



Google

Developers

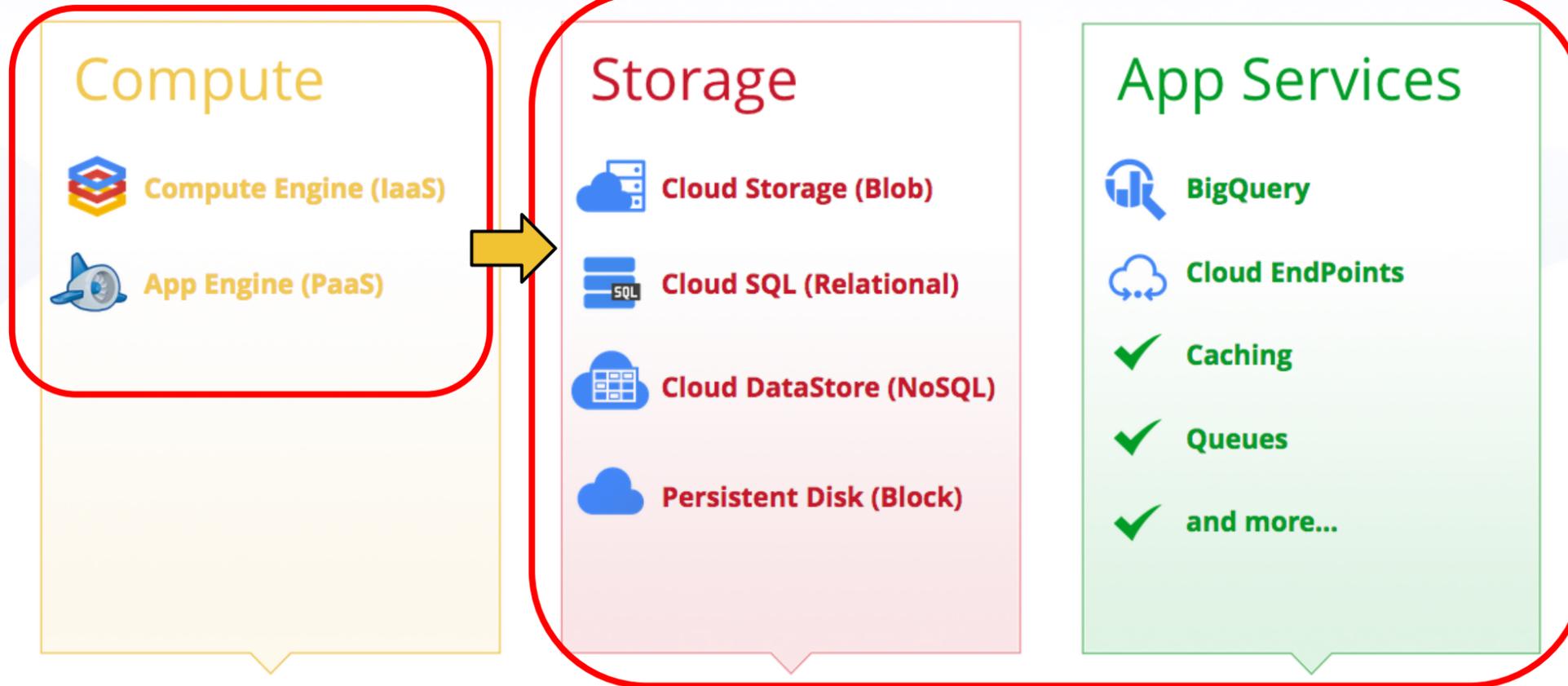


Keys to the Kingdom: OAuth in Google Cloud Platform

Adam Eijdenberg
Product Manager
eijdenberg@google.com

Ben Sittler
Software Engineer
bsittler@google.com





Google Infrastructure

- ✓ Global Data Centers
- ✓ 99.95% Uptime SLA
- ✓ Performance
- ✓ Redundancy
- ✓ Disaster Recovery
- ✓ Audits & Certifications
- ✓ Security
- ✓ Energy Efficient



What are we covering today?

- **Authentication**

- From first principles, using Google Client Libraries
- From anywhere, Google App Engine, Google Compute Engine

- **Authorization**





Let's make an API call

Analyze terabytes of data with just a click of a button

Use Google BigQuery to interactively analyze massive datasets — up to billions of rows.

