
Professional Certificate in Google Apps Script Advanced Techniques

Advanced Google Apps Script for Gmail and Calendar

Google Apps Script is a JavaScript-based scripting language developed by Google for light-weight application development in the G Suite platform. Advanced Google Apps Script for Gmail and Calendar is a course that focuses on extending the functionality of Gmail and Google Calendar using Google Apps Script. This explanation covers key terms and vocabulary that learners are expected to understand in this course.

1. Google Apps Script (GAS)

Google Apps Script is a JavaScript-based scripting language that allows developers to automate and extend G Suite applications such as Gmail, Google Sheets, Google Forms, Google Slides, and Google Calendar. GAS is built on top of the JavaScript language and provides additional services, classes, and methods to interact with Google services.

2. Gmail Apps Script Services

Gmail Apps Script services provide methods to interact with Gmail, including sending emails, managing labels, and filtering messages. The Gmail service includes methods for reading and writing messages, managing attachments, and configuring filters. The GmailApp service provides simplified methods for common Gmail tasks, while the Gmail Advanced Service provides more advanced methods for power users.

3. Google Calendar Apps Script Services

Google Calendar Apps Script services provide methods to interact with Google Calendar, including creating and managing events, calendars, and attendees. The Calendar service includes methods for creating and modifying events, managing calendars, and configuring notifications. The CalendarApp service provides simplified methods for common Calendar tasks, while the Calendar Advanced Service provides more advanced methods for power users.

4. Triggers

Triggers are automated events that run scripts in response to a specific event. Triggers can be time-based, such as running a script every hour, or event-based, such as running a script when a new email arrives in Gmail. Triggers can be configured in the Apps Script editor or programmatically using the ScriptApp service.

5. ScriptApp Service

The ScriptApp service provides methods for managing scripts, including creating and deleting scripts, managing triggers, and running scripts programmatically. The ScriptApp service also provides methods for interacting with the Apps Script API, allowing developers to manage scripts and triggers programmatically.

6. Apps Script API

The Apps Script API is a RESTful API that allows developers to manage scripts and triggers programmatically. The Apps Script API can be used to create, delete, and modify scripts, as well as manage triggers and deployments. The Apps Script API can be accessed using OAuth 2.0 authentication and can be used with a variety of programming languages.

7. OAuth 2.0

OAuth 2.0 is an authorization protocol that allows users to grant third-party applications access to their Google services without sharing their passwords. OAuth 2.0 is used to authenticate users and authorize applications to access Google services, including Google Apps Script. OAuth 2.0 uses access tokens to authenticate requests, which are short-lived and can be revoked by the user at any time.

8. Access Tokens

Access tokens are short-lived credentials that are used to authenticate requests to Google services. Access tokens are issued by the Google OAuth 2.0 server and are included in the Authorization header of HTTP requests. Access tokens are valid for a limited time and must be refreshed periodically to maintain access to Google services.

9. Refresh Tokens

Refresh tokens are long-lived credentials that are used to obtain new access tokens. Refresh tokens are issued by the Google OAuth 2.0 server and are stored securely on the client side. Refresh tokens can be used to obtain new access tokens without requiring the user to re-authenticate.

10. Scopes

Scopes are permissions that applications request to access Google services. Scopes define the level of access that an application requires, such as read-only access to Gmail or write access to Google Calendar. Scopes are defined in the Google OAuth 2.0 server and are included in the authentication request.

11. Deployments

Deployments are versions of a script that can be published and shared with others. Deployments provide a way to manage different versions of a script and allow users to test and debug scripts before publishing them. Deployments can be configured to run with different permissions, such as running as the owner or running as the user.

12. Web Hooks

Web hooks are HTTP callbacks that are triggered by specific events. Web hooks can be used to send notifications or trigger scripts when specific events occur, such as when a new email arrives in Gmail or when a new event is created in Google Calendar. Web hooks can be configured using the Google Apps Script editor or programmatically using the ScriptApp service.

13. JSON

JSON (JavaScript Object Notation) is a lightweight data interchange format that is widely used in web development. JSON is a text-based format that is easy to read and write, and can be parsed and generated using JavaScript and many other programming languages. JSON is used to exchange data between Google Apps Script and other applications, such as sending data to a web hook or receiving data from a RESTful API.

14. JDBC

JDBC (Java Database Connectivity) is a Java API for accessing relational databases. JDBC provides a standard interface for connecting to databases, executing SQL queries, and processing results. JDBC can be used with Google Apps Script to store and retrieve data in a relational database, such as Google Cloud SQL or a local MySQL database.

15. Google Sheets

Google Sheets is a spreadsheet application that is part of the G Suite platform. Google Sheets can be used to store and analyze data, and can be accessed programmatically using Google Apps Script. Google Sheets provides a range of methods for reading and writing data, creating charts, and formatting cells.

16. Google Forms

Google Forms is a survey application that is part of the G Suite platform. Google Forms can be used to create surveys, quizzes, and polls, and can be accessed programmatically using Google Apps Script. Google Forms provides a range of methods for creating forms, managing responses, and analyzing data.

17. Google Slides

Google Slides is a presentation application that is part of the G Suite platform. Google Slides can be used to create presentations, and can be accessed programmatically using Google Apps Script. Google Slides provides a range of methods for creating slides, managing layouts, and formatting text.

18. Google Drive

Google Drive is a file storage and synchronization service that is part of the G Suite platform. Google Drive can be used to store and share files, and can be accessed programmatically using Google Apps Script. Google Drive provides a range of methods for creating, reading, and writing files, as well as managing permissions and sharing settings.

19. Google Cloud Platform

Google Cloud Platform is a cloud computing platform that provides a range of services for building, deploying, and managing applications. Google Cloud Platform can be accessed programmatically using Google Apps Script, and provides a range of services for storing data, processing data, and analyzing data.

20. Google App Engine

Google App Engine is a platform as a service (PaaS) that allows developers to build and deploy web applications in the cloud. Google App Engine can be accessed programmatically using Google Apps Script, and provides a range of services for building, deploying, and managing web applications.

Examples and Practical Applications:

- * Use the Gmail service to send automated email reminders to team members.
- * Use the Google Calendar service to create and manage events programmatically.
- * Use triggers to automate repetitive tasks, such as sending daily reports or backing up data.
- * Use the ScriptApp service to manage scripts and triggers programmatically.
- * Use the Apps Script API to manage scripts and triggers using a RESTful API.
- * Use OAuth 2.0 to authenticate users and authorize applications to access Google services.
- * Use access tokens and refresh tokens to maintain access to Google services.
- * Use scopes to request permissions from users and limit access to Google services.
- * Use deployments to manage different versions of a script and allow users to test and debug scripts before publishing them.
- * Use web hooks to send notifications or trigger scripts when specific events occur.
- * Use JSON to exchange data between Google Apps Script and other applications.
- * Use JDBC to store and retrieve data in a relational database.
- * Use Google Sheets to store and analyze data, and create charts and reports.
- * Use Google Forms to create surveys, quizzes, and polls, and manage responses and data.
- * Use Google Slides to create presentations and manage layouts and formatting.
- * Use Google Drive to store and share files, and manage permissions and sharing settings.
- * Use Google Cloud Platform to build, deploy, and manage applications in the cloud.
- * Use Google App Engine to build and deploy web applications in the cloud.

Challenges: