Essential Apps Script livecoding 13 walkthrough: Add permissions

This walkthrough guide accompanies the Essential Apps Script 13: Add permissions video and the Essential Apps Script guide. You can use the video, this document, or both, to help you do this livecoding exercise.

In this exercise, we are going to use Apps Script to add sharing permissions to files in Google Drive.

Livecoding instructions

1. In your Google Drive folder for the course, create a new Google Doc that we can use to share and call it **Add Permissions Template File** then leave the Doc open so we can get its ID later. Back in the Google Drive folder, create a new Google Sheet and call it **Livecoding add permissions** then open the Apps Script editor from **Extensions** > **Apps Script** and give the project the same name.

2. Rename the script file on the left hand side to **addPermissions** and then rename the function to **addPermissions()** as well. Put your cursor at the end of the line and hit Enter twice.

```
function addPermissions(){



}
```

3. To add permissions to a file we need to get the file and to do that we need the file ID, so write a comment **// get the file ID** and then hit Enter. On the new line, create a variable **var fileID =** and then put a set of single or double quotes e.g. **""**. Go back to the Google Doc you created earlier and highlight everything after the *d/* and before the */edit* to get the file ID, then copy and paste that inside the quotes. End the line with a semicolon.

```
function addPermissions(){

    // get the file ID
    var fileID = "ENTER DOCUMENT ID HERE";

}
```

4. Hit Enter twice, then create a new comment **// get the file** and hit Enter again. Create a variable using **var file =** and then we are going to use **DriveApp.getFileById()** to get the file. Inside the brackets, put the name of the variable containing the ID, **fileID** and then end the line with a semicolon.

```
function addPermissions(){

    // get the file ID
    var fileID = "ENTER DOCUMENT ID HERE";

    // get the file
    var file = DriveApp.getFileById(fileID);

}
```

5. Now we can add permissions. We will add both a commenter and an editor.

    a. Write a comment **// add a 'Commenter'** and then hit Enter. We are working with the file so write **file.addCommenter()** and then inside the brackets put an email address inside single or double quotes e.g. **'my-email@york.ac.uk'** and then end the line with a semicolon.

    b. Hit Enter twice and then write another comment **// add an 'Editor'** and then hit Enter. Write **file.addEditor()** and then put another email address inside single or double quotes e.g. **'another-email@york.ac.uk'** and then end the line with a semicolon.

    ```
    function addPermissions(){

        // get the file ID
        var fileID = "ENTER DOCUMENT ID HERE";

        // get the file
        var file = DriveApp.getFileById(fileID);

        // add a 'Commenter'
        file.addCommenter('my-email@york.ac.uk');

        // add an 'Editor'
        file.addEditor('another-email@york.ac.uk');

    }
    ```

6. Hit Save and then run the function. You will get the authorisation popup to use the DriveApp and then we can check the code worked by checking the Google Doc using the **Share** button to see if the sharing permissions have changed.

Now you could have a go at adding sharing permissions to a project you've made.

Reference: full code from exercise:

```
function addPermissions(){

    // get the file ID
    var fileID = "ENTER DOCUMENT ID HERE";

    // get the file
```

```
    var file = DriveApp.getFileById(fileID);

    // add a 'Commenter'
    file.addCommenter('my-email@york.ac.uk');

    // add an 'Editor'
    file.addEditor('another-email@york.ac.uk');

}
```