Compose Query ? ×

```
SELECT timestamp, title, COUNT(*) as cnt
FROM publicdata:samples.wikipedia
WHERE LOWER(title) CONTAINS 'speed' AND wp_namespace = 0
GROUP BY title, timestamp ORDER BY cnt DESC LIMIT 20;
```

RUN QUERY Query complete (4.1s elapsed, 11.5 GB processed)

Query Results

[Download as CSV](#) [Save as Table](#)

Row	timestamp	title	cnt
1	1196276720	New Hampshire Motor Speedway	2
2	1187028345	Speedway World Team Cup	2
3	1043861144	Speed of gravity	2

What is Google BigQuery?

Google BigQuery is a web service that lets you do interactive analysis of massive datasets—up to billions of rows. Scalable and easy to use, BigQuery lets developers and businesses tap into powerful data analytics on demand.

[Learn More](#)

Sign Up for BigQuery

Ready to try out BigQuery? BigQuery is now available to the public! Sign up for the service now, or [learn more](#).

Need enterprise level support? [Contact a sales representative](#).

[Sign Up](#)



All I want to do is make an API call!

- What is the minimum I need to do to make a successful API call against a Google API?

```
GET /bigquery/v2/projects/1234/datasets HTTP/1.1
```

```
Host: www.googleapis.com
```

```
Authorization: Bearer yaXXXXXXXXXX
```

- You need a token - This presentation will show you how to get one - painlessly!



What is a token?

- Called an "Access Token" - identifies 3 things to Google:
 1. The application (called **Client**) making the request
 2. The **User** (if applicable) authorizing the request
 3. What class of actions the user has authorized the application to perform (called **Scope**)



How do I get one?

- This depends on *whose* data is being accessed
- Data owned by a Google **user**
 - e.g. show consent screen for your application to access data on behalf of a user, e.g. Google Drive, Google Calendar
- Data owned by my **application** or provided by Google
 - e.g. Google Cloud Storage, Google BigQuery API, Google Compute Engine
- This presentation will focus on accessing data belonging to your **application** - typical pattern for Cloud Platform APIs



