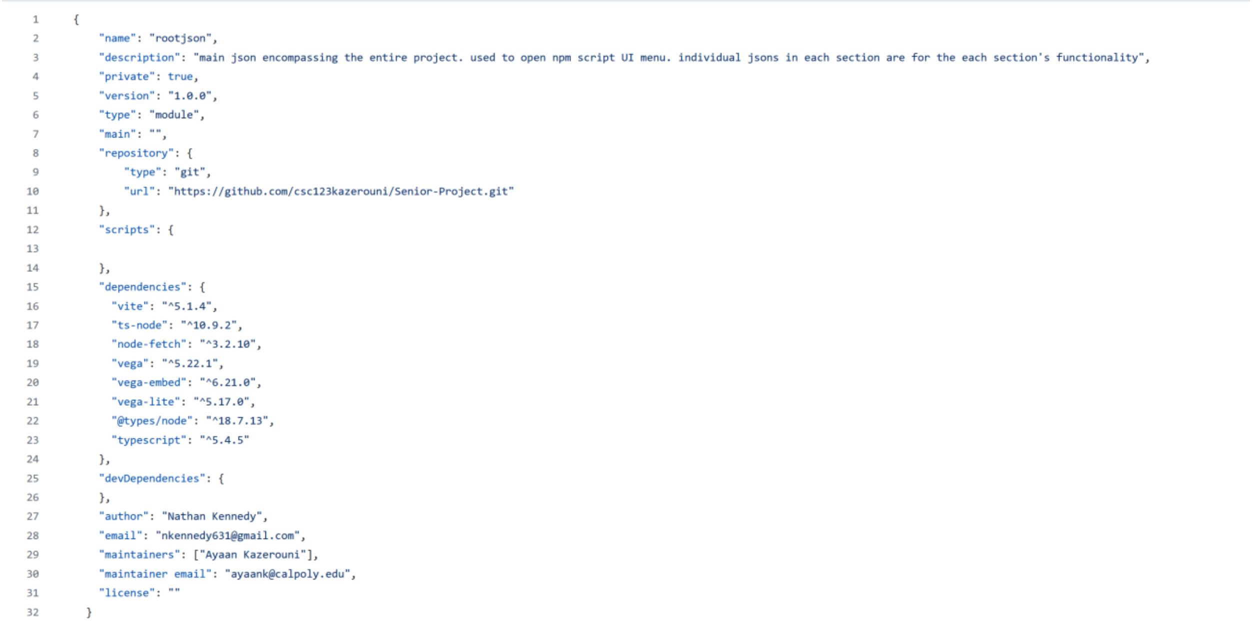
To add further extensions to the devcontainer, simply go to the extensions tab in VSCode, find your desired extension, select "Copy Extension ID", and then paste that ID under the "extensions" section in the devcontainer as shown above with LiveShare. Here is the image of where to get the extension ID:


C/C++ Themes
v2.0.0
Microsoft microsoft.com
31,068,448
(13)
UI Themes for C/C++ extension.
Set Color Theme
Disable
Uninstall
This extension is enabled globally.
DETAILS
FEATURES
C/C++ Extension UI Themes
Copy
Copy Extension ID
Apply Extension to all Profiles
Add to Workspace Recommendations
Themes

Extensions: Currently, only one extension is in use, which is:

- LiveShare: Creates a personal server for hosting other students to program in the same Codespace over a browser. Used for group projects.

Package.json: In each repository, there is a package.json in the root. Ensure that this file is in the root so that the NPM scripts sidebar will open for the students to use the script buttons. Here is the typical structure of a package.json file for each assignment:



```
{
    "name": "rootjson",
    "description": "main json encompassing the entire project. used to open npm script UI menu. individual jsons in each section are for the each section's functionality",
    "private": true,
    "version": "1.0.0",
    "type": "module",
    "main": "",
    "repository": {
        "type": "git",
        "url": "https://github.com/csc123kazerouni/Senior-Project.git"
    },
    "scripts": {

    },
    "dependencies": {
      "vite": "^5.1.4",
      "ts-node": "^10.9.2",
      "node-fetch": "^3.2.10",
      "vega": "^5.22.1",
      "vega-embed": "^6.21.0",
      "vega-lite": "^5.17.0",
      "@types/node": "^18.7.13",
      "typescript": "^5.4.5"
    },
    "devDependencies": {
    },
    "author": "Nathan Kennedy",
    "email": "nkennedy631@gmail.com",
    "maintainers": ["Ayaan Kazerouni"],
    "maintainer email": "ayaank@calpoly.edu",
    "license": ""
  }
```

