Google Developers

Google Drive API

Deep Dive

Dr. Ali Afshar, M.D. - Developer Programs Engineer, Google Drive SDK
Ivan Lee - Software Engineer, Google Drive SDK

Google 12 I/O
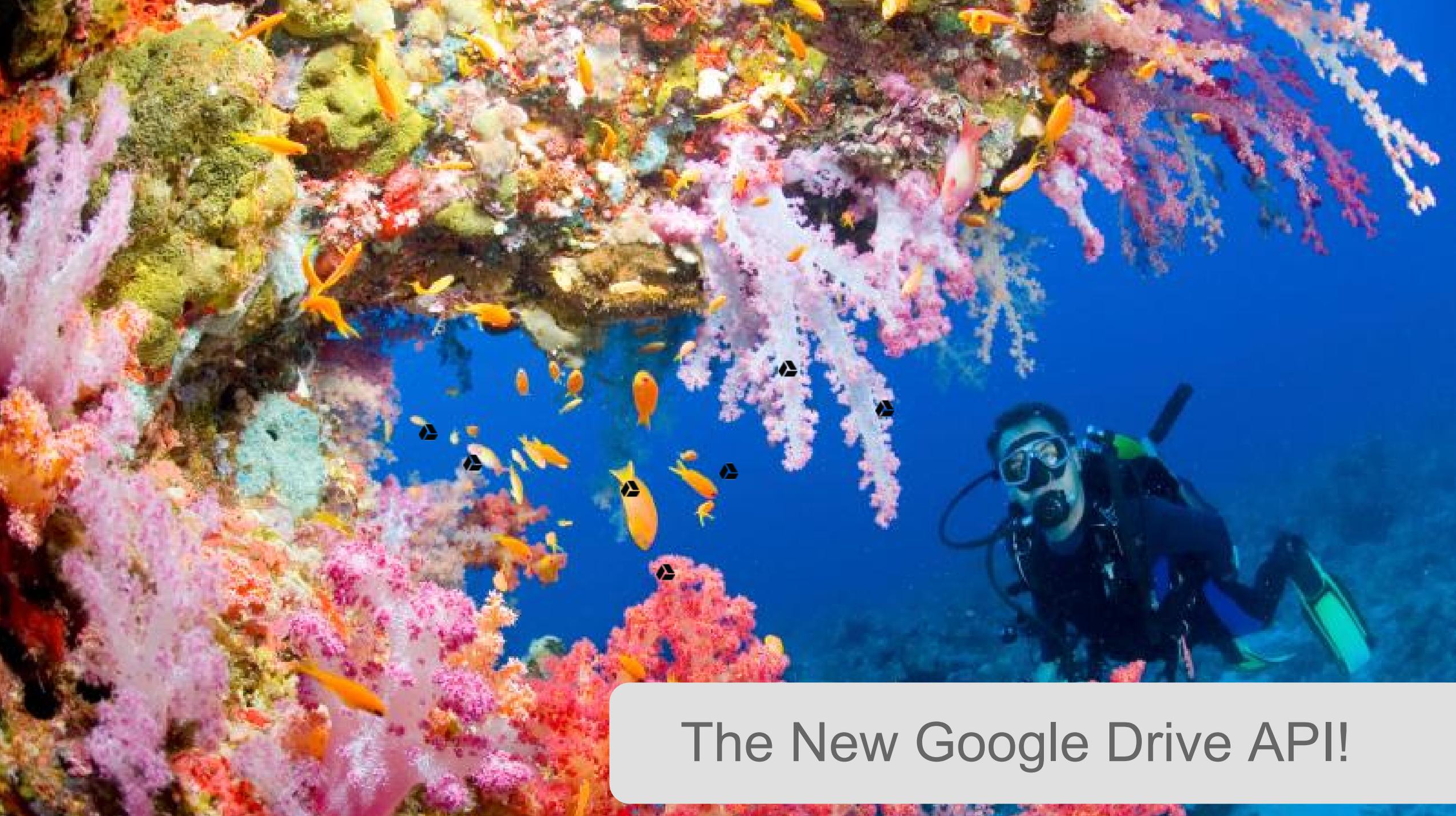
The New Google Drive API!

Google Drive

Access everywhere
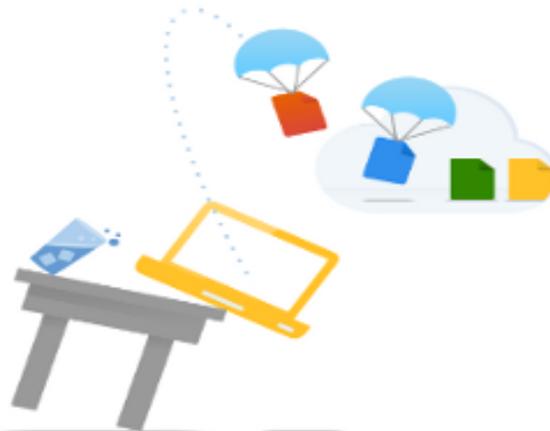
Store files in a safe place

Bring your files to life

Google Drive SDK
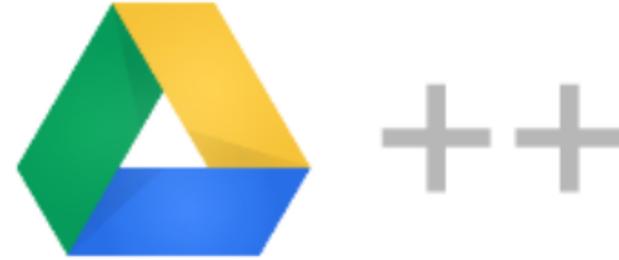
Extensive Reach

Put your app in front of xillions of users with zillions of files

Effortless Integration

Get the best of Google Drive's sharing capabilities, storage capacity and user identity management so you can focus on your app

Google Drive API
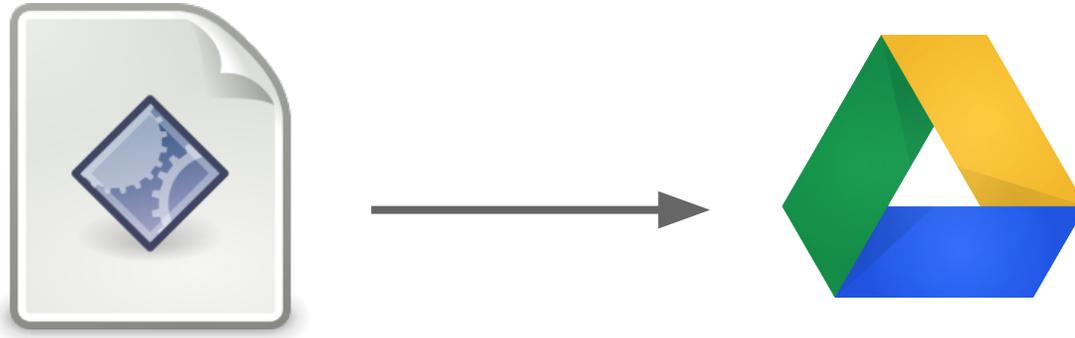
Programmatically manage a user's Google Drive

Features of the Drive API

Replacing the Documents List API

- Mature, widely-used and well-tested technology

- Improved usability

- Standardized client libraries with increased support

- New scopes for varying levels of access

Authorization Scopes

- Full access

- Read-only access with downloads

- Read-only access without downloads

- Per app, per file

- Anonymous

JSON | HTTPS | OAuth 2.0

or

use a Client library

Java	.NET	Objective-C	Python
Ruby	PHP	Javascript	Go

Files, Files, Files!

Everything is a file

Folders are files

What is a file?

- Metadata: title, description, etc.

- Some files have binary data or can be exported as binary data

- Top level drive is the root folder

What's in a File?

```json
{ "kind": "drive#file",
  "id": "0Ak4pWYC-9qb5cGs1T3hoMmVXLUxSdTdvMWt6ZUVNUGc",
  "title": "Feeding plan",
  "mimeType": "application/vnd.google-apps.spreadsheet",
  "labels": { ... },
  "createdDate": "2008-10-13T13:06:03.186Z",
  "modifiedDate": "2012-06-06T16:06:32.795Z",
  "modifiedByMeDate": "2012-03-01T03:26:13.165Z",
  "lastViewedDate": "2012-03-01T03:26:24.590Z",
  "exportLinks": { ... },
  "userPermission": { ... },
  "quotaBytesUsed": "0",
  "editable": true, ... }
```

JSON

JSON is fun and all, but what about some context?

Photo Album Walkthrough

Photo Album

- Key features needed
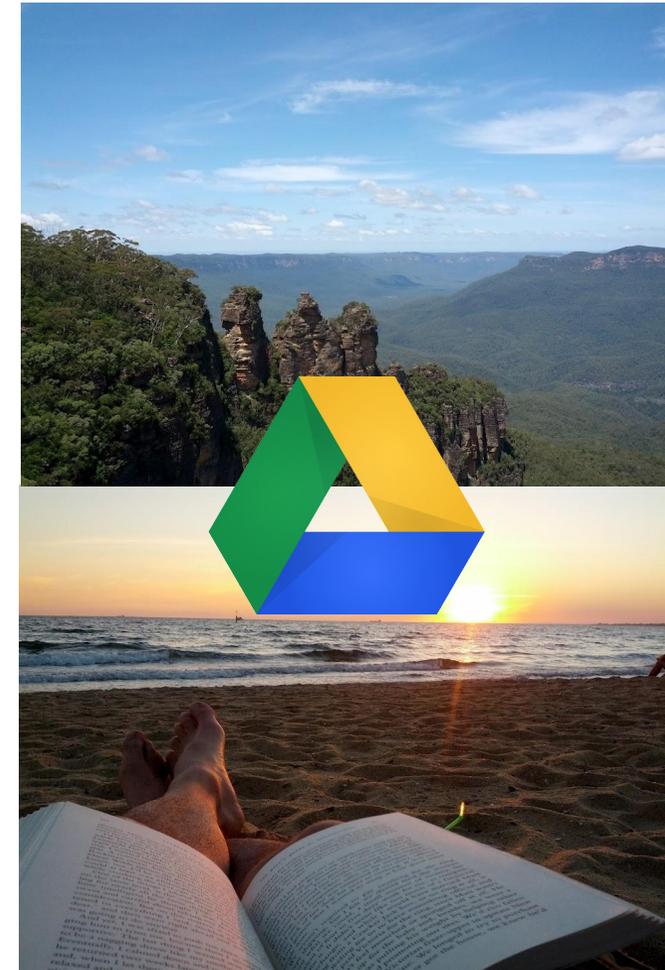
Why Do Spicy Peppers Like Uploads So Much?

So they can get jalapeño Drive!

Uploading a Photo

- ○ Simple

- ○ Multipart

- ○ Resumable

- ● Client libraries wrap upload logic for you

Processing on Upload

Optical Character Recognition (OCR)

`?ocr=true&ocrLanguage=fr`   **HTTP**

Google Docs Conversion

`?convert=true`   **HTTP**

Translation

`?sourceLanguage=fr&targetLanguage=en`   **HTTP**

Simple Upload

- 2 MB size limit

- No metadata

Request

```
POST /upload/drive/v2/files?uploadType=media      HTTP
Content-Type: image/png

<binary png data>
```

Response

```
201 Created                                       HTTP
```

Multipart Upload

Request

```
POST /upload/drive/v2/files?uploadType=multipart
Content-Length: 44332
Content-Type: multipart/related; boundary="
================0124852474844509178=="


================0124852474844509178==
Content-Type: application/json

{'title': 'Funny Photo'}
================0124852474844509178==
Content-Type: image/png

<binary png data>
================0124852474844509178==
```

HTTP

Resumable Upload

Initiate upload
with metadata

<Session URL>

mypicture.jpg

Upload File

<Session URL>

201 Created
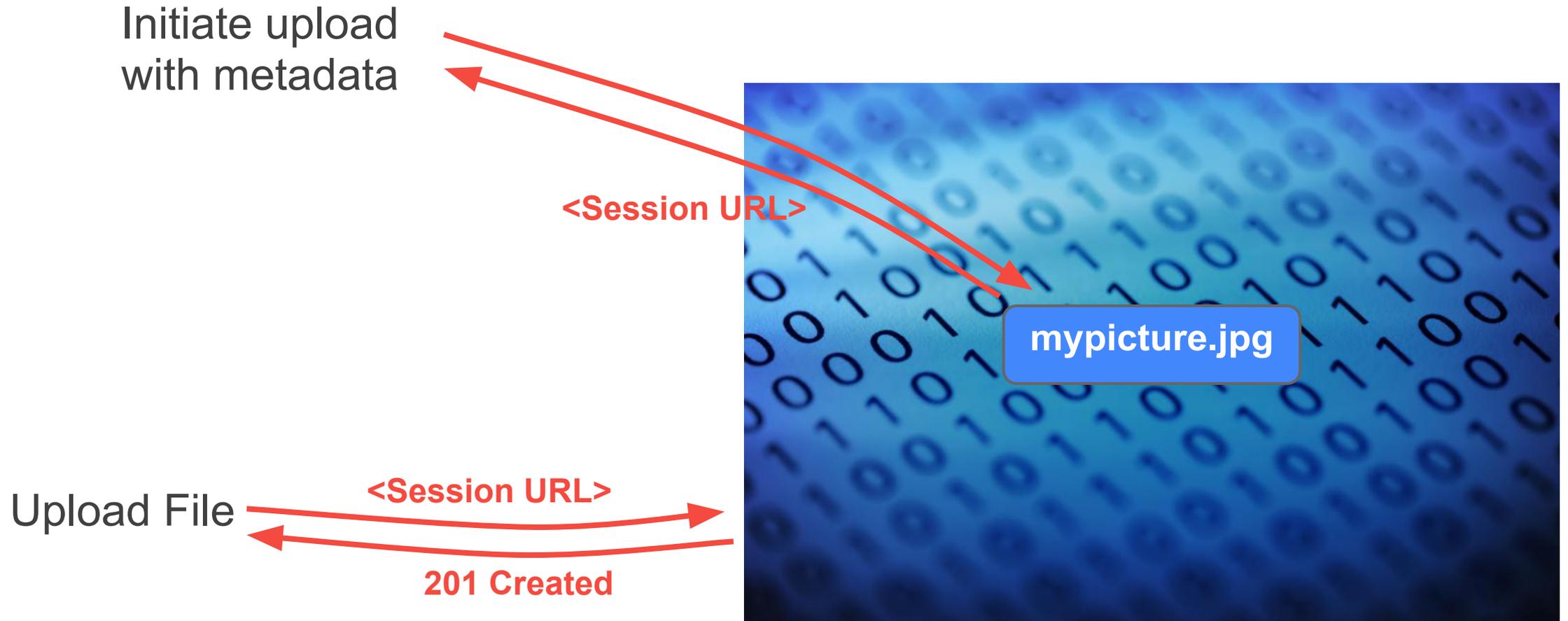
Resumable Upload (failure)

Request Progress

<Session URL>

308 Resume Incomplete

Upload Remaining Bytes

<Session URL>

201 Created
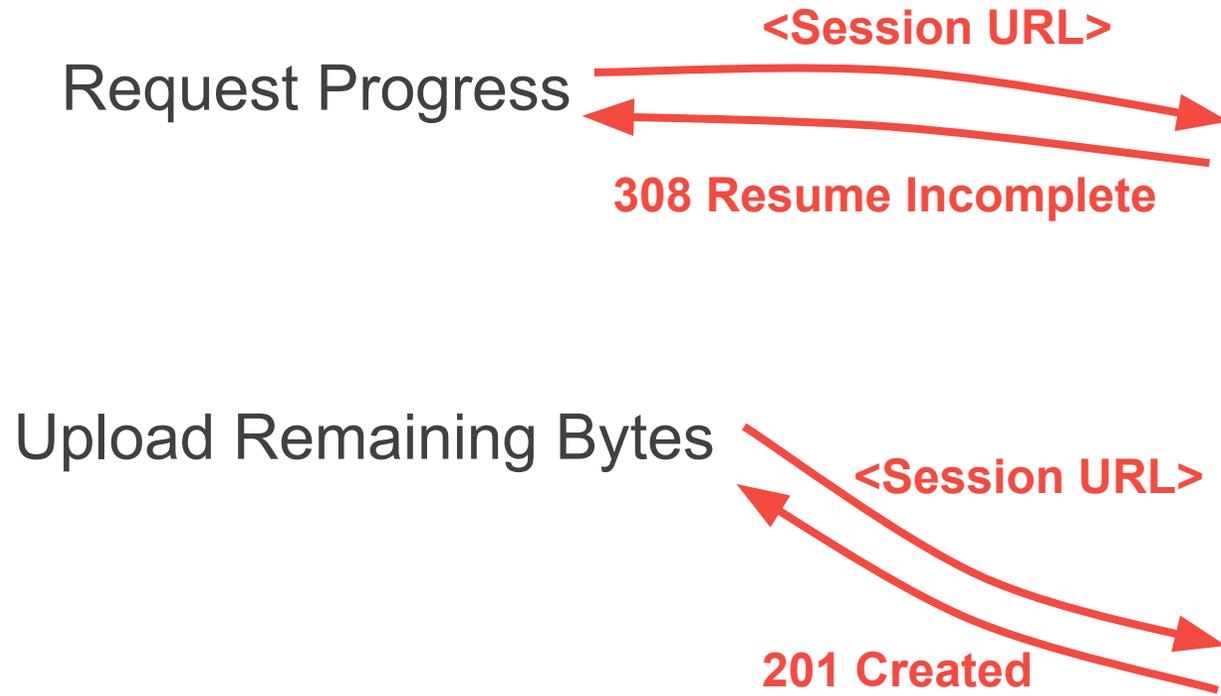
Starting the Upload Session

Request

```
POST /upload/drive/v2/files?uploadType=resumable                    HTTP
Content-Type: application/json
Content-Length: 24
X-Upload-Content-Type: image/png
X-Upload-Content-Length: 10737418

{ "title": "Happy Picture" }
```

Response

```
200 OK                                                              HTTP
Location: <Session URL>
```

Uploading the Bytes

Request to send full file

```
PUT <Session URL>
Content-Length: 10737418
Content-Type: image/png
Content-Range: bytes 0-10737417/10737418

<binary png data>
```

Response on success

```
201 Created
```

Resuming Uploads After Failure

Request to check how much data has been uploaded

```
PUT <Session URL>                                        HTTP
Content-Range: /
```

Response

```
308 Resume Incomplete                                    HTTP
Range: 0-524287
```

Continue from the next byte

```
PUT <Session URL>                                        HTTP
Content-Range: bytes 524288-10737417/10737418

<binary png data>
```

Or Just Use the Client Library

Request

```java
import com.google.api.client.http.FileContent;
import com.google.api.services.drive.model.File;

// Set the metadata
File body = new File();
file.setTitle("Happy Picture");

// Set the file content
java.io.File file = new java.io.File("/home/ivan/image.jpg");
FileContent mediaContent = new FileContent("image/jpeg", file);

// Make the request
service.files().insert(body, mediaContent).execute();
```

Upload
Download
Manage revisions
Copy
Delete
Organize
Share
Search
Changes

Give Me My Photo Back!

Download a File

```json
{ "kind": "drive#file",
  "id": "0Ak4pWYC-9qb5cGs1T3hoMmVXLUxSdTdvMWt6ZUVNUGc",
  "title": "Crazy Picture",
  "mimeType": "image/jpeg",
  "labels": { ... },
  "createdDate": "2008-10-13T13:06:03.186Z",
  "modifiedDate": "2012-06-06T16:06:32.795Z",
  "modifiedByMeDate": "2012-03-01T03:26:13.165Z",
  "lastViewedDate": "2012-03-01T03:26:24.590Z",
  "downloadUrl": "https://...",
  "userPermission": { ... },
  "quotaBytesUsed": "10738294",
  "editable": true, ... }
```

JSON

Downloading Files

Request

```
GET <downloadUrl>                                    HTTP
Authorization: Bearer <token>
```

Response

```
200 OK                                               JSON
<binary data>
```

Resumable Download

Request

```
GET <downloadUrl>
Content-Range: 524288-1048575
```
HTTP

Response

```
200 OK
<binary data>
```
JSON

Restricting Downloads

```json
{ "kind": "drive#file",
  "id": "0Ak4pWYC-9qb5cGs1T3hoMmVXLUxSdTdvMWt6ZUVNUGc",
  "title": "Crazy Picture",
  "mimeType": "image/jpeg",
  "labels": {
    "restricted": true,
    ...
  },
  "createdDate": "2008-10-13T13:06:03.186Z",
  "modifiedDate": "2012-06-06T16:06:32.795Z",
  "lastViewedDate": "2012-03-01T03:26:24.590Z",
  "userPermission": { ... },
  "editable": false, ... }
```

JSON

Exporting a Google Doc

```json
{ "kind": "drive#file",
  "id": "0Ak4pWYC-9qb5cGs1T3hoMmVXLUxSdTdvMWt6ZUVNUGc",
  "title": "Feeding plan",
  "mimeType": "application/vnd.google-apps.document",
  "labels": { ... },
  "exportLinks": {
    "text/html": "https://...",
    "application/pdf": "https://...",
    ...
  },
  "userPermission": { ... },
  "quotaBytesUsed": "0",
  "editable": true, ... }
```

JSON

Adding Touch Up Revisions

Creating a New Revision of a Photo

Request

`PUT /upload/drive/v2/files/<file-id>?newRevision=true`   **HTTP**

Response

`200 OK`   **JSON**

Java

```
Files.Update request =
    service.files().update("<file-id>", mediaContent);
request.setNewRevision(true);
request.execute();
```
JAVA

Pinning Revisions

Request

```
PATCH /drive/v2/files/<file-id>/revisions/<revision-id>
{ "pinned": true }
```

`HTTP`

Response

```
{ "kind": "drive#revision",
  "id": "1339430130979000",
  "downloadUrl": "https://..." } }
```

`JSON`

Java

```java
Revision patchedRevision = new Revision();
patchedRevision.setPinned(true);
service.revisions().patch(
    "<file-id>", "<revision-id>", patchedRevision).execute();
```

`JAVA`

Copies All Around

Copying a Photo

Request

```
POST /drive/v2/files/<file-id>/copy                    HTTP
{ "title": "Fancier Photo", ... }
```

Response

```
201 Created                                            JSON
{ "title": "Fancier Photo", ... }
```

Java

```java
File newCopy = new File();                             JAVA
newCopy.setTitle("Fancier Photo");
service.files().copy(originalFileId, newCopy).execute();
```

Processing on Copy

Request

`POST /drive/v2/files/<file-id>/copy?convert=true`

HTTP

Getting Rid of the Evidence

Trashing a Photo

Request

```
POST /drive/v2/files/<file-id>/trash
```

HTTP

Response

```
200 OK
{
  "labels": {
    "trashed": true,
    ...
  }
}
```

JSON

Java

```java
service.files().trash("<file-id>").execute();
```

JAVA

Deleting a Photo (Skips the Trash!)

Request

HTTP

```
DELETE /drive/v2/files/<file-id>
```

Response

JSON

```
200 OK
```

Java

JAVA

```java
service.files().delete("<file-id>").execute();
```

Upload
Download
Manage revisions
Copy
Delete
Organize
Share
Search
Changes

Photo Clutter

Adding a Photo to a Folder

Request

HTTP

```
POST /drive/v2/files/<folder-id>/children
{ "id": "<file-id>" }
```

Response

JSON

```
200 OK
```

Java

JAVA

```java
ChildReference newChild = new ChildReference();
newChild.setId("<file-id>");
service.children().insert("<folder-id>", newChild).execute();
```

Uploading Directly to Folders

Request

```
POST /upload/drive/v2/files
{
  "parents": [
    {
      "id": "<folder-id>"
    }
  ]
}
```

HTTP

Check Out My Pics!

Sharing Types

user

group

domain

Google.com
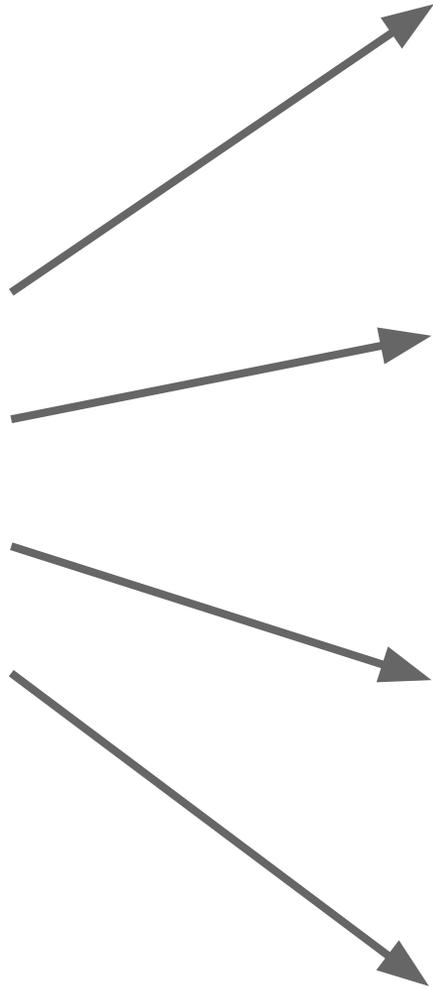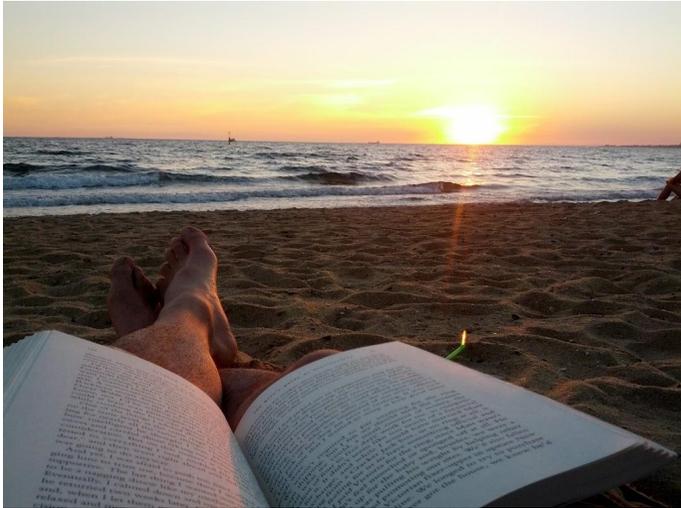
anyone