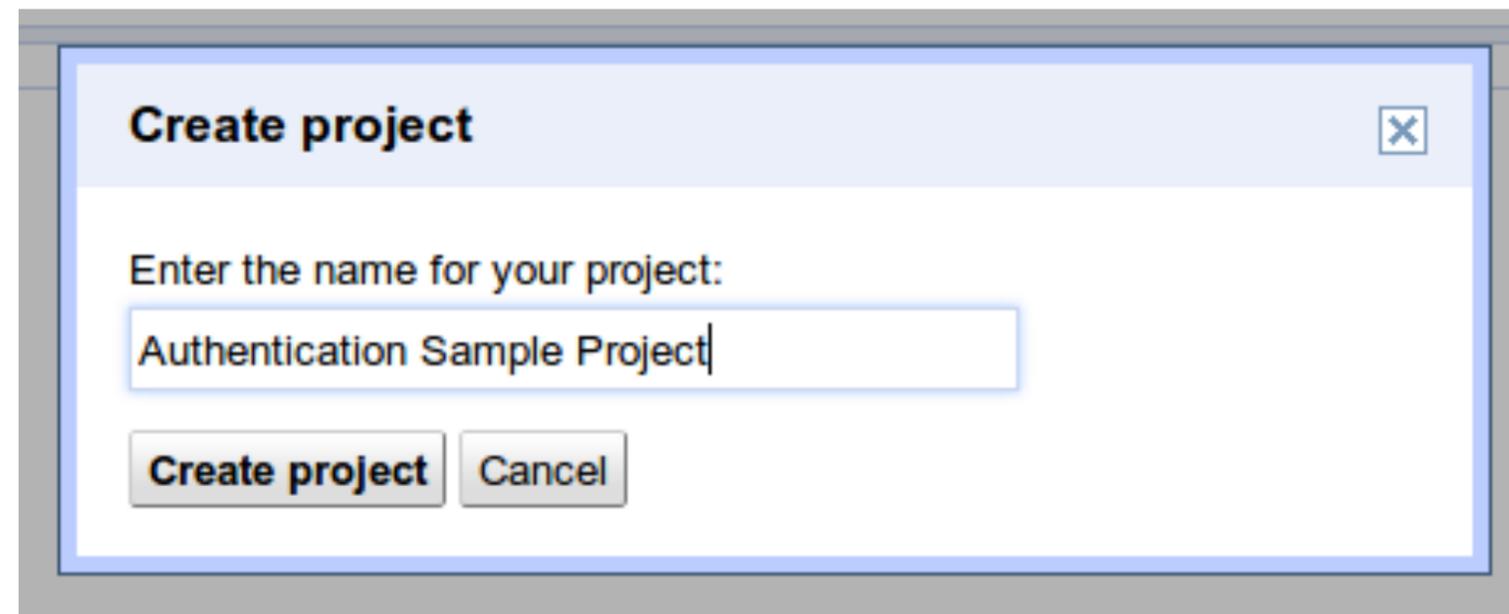


Configuring the project

Accessing data owned by my application

- Let's get started with BigQuery

<https://developers.google.com/console/>



The image shows a screenshot of a 'Create project' dialog box. The dialog has a title bar with the text 'Create project' and a close button (X) in the top right corner. Below the title bar, there is a text prompt: 'Enter the name for your project:'. Underneath this prompt is a text input field containing the text 'Authentication Sample Project'. At the bottom of the dialog, there are two buttons: 'Create project' and 'Cancel'.



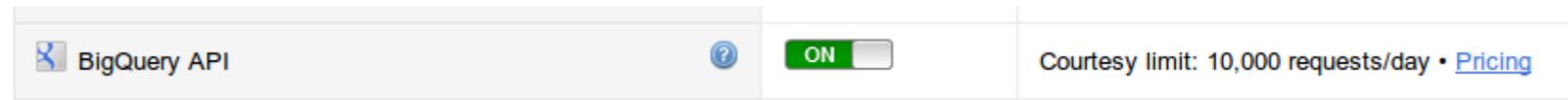
Project

- Container for:
 - Resources
 - Google Cloud Storage buckets
 - Google Compute Engine virtual machines
 - BigQuery datasets
 - Registered applications and credentials
- Holds configuration for:
 - Authorization
 - Billing / quota management
 - Services enabled



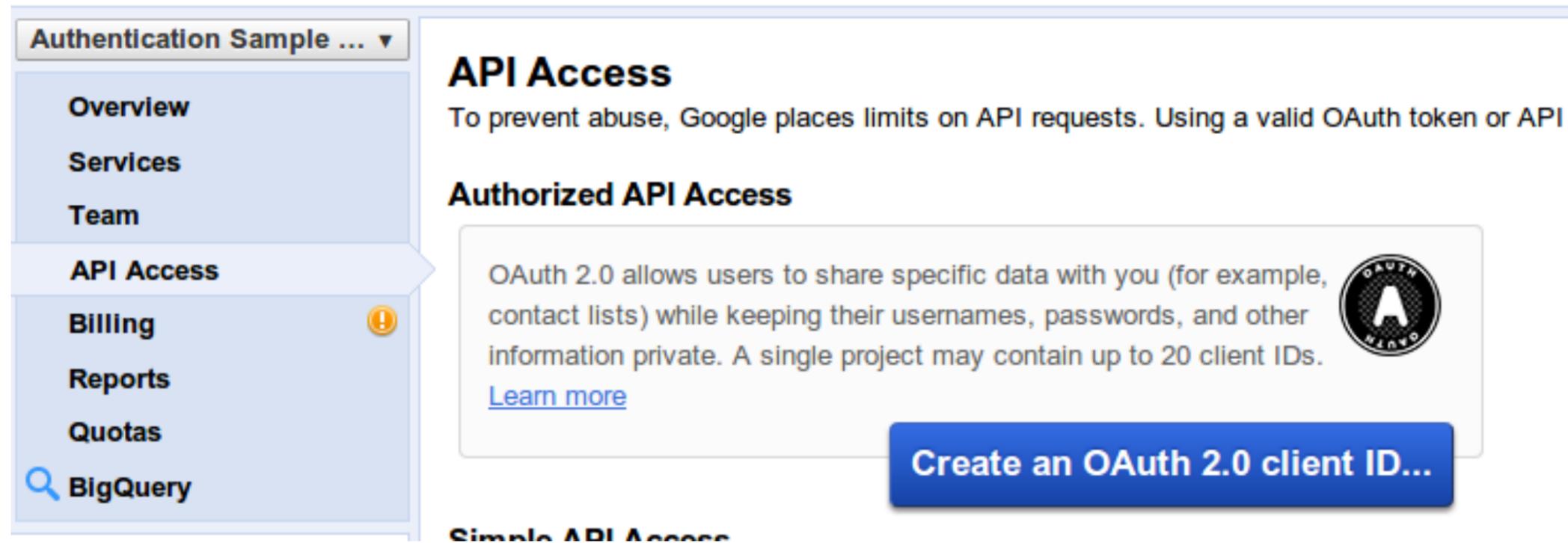
Enable the service

- Remember the access token identifies the application making the request?
- The API you are calling must be enabled within the project where the application is registered (next step)