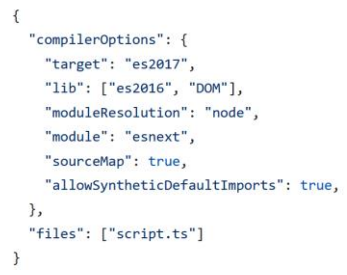
The sections to note are:

- Type: The type will typically be set to "module". However, as seen in the typescript assignments, the type will need to be blank to allow for the browser to properly read certain imports and exports for the .ts files.
- Main: Main will be set to whatever file that Vite (The server-running program) wants to open in the browser. In html assignments, it is typically set to "index.html". In the typescript assignments, it is typically set to "index.ts" or "script.ts".
- Scripts: This is the section where you can create a script with preset commands to run. These scripts can either be run in the command line by typing their name or run using the NPM scripts sidebar buttons.
- Dependencies: These are the software dependencies that each program uses to run. They can be set and then either clicking the "Install" script button or running "npm install" in the terminal will download and update these packages.

Here is a typical script setup for a TSServer assignment:

```
"scripts": {
  "OpenTSServerAssignment": "code index.html && code script.ts && code dataset.json && code instructions.md",
  "setup": "tsc -w -p tsconfig.json",
  "run": "vite",
  "Install": "npm install"
```

Note: A quirk for the TSServer assignment is that it uses a watch (-w) flag to watch the .ts file for changes in order to auto-update the .js file. However, instead of calling tsc on an “script.ts” file like normal, it calls it on a “tsconfig.json” file using the program (-p) flag. This allows us to configure the compiler to run using the “esnext” module, which allows for Vite to watch for typescript updates. This setup is necessary in order for the .ts -> .js compiler to properly compile the import and export commands, as there needs to be specific commands in order to run on a browser. The target file for Vite is under “files”, which is set to “script.ts” in this image:



```
{
  "compilerOptions": {
    "target": "es2017",
    "lib": ["es2016", "DOM"],
    "moduleResolution": "node",
    "module": "esnext",
    "sourceMap": true,
    "allowSyntheticDefaultImports": true,
  },
  "files": ["script.ts"]
}
```

Dependencies: There are many dependency combinations used in the various repositories and different assignments. Here are all of the ones used so far over all of the assignments:

- Vite: Build tool used to create a local webserver in the browser for development use.
- Ts-node: Used to compile and run typescript code without making a .js output file.
- Node-fetch
- Vega
- Vega-embed
- Vega-lite
- @types/node
- Typescript: Used to compile and run typescript code by making a .js output file.

Tips, Notes, and Quirks

- There is a lot of reused information. For example, the “Senior-Project” repo is a functional copy of the “File-Import-From-Replit” repo. The “Templates” repo are copies of actual assignments that were turned into basic setups. Each of the specific assignment repositories are functional offshoots of the items in the “Senior-Project” repo. This is for ease of use and replication. Just a heads up so that you are not surprised by the amount of information and data that is reused.
- While LiveShare does allow all invited users to access files and program on their own, only the host has access to the npm scripts sidebar. The host can give access to the terminal to other students, but that is not the simple script setup that is intended for the students. So, with group projects, everyone can program, but only the host can run the scripts from the npm scripts sidebar.
- The typescript in the command line assignments (not server) run using ts-node, which does not generate a .js file, it just runs the program. The server assignments generate a .js file. This is simpler for the students.

Other Useful Resources

Codespaces official documentation: https://docs.github.com/en/codespaces

GitHub official documentation: https://docs.github.com/en

GitHub pricing information: https://github.com/pricing

Webpage to setup student and instructor educational benefits to get GitHub Pro for free: https://education.github.com/discount_requests/application

Package.json basics and overview: https://nodesource.com/blog/the-basics-of-package-json/

Semver basics and overview (How to setup dependency versions in the package.json file): https://nodesource.com/blog/semver-a-primer/

Setting up a .devcontainer using Vite: https://dev.to/kkoziarski/react-vite-github-codespaces-5529

Devcontainer scripting setup commands and usage: https://containers.dev/implementors/json_reference/#lifecycle-scripts

While the entire GitHub documentation is useful, here are specific Github documentation pages that have proven useful for this project. While I have elaborated on as much of the necessary info as I could, if you have the time, I suggest going over these pages for sure before starting to use and manipulate this project. This is a significant amount of information, but I have tried to narrow down the documentation to the parts useful to this project in case it is necessary to use them:

- Setup Codespaces pre-configured: https://docs.github.com/en/codespaces/getting-started/deep-dive
- GitHub Codespaces pre-builds: https://docs.github.com/en/codespaces/prebuilding-your-codespaces/about-github-codespaces-prebuilds
- Codespaces with Classroom: https://docs.github.com/en/education/manage-coursework-with-github-classroom/integrate-github-classroom-with-an-ide/using-github-codespaces-with-github-classroom#enabling-codespaces-for-your-organization
- Persisting variables in Codespaces: https://docs.github.com/en/codespaces/developing-in-a-codespace/persisting-environment-variables-and-temporary-files
- Forwarding ports in Codespaces (For the group projects, Liveshare is setup for ease of student use. However, if port forwarding is required in the future, this information

could be useful): https://docs.github.com/en/codespaces/developing-in-a-codespace/forwarding-ports-in-your-codespace

- Rebuilding Codespace containers using the .devcontainer file: https://docs.github.com/en/codespaces/developing-in-a-codespace/rebuilding-the-container-in-a-codespace
- Devcontainer overview: https://docs.github.com/en/codespaces/setting-up-your-project-for-codespaces/adding-a-dev-container-configuration/introduction-to-dev-containers
- Setting up Codespaces for a specific project (In this example, it uses node.js): https://docs.github.com/en/codespaces/setting-up-your-project-for-codespaces/adding-a-dev-container-configuration/setting-up-your-nodejs-project-for-codespaces
- Automatically opening files in Codespaces using the devcontainer(As of when this project was created, this feature did not function. However, it may be useful in the future if GitHub corrects this issue): https://docs.github.com/en/codespaces/setting-up-your-project-for-codespaces/configuring-dev-containers/automatically-opening-files-in-the-codespaces-for-a-repository
- Creating Codespace links for sharing: https://docs.github.com/en/codespaces/setting-up-your-project-for-codespaces/setting-up-your-repository/facilitating-quick-creation-and-resumption-of-codespaces
- Codespace settings setup for a GitHub Organization: https://docs.github.com/en/codespaces/managing-codespaces-for-your-organization/choosing-who-owns-and-pays-for-codespaces-in-your-organization
- Codespaces command pallet usage: https://docs.github.com/en/codespaces/reference/using-the-vs-code-command-palette-in-codespaces
- Working collaboratively in Codespaces (Very simple LiveShare overview. I elaborate further on LiveShare usage in the readme docs of the group project items): https://docs.github.com/en/codespaces/developing-in-a-codespace/working-collaboratively-in-a-codespace

Useful videos and tutorials:

- Instantiating Codespaces with dependencies: https://www.youtube.com/watch?v=8KwoKgYz85k
- JSON crash course: https://www.youtube.com/watch?v=6OhMbf2v_jI
- Creating assignments in GitHub Classroom: https://www.youtube.com/watch?v=6QzKZ63KLss

- How student complete assignments in GitHub classroom: https://www.youtube.com/watch?v=ObaFRGp_Eko
- How instructors review assignments in GitHub classroom: https://www.youtube.com/watch?v=g45OJn3UyCU

<u>Final Thanks</u>

This senior project has been a very exciting and formative task for me these last 2 quarters. While there was a definite structure and end goal, I wholly enjoyed the freedom and open-endedness granted to me during this project. It was a fresh change of pace to start from scratch and have to perform my own research to find resources and useful tools in order to complete this task. It was also very exciting to create something that could possibly have a significant positive impact for future generations of students and programmers starting out their higher education journey. When I first started studying computer engineering, I had no experience with programming. It was an entirely new concept for me, and my early classes were wrought with feelings of inadequacy and extreme confusion as I tried to keep up with the rigors of this major. I didn't even know what an engineer was! I initially chose this major because I enjoyed technology, and I heard that engineers made lots of money. Boy did I set myself up, ha-ha. But I stuck with it, and I am finally graduating with this degree. I couldn't be more proud or excited about my accomplishment. If this project that I have built can make it easier for new students to acclimate to this educational pathway, then I have accomplished what I intended when I started it.

I greatly appreciate the opportunity to work with Ayaan Kazerouni on this project. He is an excellent instructor and mentor. Interestingly enough, I was actually a student in one of his first classes at Cal Poly in 2020. It was CPE 203. Now, he is my advisor for senior project in my last 2 quarters at Cal Poly. My education has now come full circle. I am certain that future generations of students will continue to be grateful for the care and expertise that he has brought to this program at Cal Poly and beyond.

Thank you,

Nathan Kennedy

Computer Engineering '24

:)