



Embrace Opportunity



Software Engineering Day 13

Multi-part form data, Multer, Storing Images Locally

jX



Today's Overview

- Introduction to Multer Library for Backend.
- Introduction to Multi-part/FormData for Frontend.
- Cheat Sheet:
 - [iX Cheat Sheet](#)
 - [iX Cheat Sheet - Day 14](#)

Multer

Mutlipart/form-data Middleware

Multer - Introduction

- Multer is a powerful middleware for Node.js that simplified the file upload process.
- It handles the multipart/form-data requests.
- Multer adds to the *request* object
 - *body* object
 - Contains the values of the text fields of the form
 - *file* or *files* object
 - Contains the files uploaded via the form
- Multer only handles *multipart/form-data*, anything else it will not handle
 - It is important that the form includes: *enctype="multipart/form-data"*

Multer - Installation & Requirements

- Install:

```
npm install multer
```

sh

- Import:

```
const express = require('express');
const multer = require('multer');
```

Multer - Storage

- multer.diskStorage handles storing files to disk
- Accepts two functions:
 - destination:
 - Determine within which folder the uploaded files should be stored.
 - filename:
 - Determine what the file should be named inside the folder.
- Note:
 - Multer does not append any file extension, the function should include it.
 - Each function gets passed both the request (*req*) and some information about the file (*file*).

Multer - Storage

```
const storage = multer.diskStorage({  
  destination: function (req, file, cb) {  
    cb(null, '/uploads')  
  },  
  filename: function(req, file, cb) {  
    const uniqueSuffix = Date.now() + '-' + file.originalname;  
  }  
});  
  
const upload = multer({ storage: storage });
```

Multer - Route

- Handling the route of the upload request.
- '/upload' is the route in this example
- Using the *single* method for a single file upload.

```
const express = require('express');
const app = express();
const upload = require('./upload');

app.post('/upload', upload.single('file'), (req, res) => {
  res.json({ message: 'File uploaded successfully' });
});
```

Multer - Route Methods

- Single file:
 - `.single(fieldname)`
 - Accepts a single file.
 - Single file stored in `req.file`

```
app.post('/profile', upload.single('avatar'), function (req, res, next) {})
```

Multer - Route Methods

- Array of files:
 - `.array(fieldname[, MaxCount])`
 - `MaxCount` - Optional for maximum uploads.
 - Accepts an array of files.
 - Array stored in `req.files`

```
app.post('/photos/upload', upload.array('photos', 12), function (req, res, next) {})
```

Multer - Route Methods

- Mix of file:
 - `.fields(fields)`
 - Accepts a mix of files, specified by fields.
 - Object stored in `req.file`

```
const cpUpload = upload.fields([{ name: 'avatar', maxCount: 1 },
{ name: 'gallery', maxCount: 8 }])
app.post('/cool-profile', cpUpload, function (req, res, next) {})
```

Multi-part / FormData

File Upload

Multi-part/FormData - Introduction

- Multi-part/FormData is used for uploading files and sending data through an HTTP POST request.
- FormData object lets you compile a set of key/value pairs to send using the *fetch* API.
- Syntax:

```
const formData = new FormData();
```

FormData - Methods

```
// Append a key-value pair to the object
append()
// Delete a key-value pair
delete()
// Returns an iterator object, used to list key value pairs
entries()
// Return the value of a key
get()
// Return all values of a key
getAll()
// Check if key exists
has()
// Returns an iterator object, used to list available keys
keys()
// Add value to a key, replaces the value if key already exists
set()
// Returns an iterator object for the values of the FormData object
value()
```

Frontend

File Upload

React - FormImage

- We need a component to allow the user to choose an image file from their computer.
- On day 2 we learned we can use the input component with the type set to file.
- Let's create a FormImage component to do this.

```
<input  
  ref={fileInput}  
  style={{ display: "none" }}  
  id="fileInput"  
  type="file"  
  name="img"  
  onChange={onChange}  
></input>
```

[Full code snippet](#)

React - Update AddEditBlogModal

- Let's update AddEditBlogModal component to cater for the new FormImage component and transform the image along with the blog details into multi-part form data and pass it to the blog service when creating and editing blogs.

[Full code snippet](#)

React - Update Blog Service

- Let's update blog service to remove the content type header to multi-part form payloads.

