
Professional Certificate in Google Apps Script Advanced Techniques

Creating and Managing Google Drive Files with Apps Script

Google Drive is a cloud-based storage service that allows users to store, share, and access files from any device with an internet connection. Google Apps Script is a JavaScript-based scripting language that enables users to automate, extend, and integrate Google Drive and other Google services. In this explanation, we will cover key terms and vocabulary related to creating and managing Google Drive files with Google Apps Script.

1. **Google Drive File:** A Google Drive file is a document, spreadsheet, presentation, or other type of file that is stored on Google Drive. Google Drive files can be created and managed using Google Apps Script.
2. **Google Apps Script:** Google Apps Script is a JavaScript-based scripting language that enables users to automate, extend, and integrate Google Drive and other Google services. Google Apps Script can be used to create custom functions, add-ons, and scripts that interact with Google Drive files.
3. **DriveApp:** DriveApp is a Google Apps Script class that provides methods for interacting with Google Drive files. DriveApp can be used to create, upload, delete, and manage Google Drive files.
4. **File:** A File is a Google Apps Script class that represents a Google Drive file. The File class provides methods for getting and setting file metadata, such as the file name, description, and MIME type.
5. **Folder:** A Folder is a Google Apps Script class that represents a Google Drive folder. The Folder class provides methods for creating, deleting, and managing folders and the files within them.
6. **MIME Type:** A MIME type is a standard that identifies the type of data contained in a file. For example, the MIME type for a text file is "text/plain". Google Apps Script provides methods for getting and setting the MIME type of a Google Drive file.
7. **Permissions:** Permissions are the access levels granted to users for a Google Drive file. Google Apps Script provides methods for granting and revoking permissions for Google Drive files.
8. **User:** A User is a Google Apps Script class that represents a user in a Google Workspace domain. The User class provides methods for getting information about a user, such as their email address and name.
9. **Triggers:** Triggers are events that can trigger the execution of a Google Apps Script function. Google Apps Script provides several types of triggers, including time-driven triggers, which can be used to automate the creation and management of Google Drive files.
10. **addEditor(user):** The addEditor(user) method is a DriveApp method that grants edit permissions to a user for a Google Drive file.
11. **addViewer(user):** The addViewer(user) method is a DriveApp method that grants view permissions to a user for a Google Drive file.
12. **createFile(name, content):** The createFile(name, content) method is a DriveApp method that creates a new Google Drive file with the specified name and content.
13. **createFolder(name):** The createFolder(name) method is a DriveApp method that creates a new Google Drive folder with the specified name.

14. `getBlob()`: The `getBlob()` method is a File method that returns the content of a Google Drive file as a Blob object.
15. `getFiles()`: The `getFiles()` method is a DriveApp method that returns an array of all the files in the user's Google Drive.
16. `getFolders()`: The `getFolders()` method is a DriveApp method that returns an array of all the folders in the user's Google Drive.
17. `getFileById(id)`: The `getFileById(id)` method is a DriveApp method that returns the File object associated with the specified file ID.
18. `getFolderById(id)`: The `getFolderById(id)` method is a DriveApp method that returns the Folder object associated with the specified folder ID.
19. ID: An ID is a unique identifier assigned to every Google Drive file and folder. Google Apps Script provides methods for getting and setting the ID of a Google Drive file or folder.
20. `newFile(name, content)`: The `newFile(name, content)` method is a File constructor that creates a new File object with the specified name and content.
21. `newFolder(name)`: The `newFolder(name)` method is a Folder constructor that creates a new Folder object with the specified name.
22. `removeEditor(user)`: The `removeEditor(user)` method is a DriveApp method that revokes edit permissions for a user for a Google Drive file.
23. `removeViewer(user)`: The `removeViewer(user)` method is a DriveApp method that revokes view permissions for a user for a Google Drive file.
24. `setName(name)`: The `setName(name)` method is a File method that sets the name of a Google Drive file.
25. `setDescription(description)`: The `setDescription(description)` method is a File method that sets the description of a Google Drive file.
26. `setMimeType(mimeType)`: The `setMimeType(mimeType)` method is a File method that sets the MIME type of a Google Drive file.
27. `setTrashed(trashed)`: The `setTrashed(trashed)` method is a File method that moves a Google Drive file to the trash or restores it from the trash.

Example:

In this example, we will create a new Google Drive file using Google Apps Script. We will then add a user as an editor and set the name, description, and MIME type of the file.

1. Create a new Google Apps Script project.
2. In the script editor, enter the following code:

```
function createFile() {  
  var name = "My File";  
  var description = "This is my file.";  
  var mimeType = "text/plain";  
  var content = "Hello, world!";  
  
  var file = DriveApp.createFile(name, content, mimeType);  
  file.setDescription(description);  
}
```

```
file.setMimeType(mimeType);  
  
var user = Session.getActiveUser().getEmail();  
file.addEditor(user);  
}
```

3. Save the project.
4. Run the createFile() function.
5. A new Google Drive file will be created with the name "My File" and the content "Hello, world!". The file will be of type "text/plain", and the user who ran the function will be granted edit permissions for the file.

Challenge:

Create a Google Apps Script function that creates a new Google Drive folder and adds a user as a viewer for the folder. The function should prompt the user for the folder name and the email address of the user to add as a viewer.

Solution:

1. Create a new Google Apps Script project.
2. In the script editor, enter the following code:

```
function createFolder() {  
  var name = Prompt.prompt("Enter the folder name:");  
  var userEmail = Prompt.prompt("Enter the email address of the user to add as a viewer:");  
  
  var folder = DriveApp.createFolder(name);  
  var user = Session.getActiveUser().getEmail();  
  folder.addViewer(userEmail);  
}
```

3. Save the project.
4. Run the createFolder() function.
5. Enter the folder name and the email address of the user to add as a viewer.
6. A new Google Drive folder will be created with the specified name, and the specified user will be granted view permissions for the folder.