For almost all our supported languages, you can also download a single client library that provides an interface to all supported Cloud APIs. You can find links to get started with these and their reference documentation on our [Cloud Client Libraries page](https://cloud.google.com/apis/docs/cloud-client-libraries).

Firebase is the Google-wide solution for building applications on mobile devices. It offers an SDK with client code that lets you access mobile-relevant Cloud APIs directly from iOS, Android, and Web apps. Visit the [Firebase documentation](https://firebase.google.com/docs) for more information on which Cloud APIs are supported and how to get started with Firebase.

REST/HTTP APIs

All Cloud APIs expose a simple traditional JSON/REST interface. If you need to write your own custom code to directly access the REST API using a third-party HTTP client library of your choice, you can find out more about how Cloud APIs work with different HTTP versions and implementations in our [HTTP Guidelines](https://cloud.google.com/apis/docs/http).

Service accounts are managed by [Cloud IAM](https://cloud.google.com/iam/docs/understanding-service-accounts), and they represent non-human users. They are intended for scenarios where your application needs to access resources or perform actions on its own, such as running App Engine apps or interacting with Compute Engine instances.

See [Cloud IAM Overview](https://cloud.google.com/iam/docs/overview) for more information about each account type

To build an application using GCP APIs, follow these general steps:

* Choose and use the provided Cloud Client Libraries
* Determine the correct authentication flow for your application
* Find or create the application credentials needed for your application
* Pass the application credentials to the client libraries at application startup time, ideally through [Application Default Credentials](https://cloud.google.com/docs/authentication/production#finding_credentials_automatically) (ADC)

Accessing private data on behalf of a service account outside GCP environments Service account key You need to create a service account, and download its private key as a JSON file. You need to pass the file to Cloud Client Libraries, so they can generate the service account credentials at runtime.

Cloud Client Libraries will automatically find and use the service account credentials by using the GOOGLE\_APPLICATION\_CREDENTIALS environment variable.

Enabling an API requires you to accept the Terms of Service and billing responsibility for the API. You need permission serviceusage.services.enable on the project and permission servicemanagement.services.bind on the API to enable it. For more information, see Service Usage [Access Control](https://cloud.google.com/service-usage/docs/access-control).