[Full code snippet](#)

Consuming Data

From server

Consuming Data

- We can expose the files they were uploaded with express by using the use middleware and path package to route http get requests to /uploads/[filepath] and return the image if it exists.

```
...
const path = require("path");
...

app.use("/uploads", express.static(path.join(__dirname, "../uploads")));
```

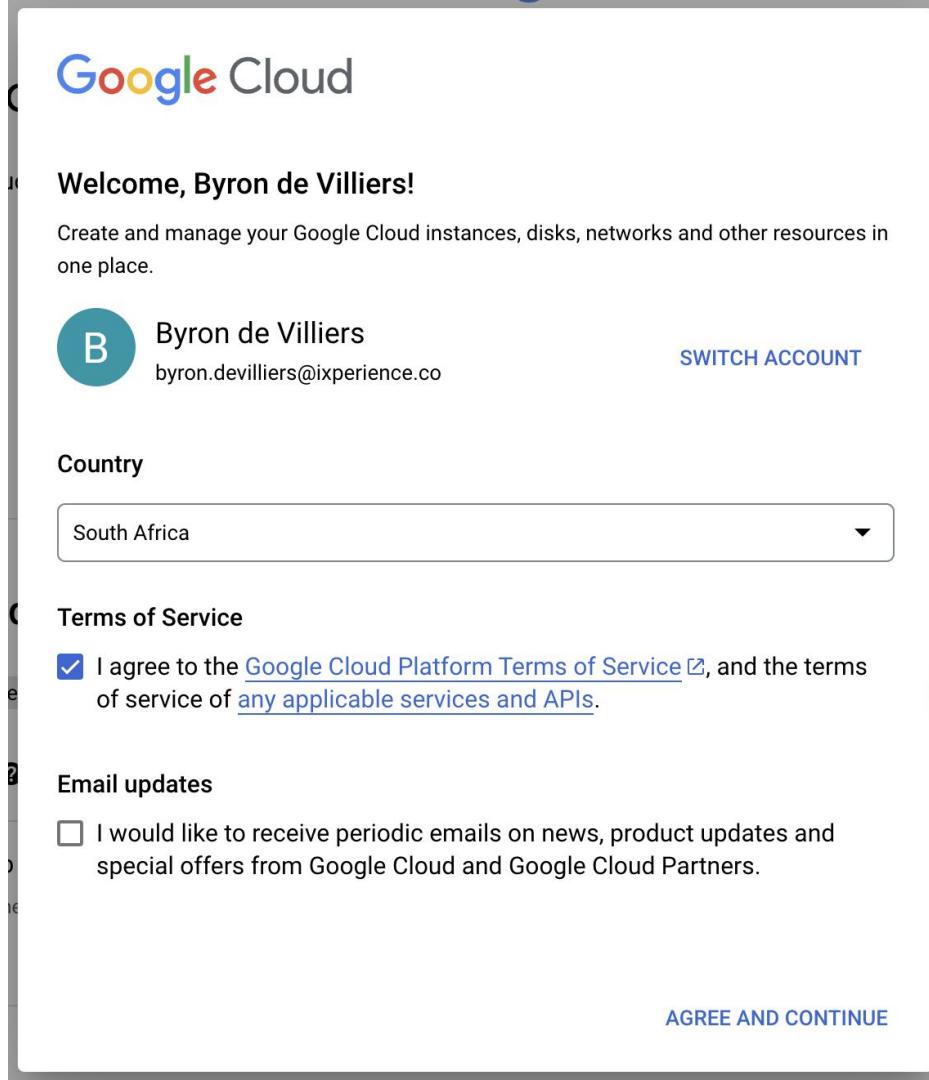
Google Cloud Storage

Save images in GPC

Create Google Cloud Account

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

*Note: You'll have to enter in credit card details.



The screenshot shows the Google Cloud account creation interface. At the top, the Google Cloud logo is displayed. Below it, a welcome message reads "Welcome, Byron de Villiers!". A brief description follows: "Create and manage your Google Cloud instances, disks, networks and other resources in one place." On the left, there's a circular profile picture containing a white letter 'B'. To its right, the user's name "Byron de Villiers" and email "byron.devilliers@ixperience.co" are listed. A "SWITCH ACCOUNT" link is located in the top right corner. The next section, "Country", has a dropdown menu set to "South Africa". Below this, under "Terms of Service", a checked checkbox indicates agreement to the Google Cloud Platform Terms of Service and other applicable services and APIs. Under "Email updates", an unchecked checkbox allows for periodic emails from Google Cloud and Google Cloud Partners. At the bottom right, a large blue button labeled "AGREE AND CONTINUE" is visible.

