**Introduction to APIs in Google Cloud**

**Overview**

APIs (Application Programming Interfaces) are software programs that give developers access to computing resources and data. Companies from many different fields offer publicly available APIs so that developers can integrate specialized tools, services, or libraries with their own applications and codebase.

This lab reviews the architecture and basic functioning of APIs. This also provides hands-on practice, where you configure and run [Cloud Storage API](https://cloud.google.com/storage/docs/json_api/) methods in Google Cloud Shell. Taking this lab helps you understand the key principles of API communication, architecture, and authentication. You also gain practical experience with APIs, which you can apply to future labs or projects.

APIs - what and why

The ability to access data and computing resources greatly increases a developer's efficiency. It is much easier to use an API than to build every single program, method, or dataset from scratch. APIs are built on the principle of abstraction—you don't need to understand the inner workings or complexities of an API to use it in your own environment.

APIs are built with the developer in mind and often don't offer a graphical user interface (GUI). However, there are exceptions to this standard. Google has released a new tool called [APIs Explorer](https://developers.google.com/apis-explorer/#p/), which allows you to explore various Google APIs interactively (be sure to check out the [APIs Explorer: Qwik Start](https://www.cloudskillsboost.google/catalog_lab/1241) lab afterwards if you are interested in learning more.)

Cloud APIs

Google offers APIs that can be applied to many different fields and sectors. APIs are often used in web development, machine learning, data science, and system administration workflows. However, these are only a handful of use cases. By exploring [AnyAPI](https://any-api.com/" \t "_blank), for example, you see that there are many APIs available.

When Qwiklabs provisions a new Google Cloud Project for a lab instance, it enables most APIs behind the scenes so you can work on the lab's tasks right away. If you create your own projects outside of Qwiklabs, you must enable certain APIs yourself.

As you gain proficiency as a Google Cloud user, you use more APIs in your workflow. Experienced users integrate and use Cloud APIs in their local environments almost exclusively, rarely using the Cloud Console to run tools and services. Dozens of Hands-on Labs are available that give you practice with various Google APIs in different languages. Here are two for example:

* [Cloud Natural Language API: Qwik Start](https://www.cloudskillsboost.google/catalog_lab/709)
* [Entity and Sentiment Analysis with the Natural Language API](https://www.cloudskillsboost.google/catalog_lab/1113)

In this lab, you explore the [API library](https://console.cloud.google.com/apis/library?project=hello-world-sean-200116&folder&organizationId) to see what Google APIs are available.

**Objectives**

In this lab, you learn about:

* Google APIs
* API architecture
* HTTP protocol and methods
* Endpoints
* REST (Representational State Transfer) and RESTful APIs
* JSON (JavaScript Object Notation)
* API authentication services

**Prerequisites**

This is an **introductory level** lab. This assumes little to no prior knowledge of APIs or experience using Google APIs. Familiarity with shell environments and command line interface tools is recommended, but not required. Familiarity with the Cloud Console and Cloud Storage is recommended, so please at a minimum take the following labs before attempting this one:

* [A Tour of Qwiklabs and Google Cloud](https://www.cloudskillsboost.google/catalog_lab/1281)
* [Cloud Storage: Qwik Start - Cloud Console](https://www.cloudskillsboost.google/catalog_lab/1089)

Once you're ready, scroll down and follow the steps below to set up your lab environment.

**How do APIs work?**

**API architecture**

APIs (Application Programming Interfaces) are a set of methods that allow programs to communicate with each other. To communicate effectively, programs must adhere to a clear protocol that governs the transfer and interpretation of data. The internet is the standard communication channel that APIs use to transmit requests and responses between programs. Web-based APIs use the client-server model as the underlying architecture for exchanging information. The client is a computing device that makes a request for some computing resource or data, and the server has data and/or computing resources stored on it, which interprets and fulfills the client's request.

**HTTP protocol and methods**

Since APIs use the web as a communication channel, many of them adhere to the HTTP protocol, which specifies rules and methods for data exchange between clients and servers over the internet. APIs that utilize the HTTP protocol use HTTP request methods (also known as "HTTP verbs") for transmitting client requests to servers. The most commonly used HTTP request methods are GET, POST, PUT, and DELETE. GET is used by a client to fetch data from a server, PUT replaces existing data or creates data if it does not exist, POST is used primarily to create new resources, and DELETE removes data or resources specified by the client on a server.

**Endpoints**

APIs use HTTP methods to interact with data or computing services hosted on a server. These methods are useless if there isn't a way to access specific resources with consistency. APIs utilize communication channels called endpoints so that clients can access the resources they need without complication or irregularity. Endpoints are access points to data or computing resources hosted on a server and they take the form of an HTTP URI. Endpoints are added to an API's base URL to create a path to a specific resource or container of resources. Additionally, query strings can be added to endpoints to pass in variables that may be needed to complete an API's request.

**REST and RESTful APIs**

APIs that utilize the HTTP protocol, request methods, and endpoints are referred to as RESTful APIs. RESTful APIs live on the server, acting as an implementer for client requests. This model defines a framework of endpoints (nouns) that HTTP methods (verbs) act on, and APIs use this framework to fulfill requests. To summarize, RESTful APIs utilize the client-server model, adhere to the HTTP protocol, utilize HTTP request methods, and utilize endpoints to access specific resources.

Authentication and authorization

**Authentication** and **authorization** are two terms that are often used interchangeably, but they are not the same thing.

* *Authentication* refers to the process of determining a client's identity.
* *Authorization* refers to the process of determining what permissions an authenticated client has for a set of resources.

Authentication identifies who you are, and authorization determines what you can do.

There are three types of authentication/authorization services that Google APIs use. These are "API Keys", "Service accounts", and "OAuth". An API uses one of these authentication services depending on the resources it requests and from where the API is called from.

API keys

**API keys** are secret tokens that usually come in the form of an encrypted string. API keys are quick to generate and use. Oftentimes, APIs that use public data or methods and want to get developers up and running use API keys to quickly authenticate users.

In Google Cloud terms, API keys identify the calling project making the call to an API. By identifying the calling project, API keys enable usage information to be associated with that project, and they can reject calls from projects that haven't been granted access or enabled by the API.

OAuth

**OAuth** tokens are similar to API keys in their format, but they are more secure and can be linked to user accounts or identities. These tokens are used primarily when APIs give a developer the means to access user data.

While API keys give developers access to all of an API's functionality, OAuth client IDs are all based on scope; different privileges are granted to different identities.

Service Accounts

A **service account** is a special type of Google account that belongs to your application or a virtual machine (VM) instead of to an individual end user. Your application assumes the identity of the service account to call Google APIs, so that the users aren't directly involved.

You can use a service account by providing its private key to your application, or by using the built-in service accounts available when running on Cloud Functions, Google App Engine, Compute Engine, or Google Kubernetes Engine.