Collaborate on ArcGIS Web-app development using Git repositories

Esri Canada GIS Centres of Excellence App Challenge, February/March 2021

Overview

In this guide, you will learn how to set up a Git repository to collaborate on Web applications for the ECCE App Challenge. This allows each of your team members to work on their own customizations for the app's HTML, CSS, and JavaScript code, while maintaining one central copy to which all customizations are added and synchronized.

Introduction

In the Esri Canada GIS Centres of Excellence App Challenge, your group will create an application using the Esri platform. This could be a Web application using the ArcGIS API for JavaScript, such as a custom app developed from scratch, or an app built with the ArcGIS configurable application templates (e.g., Map Styler, Classic Story Maps) or the Web AppBuilder for ArcGIS. The application could also use any of the ArcGIS software development kits for native apps ("SDKs" – e.g., for Android, iOS, Windows, Java, etc.) or the AppStudio for ArcGIS, among other options.

This document describes a workflow that allows your group to modify the code for ArcGIS configurable apps or Web AppBuilder apps collaboratively using Git repositories. Parts 1 and 2 of the document are also applicable if you are using a different framework, such as the ArcGIS SDKs or the AppStudio, or if you are creating an ArcGIS Web application from scratch using custom HTML, CSS, and JavaScript.

Part 1: Create a Git repository

Creating a Git repository (or "repo") to store your Web application code provides a reliable way to collaborate on application development. Compared to other approaches, such as editing a single version of the application on a network share or in cloud storage (e.g., Google Drive, Dropbox, or Microsoft OneDrive), using Git avoids situations where files are unavailable for editing due to being "locked" by other users, or where team members edit files concurrently and overwrite each other's edits when saving.

Using Git, you can create a "remote" repository, which will be used to store the master version of your application, and each member of your team can create a separate, "local" version of the repository. Each team member can use their local repository to store their own modifications to the application, and then "push" these modifications to the remote repository when ready. Likewise, each team member can "pull" the latest changes from the remote repository to keep their local repository in sync with the other members' updates. For more information on Git, please see https://git-scm.com/. Please note that a full introduction to the capabilities of Git is outside the scope of this tutorial.

One recommended approach for creating a Git repository for the App Challenge is to use GitHub, which allows you to create a private repository for free, for a team of up to four users. Creating a private repository is recommended so that your code is not publicly accessible during the ECCE App Challenge.

- **Note:** You may choose to use another online Git repository host such as **GitLab** or **Bitbucket** to create your private repo. Or, if you have access to a shared network drive (e.g., in a computer lab), you could manually create a remote Git repo there instead of in the cloud. However, for the purposes of this section of the tutorial, we will assume you are using GitHub.

To create a Git repository for the App Challenge on GitHub:

1. Have each team member create an account at https://github.com/, and make sure to confirm your email address before continuing.

   Have one team member click the **Start a project** button, which will walk you through the process to create the Git repository that will store your app's code. Enter a name for the repository and an optional description, click the **Private** option underneath, and check the option to "Initialize this repository with a README".

   Click **Create repository** when finished. You will be taken to the page for the new repository.

2. On the page for the new repository, click the **Settings** tab towards the top of the page. Click the **Collaborators** tab on the left, and add each member of your team by entering their GitHub username or their email address into the search box beside the **Add collaborator** button.

Next, each team member can download ("clone") the Git repo to their computer using a Git client. Many Git clients exist; however, for this tutorial, we will assume you are using **GitHub Desktop**.

1. Open GitHub Desktop, click "Sign in to GitHub.com", and sign in with the GitHub account you created earlier. Enter a name and email address that will appear alongside your app modifications, choose whether to submit anonymized usage data, and click **Finish**.
2. You will see a list of your GitHub repositories in the right-hand panel of the window. Click the repository you created earlier, and then click the **Clone** button that appears for it at the bottom of the panel.
3. In the window that appears, you can change the "local path" if needed – this is where the repository will be saved on your computer. When ready, click **Clone**, and the repository will be cloned to your computer.

You will be taken to the main screen in GitHub Desktop for managing the repository. This is where you can view the changes you have made in the repository folder since your last "commit", commit your changes to the repository, "push" your commits to the origin (i.e., the version of the repository hosted on GitHub.com), and "fetch" and "pull" your teammates' commits to keep your version in sync.