Google Cloud

Welcome, Byron de Villiers!

Create and manage your Google Cloud instances, disks, networks and other resources in one place.

B Byron de Villiers
byron.devilliers@ixperience.co [SWITCH ACCOUNT](#)

Country

South Africa ▾

Terms of Service

I agree to the [Google Cloud Platform Terms of Service](#), and the terms of service of [any applicable services and APIs](#).

Email updates

I would like to receive periodic emails on news, product updates and special offers from Google Cloud and Google Cloud Partners.

[AGREE AND CONTINUE](#)

Create Google Cloud Project

The screenshot shows the Google Cloud homepage with a focus on creating a new project. At the top, there's a banner offering a \$300 free trial credit. Below the banner, the navigation bar includes 'Google Cloud' and a dropdown menu labeled 'Select a project' which is highlighted with a green box. A search bar is also present. On the left, a 'Welcome' message for 'Byron de Villiers' is displayed, along with a list of benefits for the trial: 'Access to Google Cloud products and services', '90 days to spend your credits', and 'No billing during trial'. A 'TRY FOR FREE' button is located at the bottom of this section. To the right, there are sections for 'Other options' (with a 'TRY GEMINI' button), 'Configure Google Cloud for scalable, production-ready enterprise workloads' (with a 'GOOGLE CLOUD SETUP' button), and 'Popular getting started resources'. These resources are categorized by tags like 'Web, mobile, game, storage', 'Containers, VMs, hybrid/multi, move workload', 'Data, AI/ML, SAP', 'Maps, APIs', and 'General'. Below this, there are 'Pre-built solution templates' for deploying a three-tier web app, load-balanced managed VMs, or a data warehouse using BigQuery. There's also a 'View all solutions' button. The 'Products' section features four cards: 'Create a VM' (Compute Engine), 'Train and host ML models' (Vertex AI), 'Create a database' (Cloud SQL), and 'Analyse and manage data' (BigQuery).

Start your free trial with \$300 in credit. Don't worry – you won't be charged if you run out of credit. [Learn more](#)

DISMISS START FREE

Google Cloud Select a project Search / for resources, docs, products and more

Welcome, Byron de Villiers

Try Google Cloud with \$300 in free credits

- ✓ Access to Google Cloud products and services
- ✓ 90 days to spend your credits
- ✓ No billing during trial

TRY FOR FREE

Other options

TRY GEMINI

Configure Google Cloud for scalable, production-ready enterprise workloads.

GOOGLE CLOUD SETUP

Popular getting started resources

Filter by Web, mobile, game, storage Containers, VMs, hybrid/multi, move workload Data, AI/ML, SAP Maps, APIs General

Pre-built solution templates

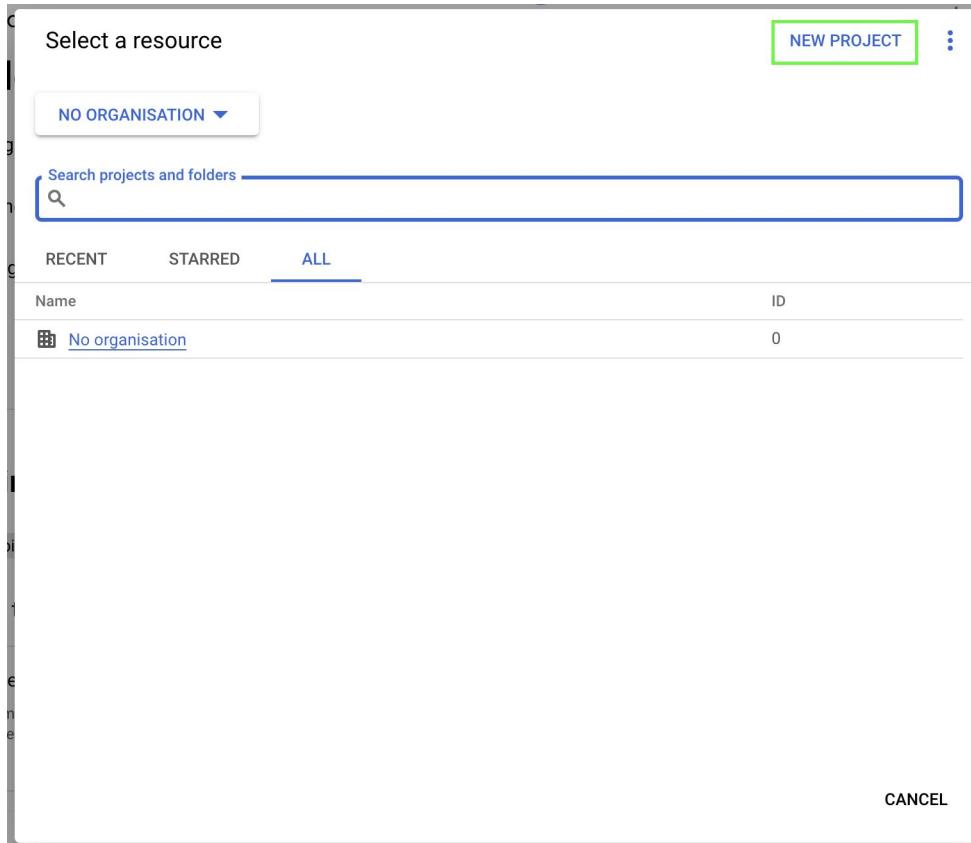
- Deploy a three-tier web app
- Deploy load-balanced managed VMs
- Create a data warehouse with BigQuery

View all solutions

Products

- Create a VM
- Train and host ML models
- Create a database
- Analyse and manage data

Create Google Cloud Project



Create Google Cloud Project

New Project



You have 18 projects remaining in your quota. Request an increase or delete projects. [Learn more](#)

[MANAGE QUOTAS](#)

Project name *

Blog App



Project ID: bubbly-shield-426823-e1. It cannot be changed later. [EDIT](#)

Location *

No organisation

BROWSE

Parent organisation or folder

CREATE

CANCEL

Create Google Cloud Project

Google Cloud Blog App Search (/) for resources, docs, products and more Search 1 ? : 🚙

Welcome

You're working in [Blog App](#)

Project number: 862222340568

Project ID: blog-app-426723

[Dashboard](#) [Recommendations](#)

[Create a VM](#) [Run a query in BigQuery](#) [Create a GKE cluster](#) [Create a storage bucket](#)

Try our most advanced model: Gemini 1.5 Pro

Try Gemini →

Quick access

[API APIs and services](#)

[IAM and admin](#)

[Billing](#)

[Compute Engine](#)

[Cloud Storage](#)

[BigQuery](#)

[VPC network](#)

[Kubernetes engine](#)

[VIEW ALL PRODUCTS](#)

Create Google Cloud set billing account

Google Cloud Search (/) for resources, docs, products and more Search

Billing SHOW INFO PANEL

MY BILLING ACCOUNTS MY PROJECTS

CREATE ACCOUNT

Filter Status : Active Enter property name or value

<input type="checkbox"/> Billing account name	Billing account ID	Status	Last 30 days' spend	Account type	Organisation	Health checks
Personal Billing Account	011DE5-799D43-9A4AD3	Active	\$0	Direct	-	 2

Google Cloud Search (/) for resources, docs, products and more Search

Billing SHOW INFO PANEL

MY BILLING ACCOUNTS **MY PROJECTS**

Filter Enter property name or value

Name	ID	Billing account ↑	Billing account ID	Actions
My Billing Account	011DE5-799D43-9A4AD3	Personal Billing Account	011DE5-799D43-9A4AD3	
Jacques Pfeiferer Account	011F00-B99F7E-960F0	My Billing Account	011F00-B99F7E-960F0	
My Billing Account	011FCA-B6E5A-ABDAB	My Billing Account	011FCA-B6E5A-ABDAB	
My Billing Account	011FCA-B6E5A-ABDAB	My Billing Account	011FCA-B6E5A-ABDAB	
Blog App	blog-app-426723	Personal Billing Account	011DE5-799D43-9A4AD3	
dangerous	dangerous-111-111	Dangerous	dangerous-111-111	
dangerous	dangerous-111-111	Dangerous	dangerous-111-111	

Create Google Cloud Create Bucket

The screenshot shows the Google Cloud Platform dashboard with the 'Blog App' selected in the top navigation bar. The left sidebar lists various services under 'Cloud Storage' (highlighted with a green box) and 'Buckets' (also highlighted with a green box). The main content area displays three promotional cards:

- Soft delete is now available**: A card explaining that soft delete protects objects from accidental or malicious deletions. It mentions a seven-day soft delete policy has been added to all existing buckets and is the default policy for new buckets.
- Power near real-time analytics and replication with event-driven transfers**: A card about capturing changes faster at Google Cloud Storage and Amazon S3 sources via event-driven transfers. It includes a 'CREATE TRANSFER JOB' button.
- Introducing the folder browser and folder permissions (using managed folders)**: A card about the enhanced object browser with folder permissions. It includes a 'LEARN MORE' button.

At the bottom right of the dashboard, there is a small icon with a yellow dot and a red dot.

Create Google Cloud Create Bucket

✓ Name your bucket

Pick a globally unique, permanent name. [Naming guidelines](#)

ix-blog-app-session-1

Tip: Don't include any sensitive information

✗ LABELS (OPTIONAL)

[CONTINUE](#)

• Choose where to store your data

This choice defines the geographic placement of your data and affects cost, performance and availability. Cannot be changed later. [Learn more](#)

Location type

Multi-region

Highest availability across largest area

us (multiple regions in United States)

Dual-region

High availability and low latency across 2 regions

Region

Lowest latency within a single region

[CONTINUE](#)

• Choose a storage class for your data

A storage class sets costs for storage, retrieval and operations, with minimal differences in uptime. Choose if you want objects to be managed automatically or specify a default storage class based on how long you plan to store your data and your workload or use case. [Learn more](#)

Autoclass

Automatically transitions each object to Standard or Nearline class based on object-level activity, to optimise for cost and latency. Recommended if usage frequency may be unpredictable. Can be changed to a default class at any time. [Pricing details](#)

Set a default class

Applies to all objects in your bucket unless you manually modify the class per object or set object lifecycle rules. Best when your usage is highly predictable.

Standard

Best for short-term storage and frequently accessed data

Nearline

Best for backups and data accessed less than once a month

Coldline

Best for disaster recovery and data accessed less than once a quarter

Archive

Best for long-term digital preservation of data accessed less than once a year

[CONTINUE](#)

• Choose how to control access to objects

Prevent public access

Restrict data from being publicly accessible via the Internet. Will prevent this bucket from being used for web hosting. [Learn more](#)

Enforce public access prevention on this bucket

Access control

Uniform

Ensure uniform access to all objects in the bucket by using only bucket-level permissions (IAM). This option becomes permanent after 90 days. [Learn more](#)

Fine-grained

Specify access to individual objects by using object-level permissions (ACLs) in addition to your bucket-level permissions (IAM). [Learn more](#)

[CONTINUE](#)

• Choose how to control access to objects

Prevent public access

Restrict data from being publicly accessible via the Internet. Will prevent this bucket from being used for web hosting. [Learn more](#)

Enforce public access prevention on this bucket

Access control

Uniform

Ensure uniform access to all objects in the bucket by using only bucket-level permissions (IAM). This option becomes permanent after 90 days. [Learn more](#)

Fine-grained

Specify access to individual objects by using object-level permissions (ACLs) in addition to your bucket-level permissions (IAM). [Learn more](#)

[CONTINUE](#)



Create Google Cloud Service Account

The screenshot shows two views of the Google Cloud IAM interface. On the left, the navigation pane for a project named 'Blog App' is visible, with the 'Cloud Storage' section currently selected. Under 'Cloud Storage', the 'Service accounts' option is highlighted with a green box. On the right, the main 'IAM' page for the 'Blog App' project is displayed. The 'Service accounts' tab is selected, also highlighted with a green box. A single entry for 'Byron.div@gmail.com' is listed under the 'Service accounts' table, with the role 'Owner' assigned.

Google Cloud

Blog App

Search (/)

Cloud overview

Solutions

PINNED PRODUCTS

API APIs and services

Billing

IAM and admin

Marketplace

Compute Engine

Kubernetes Engine

Cloud Storage

BigQuery

VPC network

Cloud Run

SQL

Logging

Security

VIEW ALL PRODUCTS

Bucket details

IAM

PAM NEW

Principal access boundary NEW

Identity and organisation

Policy troubleshooter

Policy analyser NEW

Organisation policies

Service accounts

Workload Identity Federation

Workforce identity federation

Labels

Tags

Settings

Privacy and security

Identity-Aware Proxy

Roles

Audit logs

Manage resources

Create a project

Essential contacts

Asset inventory

Quotas and system limits

Google Cloud

Blog App

Search (/) for resources, docs, products and more

1

?

LEARN

IAM and admin

IAM

PAM NEW

Principal access boundary

Identity and organisation

Policy troubleshooter

Policy analyser NEW

Organisation policies

Service accounts

Workload Identity Federation

Workforce identity federation

Labels

PERMISSION

RECOMMENDATIONS HISTORY

Permissions for project Blog App

These permissions affect this project and all of its resources. [Learn more](#)

Include Google-provided role grants

VIEW BY PRINCIPALS

VIEW BY ROLES

GRANT ACCESS

REMOVE ACCESS

Filter Enter property name or value

Type	Principal ↑	Name	Role	Security insights
	Byron.div@gmail.com	Byron de Villiers	Owner	

iX

Create Google Cloud Service Account

1 Service account details

Service account name

Display name for this service account

Service account ID * X C

Email address: storage-account@blog-app-426723.iam.gserviceaccount.com ✉

Service account description

Describe what this service account will do

CREATE AND CONTINUE

2 Grant this service account access to the project (optional)

3 Grant users access to this service account (optional)

DONE

CANCEL

⬅ Create service account

1 Service account details

2 Grant this service account access to the project (optional)

Grant this service account access to Blog App so that it has permission to complete specific actions on the resources in your project. [Learn more](#)

Select a role Filter Filter by role or permission IAM condition (optional) ? ✖

Cloud Services	Roles
Cloud Spanner	Storage Admin
Cloud SQL	Storage Folder Admin
Cloud Storage	Storage Admin
Cloud Talent Solution	Storage HMAC Key Admin
Cloud Tasks	Storage Insights Collector Service
Cloud TPU	Storage Object Admin
	Storage Object Creator

Storage Admin
Grants full control of buckets and objects.

DONE MANAGE ROLES

Create Google Cloud Download Service Account Key

Service accounts + CREATE SERVICE ACCOUNT DELETE MANAGE ACCESS REFRESH LEARN

From 16 June 2024, Google will automatically disable service account keys detected in public repositories. You can customise this behaviour by using the 'iam.serviceAccountKeyExposureResponse' organisation policy. [Learn more](#)

DISMISS

Service accounts for project 'Blog App'

A service account represents a Google Cloud service identity, such as code running on Compute Engine VMs, App Engine apps or systems running outside Google. [Learn more about service accounts](#)

Organisation policies can be used to secure service accounts and block risky service account features, such as automatic IAM Grants, key creation/upload or the creation of service accounts entirely. [Learn more about service account organisation policies](#)

<input type="checkbox"/> Filter	Enter property name or value	<input type="checkbox"/>	<input type="checkbox"/>				
	Status	Name ↑	Description	Key ID	Key creation date	OAuth 2 client ID <input type="checkbox"/>	Actions
<input type="checkbox"/>	 Enabled	Storage Account	No keys			103738249147038565308	<ul style="list-style-type: none"><input type="checkbox"/> Manage details<input type="checkbox"/> Manage permissions<input checked="" type="checkbox"/> Manage keys<input type="checkbox"/> View metrics<input type="checkbox"/> View logs<input type="checkbox"/> Disable<input type="checkbox"/> Delete

iX

Create Google Cloud Download Service Account Key

[← Storage Account](#)

DETAILS PERMISSIONS KEYS METRICS LOGS

Keys

⚠ Service account keys could pose a security risk if compromised. We recommend that you avoid downloading service account keys and instead use the [Workload Identity Federation](#). You can learn more about the best way to authenticate service accounts on Google Cloud [here](#).

ℹ Starting 16 June 2024, Google will automatically disable service account keys detected in public repositories. You can customise this behaviour by using the 'iam.serviceAccountKeyExposureResponse' organisation policy. [Learn more](#)

Add a new key pair or upload a public key certificate from an existing key pair.

Block service account key creation using [organisation policies](#).

[Learn more about setting organisation policies for service accounts](#)

ADD KEY ▾

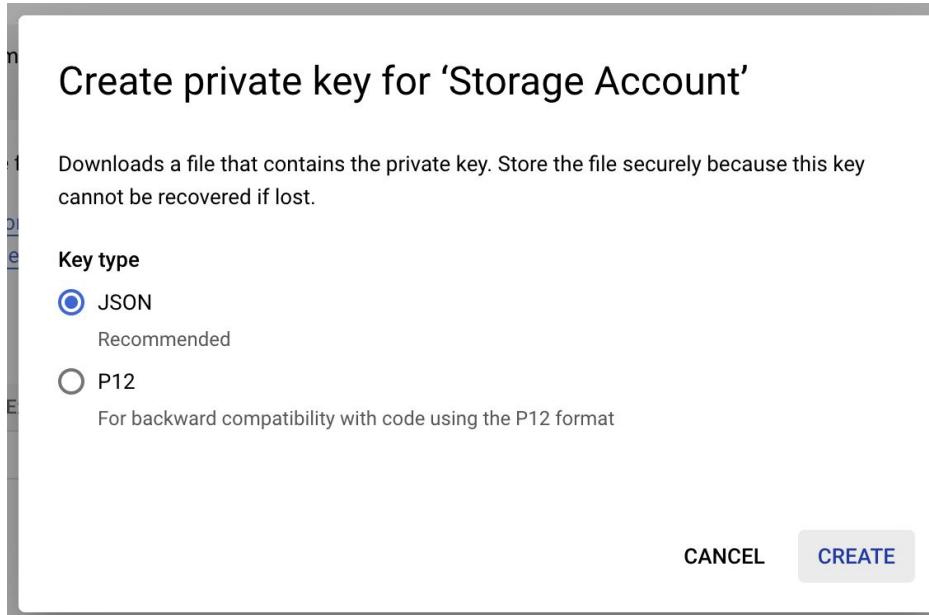
Create new key

Creation date Expiry date

Upload existing key

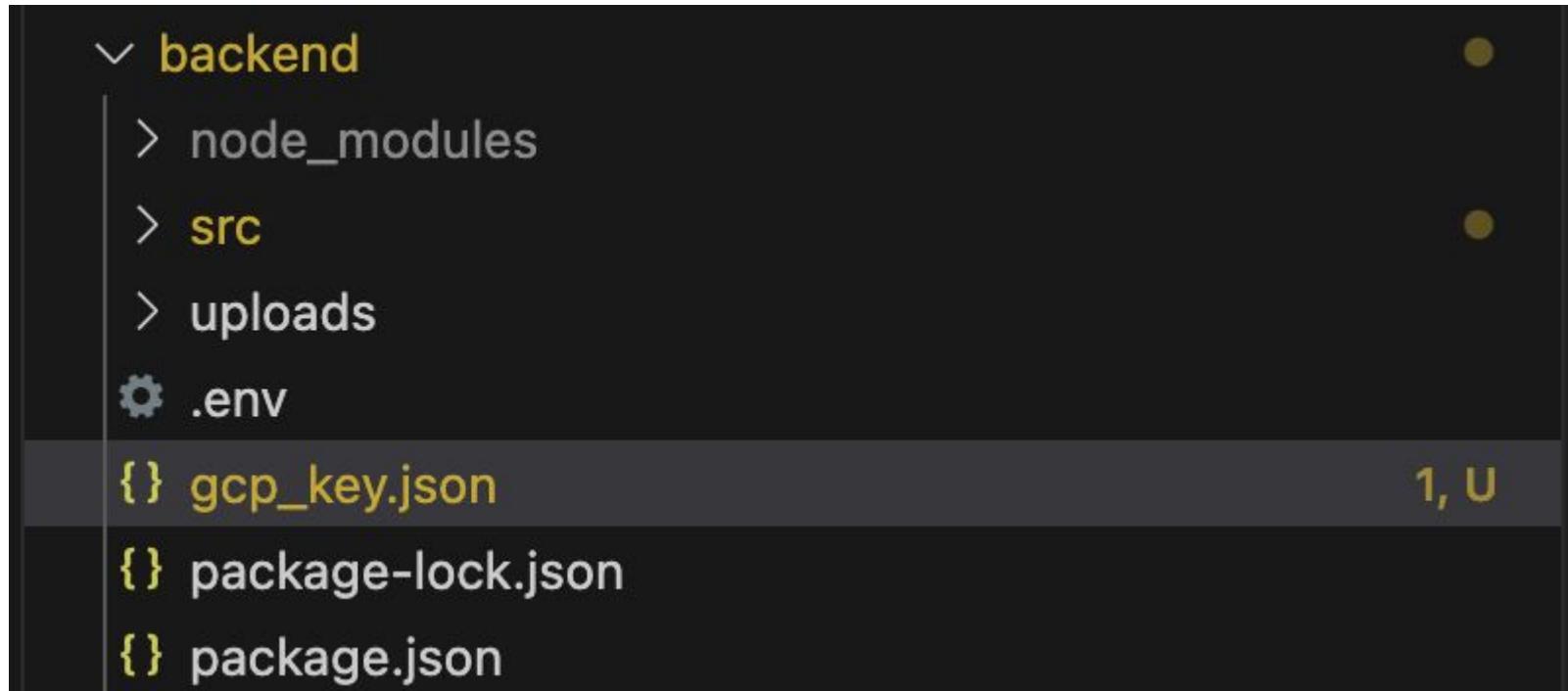


Create Google Cloud Download Service Account Key



Add Storage Service Account to project

- Rename json file to gcp_key.json and add it to `backend/`



```
backend
  node_modules
  src
  uploads
  .env
  gcp_key.json
  package-lock.json
  package.json
```

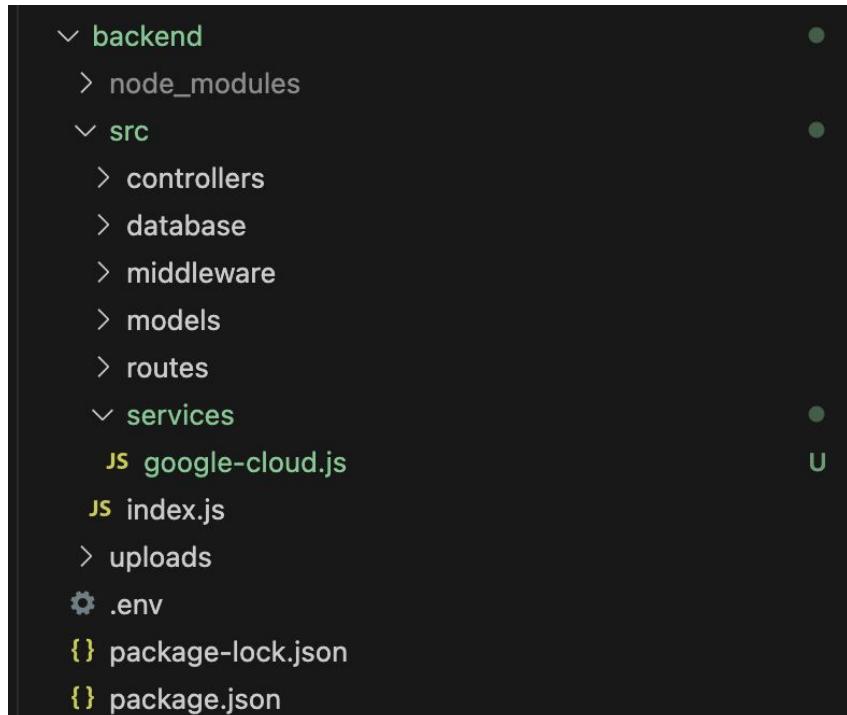
Install @google-cloud/storage

```
npm install @google-cloud/storage
```

sh

Create Google Cloud Storage

- In backend/src/services/ create a file called *cloud-storage.js*



```
└── backend
    ├── node_modules
    └── src
        ├── controllers
        ├── database
        ├── middleware
        ├── models
        ├── routes
        └── services
            ├── google-cloud.js
            ├── index.js
            ├── uploads
            └── .env
    └── package-lock.json
    └── package.json
```

Define GoogleCloudService

- Please view the [full code snippet](#).

Homework

Apply what we have learned

Homework

- Implement file upload for authenticated users.

Next Class

Finalising Capstone Projects and
Preparing for Presentations