Create a service account

- A service account is a non-human user
- Allows your application to make API calls on behalf of itself
- Identity for your application - like a role account

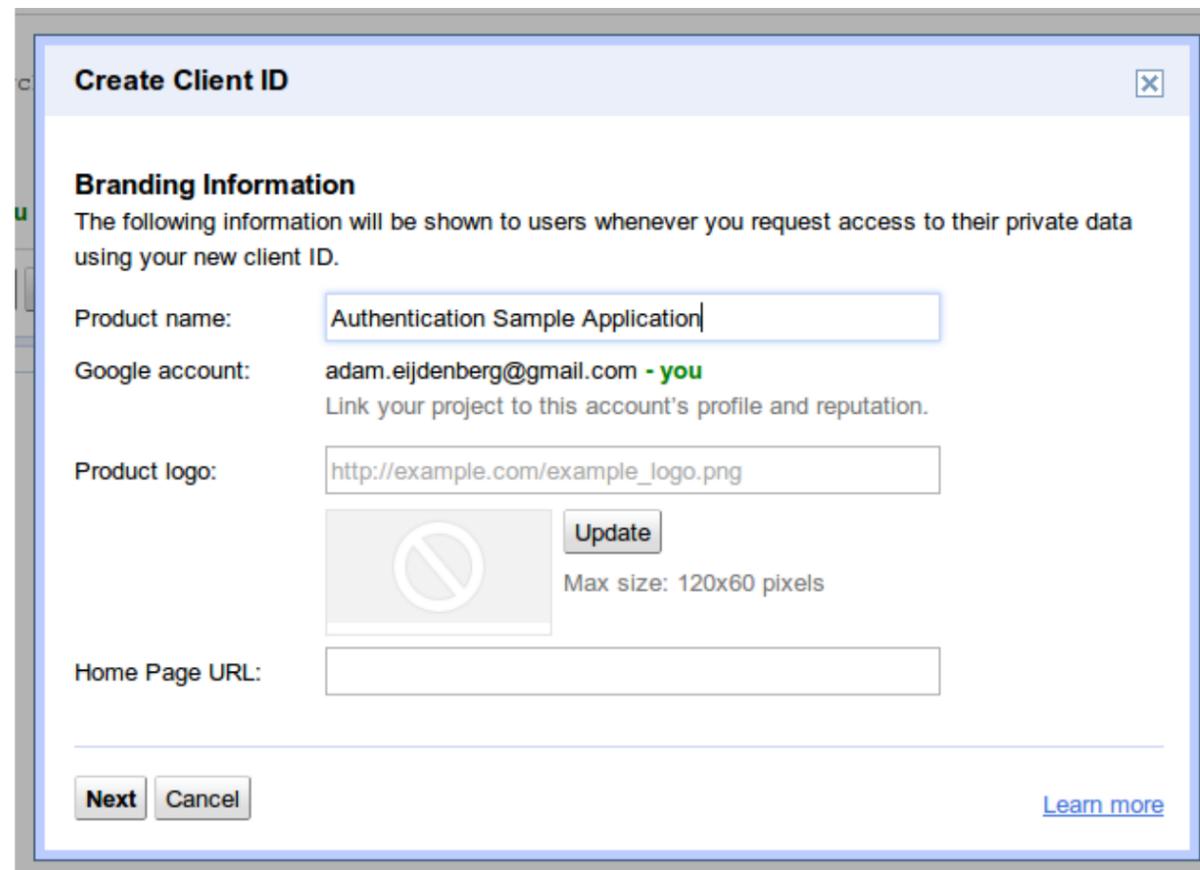


The screenshot shows the Google Cloud console interface for a project named 'Authentication Sample ...'. The left-hand navigation menu includes 'Overview', 'Services', 'Team', 'API Access' (which is highlighted), 'Billing' (with a warning icon), 'Reports', 'Quotas', and 'BigQuery'. The main content area is titled 'API Access' and contains the following text: 'To prevent abuse, Google places limits on API requests. Using a valid OAuth token or API'. Below this is a section titled 'Authorized API Access' with a text box explaining that 'OAuth 2.0 allows users to share specific data with you (for example, contact lists) while keeping their usernames, passwords, and other information private. A single project may contain up to 20 client IDs.' A 'Learn more' link is provided. To the right of the text is an OAuth logo. At the bottom right of the main content area is a prominent blue button labeled 'Create an OAuth 2.0 client ID...'. The text 'Simple API Access' is partially visible at the bottom of the page.



Branding Information screen

- The first time a Client is created you will be prompted for the following - this is not used for application authentication.



Create Client ID

Branding Information
The following information will be shown to users whenever you request access to their private data using your new client ID.

Product name:

Google account: **adam.eijdenberg@gmail.com - you**
Link your project to this account's profile and reputation.

Product logo:

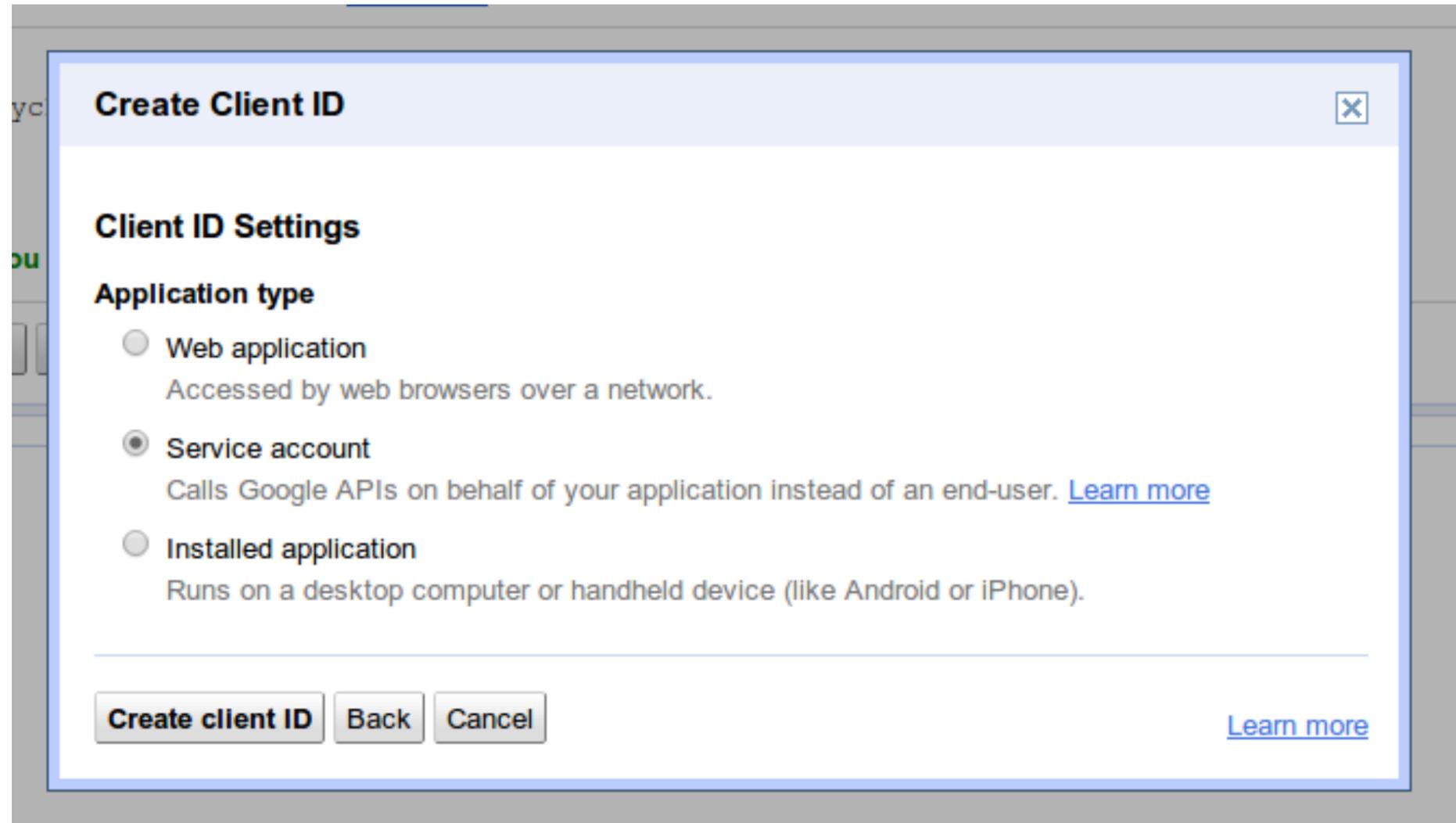

Max size: 120x60 pixels

Home Page URL:

[Learn more](#)



Select "Service Account"



Create Client ID [Close]

Client ID Settings

Application type

- Web application
Accessed by web browsers over a network.
- Service account
Calls Google APIs on behalf of your application instead of an end-user. [Learn more](#)
- Installed application
Runs on a desktop computer or handheld device (like Android or iPhone).

[Learn more](#)

Create client ID **Back** **Cancel**



Download key



Take note of the email address

Service account

Use service accounts to call Google APIs on behalf of your application instead of an end-user. [Learn more](#)

Client ID:	587080245102.apps.googleusercontent.com
Email address:	587080245102@developer.gserviceaccount.com
Public key fingerprints:	b4becc0154b4511aa1916a6e3008b45c401f1a84 – Delete...

Create another client ID...





Let's get an access token

Create an assertion

```
import time, json, base64

now = long(time.time())
assertion_input = "%s.%s" % (
    base64.urlsafe_b64encode(
        json.dumps({"alg": "RS256", "typ": "JWT"}).encode("UTF-8"))
    ).rstrip("="),
    base64.urlsafe_b64encode(json.dumps({
        "iss": "yyyyyyy@developer.gserviceaccount.com",
        "scope": "https://www.googleapis.com/auth/bigquery",
        "aud": "https://accounts.google.com/o/oauth2/token",
        "exp": now + (60 * 60), "iat": now
    }).encode("UTF-8")).rstrip("="))
```



Sign it - and swap it

```
import urllib2, urllib
from OpenSSL import crypto
```

```
access_token = json.loads(urllib2.urlopen("https://accounts.google.com/o/oauth2/token",
    urllib.urlencode({
        "grant_type": "urn:ietf:params:oauth:grant-type:jwt-bearer",