Part 2: Create a group in ArcGIS Online (optional)

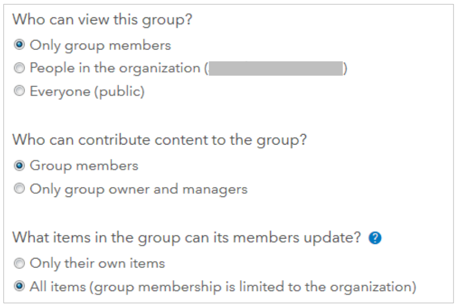
You can use the Git repository from the previous part of the tutorial to store your application code. However, if you want to collaborate on Web maps and spatial data in **ArcGIS Online** (or in an **ArcGIS Enterprise** portal hosted by your institution), without having your data available to the public, you will need an ArcGIS Online account for each team member, as well as a **group** in ArcGIS Online. This will allow you to share your Web maps, data, and applications privately within your team.

1. If your team members do not have ArcGIS Online or ArcGIS Enterprise accounts provided by your institution, you will want to create a separate **ArcGIS for Developers** account for each team member at https://developers.arcgis.com.

2. Once each team member has an account, one member can create a group in ArcGIS Online using the instructions at https://doc.arcgis.com/en/arcgis-online/share-maps/create-groups.htm.

3. Configure the group privacy and sharing settings so that they appear similarly to the screenshot at right. Note that the bottom radio button should only be set to "All items" if you are all members of the same organization (i.e., if you are not using your own individual ArcGIS for Developers accounts). Otherwise, you will not be able to invite other users to the group.

4. Invite your team members to the group, and share your ArcGIS Online content with the group.



Who can view this group?
● Only group members
○ People in the organization ([redacted])
○ Everyone (public)

Who can contribute content to the group?
● Group members
○ Only group owner and managers

What items in the group can its members update?
○ Only their own items
● All items (group membership is limited to the organization)

IMPORTANT: At the end of the App Challenge, if you have shared any of your ArcGIS Online items (i.e., Web maps, feature layers, apps, etc.) with only this group, you will want to update the sharing settings for all items so that they are viewable by "Everyone (public)". This can be accomplished in the **Content** page for your ArcGIS Online account.

Part 3: Create an ArcGIS Online Web mapping application

Next, you can create and configure the Web application in ArcGIS Online and download the application code to modify it for the App Challenge. If you plan to create an application "from scratch," that is, by writing all code manually, you can disregard the rest of this tutorial and add your code directly to your local repositories from Step 1 above.

Otherwise, there are two main options for creating Web applications on the ArcGIS platform: using configurable app templates or creating an app using the Web AppBuilder for ArcGIS. The steps to collaborate on either type of app are described in the sections below.

Option A: Configurable app templates

When building a Web mapping app using one of the ArcGIS Online configurable app templates – such as the Basic Viewer or Map Styler templates, or a Classic Story Map template – the configuration data for your app is stored in ArcGIS Online. The configuration data includes which Web maps, widgets, layouts, or colour schemes you selected in the app's online configuration tool. However, in addition to these configuration options, you can also export the app from ArcGIS Online to make custom changes to the app's HTML, CSS, and JavaScript. Using Git allows your team to collaborate on these modifications.

1. In ArcGIS Online, create a new Web application. You can do this by going to your account's **Content** page and clicking the **Create** button, or by creating a Web map in the ArcGIS Online Map Viewer and clicking **Share** in the Map Viewer toolbar.

2. Select a template for your application and configure it as needed. The exact steps for this will depend on the template you choose and are outside the scope of this tutorial. You can return later to make further changes to the app.

3. If you created an ArcGIS Online group for your team in the previous part of this tutorial, you can share the app with your team members. Go to your **Content** page in ArcGIS Online, click the checkbox for the application you just created, click **Share**, and share it with the group you created earlier.

Next, you can download and modify the code for your application.

4. On the help page for the ArcGIS Online configurable app templates, click the link for the template your app uses. For most templates, this will take you to the template's item page on ArcGIS Online. On the template's item page, click the **Download** button, which will take you to the GitHub repository for the application template. Alternatively, you can find the template's GitHub repository at https://github.com/Esri.

5. On the GitHub repository Web page, click the **Releases** tab towards the top of the page, and download the zipped source code for the latest version of the template. Unzip the source code into the local repository you set up in Part 1 of this document.