Sharing a Photo

Request

```
POST /drive/v2/files/<file-id>/permissions        HTTP
{ "type": "user",
  "role": "reader",
  "value": "aafshar@gmail.com" }
```

Response

```
{ "kind": "drive#permission",              JSON
  "id": "<permission-id>",
  "role": "reader",
  "type": "user" }
```

Where's That Photo?

Searching for Files

Request

```
GET /drive/v2/files?q=title contains 'birthday'
```

`HTTP`

Response

```
200 OK
{ "items": [ ... ] }
```

`JSON`

Java

```java
Files.List request = service.files().list();
request.setQ("title contains 'birthday'");
request.execute();
```

`JAVA`

Other Supported Queries

- title = 'Anniversary 2012'

- starred = true and trashed = false

- modifiedTime > '2012-06-27T14:00:00'

- fullText contains 'three sisters'

- 'john.smith@gmail.com' in writers

- mimeType = 'application/pdf'

Ooh, Activity in My Photo Album!

Monitoring Changes

Note the largest change ID for the user

unsynced

synced

current position

1   4   5   9
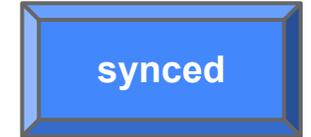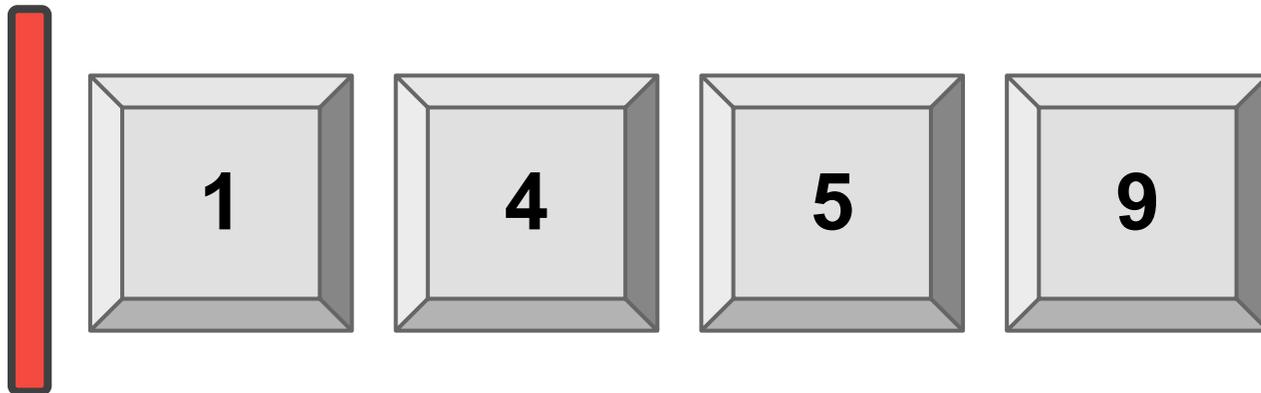
Monitoring Changes