        "assertion": "%s.%s" % (assertion_input,
            base64.urlsafe_b64encode(crypto.sign(crypto.load_pkcs12(
                file("xxxxxxxx-privatekey.p12", "rb").read(),
                "notasecret").get_privatekey(), assertion_input, "sha256"))).rstrip("="))
    })).read()["access_token"]
```



Result

```
{  
  "access_token" : "yaXXXXXXXXXXXXXXXXXXXXXXXXXXXX",  
  "token_type" : "Bearer",  
  "expires_in" : 3600  
}
```



Try using it

```
print urllib2.urlopen(urllib2.Request(  
    "https://www.googleapis.com/bigquery/v2/projects/%s/datasets"  
    % "my-test-project-id",  
    headers={'Authorization': 'Bearer %s' % access_token}  
)).read()
```

Result:

```
{  
    "kind": "bigquery#datasetList",  
    "etag": "..."  
}
```



Using Google Client libraries

```
import httplib2
from oauth2client.client import SignedJwtAssertionCredentials

http = SignedJwtAssertionCredentials(
    "yyyyyyyyyyyy@developer.gserviceaccount.com",
    file("xxxxxxxxxx-privatekey.p12", "rb").read(),
    scope="https://www.googleapis.com/auth/bigquery"
).authorize(httplib2.Http())
```



Use authorized HTTP object directly

```
print http.request("https://www.googleapis.com/bigquery/v2/projects/%s/datasets"  
    % "my-test-project-id") [1]
```

Result:

```
{  
  "kind": "bigquery#datasetList",  
  "etag": "..."  
}
```



Or the full client library stack

```
from apiclient.discovery import build

service = build('bigquery', 'v2')
print service.datasets().list(projectId="my-test-project-id").execute(http)
```

Result:

```
{
  "kind": "bigquery#datasetList",
  "etag": "..."}
}
```





Google hosted environments

Google hosted environments?