6. You will need to modify the template by inserting an app ID: this tells the template code on your machine which app it should load from ArcGIS Online (i.e., the app you created and configured in steps 1 and 2 above). The instructions for inserting the app ID can be found on the home page of the GitHub repository for your template.

7. Return to your Git client (e.g., GitHub Desktop, as described in Part 1), and the template's files should appear in the list of uncommitted changes. Commit these changes, adding a summary of the changes such as "Adding configurable app template code", and then "push" your new commit to the origin.

8. Next, each team member can fetch and pull this commit from the origin, which will download the template files onto each team member's computer.

You can now all make modifications to the template's HTML, CSS, and JavaScript. Make sure that everyone commits, pushes, and pulls the changes on a regular basis (e.g., as new sections of code are added). The team member who created the app in ArcGIS Online can also continue making changes using the online configuration tools from step 2 above.

To test your app in a Web browser, you will need to publish the app to a Web server. This could be a public Web server, such as one hosted by a research lab at your institution, or a local Web server such as Microsoft IIS or Apache installed on your own computer. You may also consider using a code editor with live previews such as Brackets, or Visual Studio Code with the Live Server extension.

Option B: Web AppBuilder for ArcGIS (for editing app code only)

If you are creating a Web AppBuilder app for the ECCE App Challenge, you can export the code for your app to make changes not possible through the main Web AppBuilder editor interface, such as creating a custom widget or theme using HTML, CSS, and JavaScript. This requires the **Web AppBuilder for ArcGIS (Developer Edition)**. After exporting the application's code, your team can use Git to collaborate on modifications to the code.

NOTE: This workflow exports a copy of your app that will exist outside of the Web AppBuilder: that is, after you export the app to store it in the Git repository, you will need to make changes to the app by editing the HTML, CSS, or JavaScript code in the repository, instead of using the Web AppBuilder editor interface.

1. One team member should download the Web AppBuilder for ArcGIS (Developer Edition): click "Download the SDK" at https://developers.arcgis.com/web-appbuilder/, and unzip the Web AppBuilder folder to a location on your hard drive.

2. Set up the Web AppBuilder as described in the video at https://youtu.be/5CK0xoELpOo?t=431 (watch up to approximately 10:15 in the video). Once finished, the Web AppBuilder home page will appear in your Web browser, at a URL similar to http://[computer_name]:3344/webappbuilder/.
3. Create a new Web AppBuilder app, or import an existing Web AppBuilder app from your ArcGIS Online account. You can use the Web AppBuilder editor interface to make changes to the app as desired.
    - If you import an existing app from your ArcGIS Online account, note that this app is now separate from the version you initially created in ArcGIS Online. That is, if you make changes to the app using the version of the Web AppBuilder on your computer, these will not be reflected in the copy of the app in ArcGIS Online, and vice versa.
4. Once finished creating, editing, or importing your app, return to the Web AppBuilder home page (i.e., the URL from step 2 above), and click the ⤓ (Download) button at the bottom of the card for the app. This will provide you with a zip file containing all the code for the app.
5. Create a folder in the local repo you set up in Part 1 of this document, and unzip the app into this folder.
6. Return to your Git client (e.g., GitHub Desktop, as described in Part 1), and the app's files should appear in the list of uncommitted changes. Commit these changes, adding a summary of the changes such as "Adding Web AppBuilder app code", and then "push" your new commit to the origin.
7. Next, each team member can fetch and pull this commit from the origin. This will download the code for the Web AppBuilder app onto each team member's computer.

You can now all make modifications to the Web AppBuilder app's HTML, CSS, and JavaScript. Make sure that everyone commits, pushes, and pulls the changes on a regular basis (e.g., as new sections of code are added).

To view and test your app in a Web browser, you will need to publish the app to a Web server. This could be a public Web server, such as one hosted by a research lab at your institution, or a local Web server such as Microsoft IIS or Apache installed on your own computer. You may also consider using a code editor with live previews, such as Brackets, or Visual Studio Code with the Live Server extension.

Conclusion

Each team member of your ECCE App Challenge team now has their own version of the configurable app or Web AppBuilder app, to which they can each make their own modifications separately. When each team member wants to make their modifications available to the rest of the team, they can return to their Git client, create a commit containing all their latest changes, and push their changes to the origin. Other team members can receive these changes by fetching and pulling the latest changes from the origin.

Happy developing!

https://ecce.esri.ca