Request all the files your app is interested in

1 4 5 9
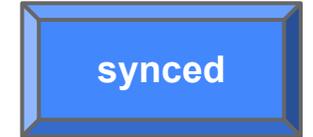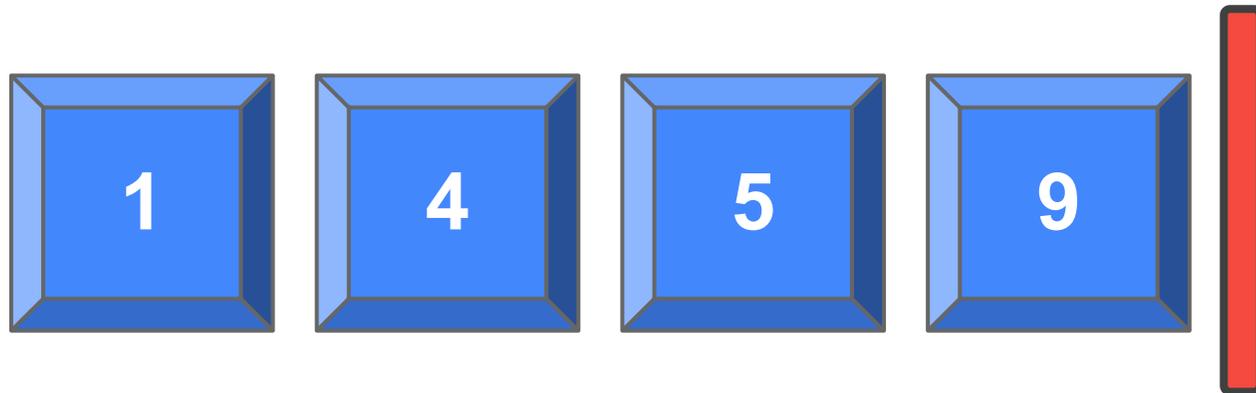
Monitoring Changes

Wait for changes
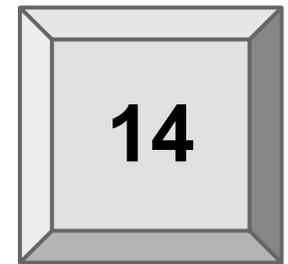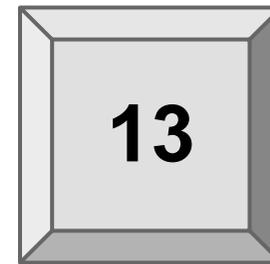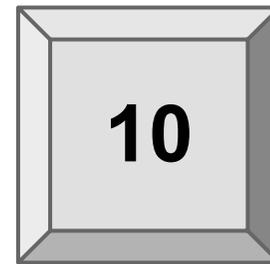
Monitoring Changes

On next poll, read in only new changes

synced

current
position

| 1 | | 5 | 9 | | 10 | 13 | 14 | |
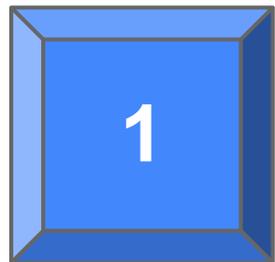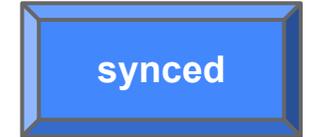
Largest Change ID

Request

```
GET /drive/v2/about
```

HTTP

Response

```json
{
    "largestChangeId": "9",
    ...
}
```

JSON

Changes Since a Change ID

Request

```
GET /drive/v2/changes?startChangeId=10
```
HTTP

Response

```
200 OK
{
  "kind": "drive#changeList",
  "nextPageToken": "16",
  "nextLink": "https://www.googleapis.com/drive/v2/changes?startChangeId=16",
  "items": [ { "kind": "drive#change",
               "id": "10",
               "fileId": "1q8IaKb8HeT3MMZmImo6pki4RtjgDDJ5HEQ7JuKW1H28",
               "deleted": false,
               "file": { "kind": "drive#file", "title": "Google I/O", ... }}]
}
```
JSON

Best Practices

Retrying Failed Calls

- Exponential Backoff

  - Start with one second

  - If fails again, retry with twice the wait

- Jitter

  - Avoid synchronized bursts of calls

- Errors

  - `403` with a rate limit exceeded or user rate

    limit exceeded exception should be retried

  - `500` should be retried

Every Byte is Sacred

- **`fields`**

    - Receive only the data you care about

- **`PATCH`**

    - Send only the data you want to

- Resumable upload

    - If the network fails, just resume the upload from where you left off

- Changes collection

    - Only grab file data that has changed

Future Considerations

Features and Improvements

- Properties for storing extra metadata

- Batching

- Performance

- Better media support for exposing common metadata

- Push notifications

- Collaboration engine

Community-driven approach to feature requests

<Thank You>
Questions?

Documentation: **https://developers.google.com/drive**

API Explorer: **https://developers.google.com/apis-explorer/#p/drive/v2/**

Ask a Question: **Stack Overflow Tag "google-drive-sdk"**

Ali Afshar

http://www.googplus.org/ali

Ivan Lee

http://www.googplus.org/ivan

Google Developers