- Built-in service account support provided for:
 - Google Compute Engine
 - Google App Engine
- You need to add the Google App Engine service account email address to the team for the project that contains the data you wish to access.
- For some APIs you will also need to pass an API key in the request



Use built-in identity in Google App Engine

```
# Built-in:  
from google.appengine.api import app_identity  
  
access_token, ttl = app_identity.get_access_token(  
    "https://www.googleapis.com/auth/bigquery")  
  
# With client library:  
from oauth2client.appengine import AppAssertionCredentials  
  
http = AppAssertionCredentials(  
    "https://www.googleapis.com/auth/bigquery").authorize(httplib2.Http())
```



Use built-in identity in Google Compute Engine

```
# When starting your VM - authorize the VM to use the scope you need:  
gcutil --project=my-test-project-id addinstance foobar \  
  --service_account_scopes=https://www.googleapis.com/auth/bigquery
```

```
# Inside the VM:
```

```
curl "http://metadata/computeMetadata/v1beta1/instance/service-accounts/default/token"
```

```
# Result:
```

```
{  
  "access_token" : "yaXXXXXXXXXXXXXXXXXXXXXXXXXXXX",  
  "token_type" : "Bearer",  
  "expires_in" : 3600  
}
```



Use client library on Google Compute Engine

```
from oauth2client.gce import AppAssertionCredentials

http = AppAssertionCredentials(
    "https://www.googleapis.com/auth/bigquery").authorize(httplib2.Http())
```



Authentication - summary

- Your applications need an identity (like a role account) when they access Google APIs
- When running in Google App Engine or Google Compute Engine use built-in credentials
- When running elsewhere (including local Google App Engine development) create a service account and download a private key



What is a token?

- Called an "Access Token" - identifies 3 things to Google:
 1. The application (called **Client**) making the request
 2. The **User** (if applicable) authorizing the request
 3. What class of actions the user has authorized the application to perform (called **Scope**)



Authorization is based on the user

- Ensure the email address associated with the user is added as a "Team" member on the project associated with the resource being accessed
- Correct scope associated with access token

The screenshot shows the Google Cloud IAM & Admin console interface. On the left is a navigation sidebar with a dropdown menu labeled 'Authentication Sample ...'. The sidebar contains the following menu items: Overview, Services, Team (highlighted), API Access, Billing (with a warning icon), Reports, Quotas, and BigQuery. The main content area is divided into two sections: 'Team' and 'Service Accounts'.

Team
You can collaborate on this project with as many or as few people as you like.

Permissions

adam.eijdenberg@gmail.com - you	Is owner ▼	Active	×
adamgaejavatest.google.com@appspot.gserviceaccount.com	Can edit ▼	Active	×

Add a teammate:

Can view ▼ Add

Service Accounts
A service account is a special type of Google Account that is used to provide identity to software. Each service account allows your application to call Google APIs and perform work on behalf of the application itself. Like a teammate, a service account's privileges can be controlled. [Learn more](#)

Permissions

587080245102@developer.gserviceaccount.com	Can edit ▼	Active
--	------------	--------



Billing/quota based on client

- Validate against the project associated with client ID:
 - Service is enabled
 - Billing state is active (where applicable)
 - Quota limit (for API call) has not been exceeded
 - Per project (e.g. courtesy limit)
 - Per user - configurable by developers
- For APIs that include accessing resources belonging to a project (e.g. most Cloud Platform APIs), the above is also validated against the project owning that resource (if different)



Summary

- Access token must be presented
 - Recommend client libraries to acquire token
- Authorize user by adding to appropriate Team
- Now, go and use our APIs to change the world!
 - (or at least solve your problems!)
- Ask questions if you are stuck!
 - [StackOverflow \(google-oauth\)](#)



Thank You!

Adam Eijdenberg - eijdenberg@google.com

Ben Sittler - bsittler@google.com



Google Cloud Platform Resources

cloud.google.com - get started today

developers.google.com - docs and developer guidance

cloud.google.com/newsletter - stay up to date

googlecloudplatform.blogspot.com - our blog

<https://developers.google.com/api-client-library/> - client libraries

<https://developers.google.com/accounts/docs/OAuth2ServiceAccount> - protocol details

Get questions answered on [StackOverflow \(google-oauth\)](https://stackoverflow.com/questions/tagged/google-oauth)





Google
Developers