**Generating an** **API Proxy Using an OpenAPI Specification**

**Apigee:**

Apigee is a platform for developing and managing APIs (Application Programming Interface). By fronting services with a proxy layer, Apigee provides an abstraction or facade for your backend service APIs and provides security, rate limiting, quotas, analytics, and more.

Apigee consists of the following primary components:

**Apigee services:** The APIs that you use to create, manage, and deploy your API proxies.

**Apigee runtime:** A set of containerized runtime services in a Kubernetes cluster that Google maintains. All API traffic passes through and is processed by these services.

In addition, Apigee uses other components including:

**GCP (Google Cloud Platform) services:** Provides identity management, logging, analytics, metrics, and project management functions.

**Back-end services:** Used by your apps to provide runtime access to data for your API proxies.

**Objectives:**

* Explore an OpenAPI specification and understand the different components.
* Create an API proxy from an OpenAPI specification using the proxy wizard.
* Trace an API proxy.

1. In Cloud console, on the top right toolbar, click the Open Cloud Shell button.
2. Click **Continue**.

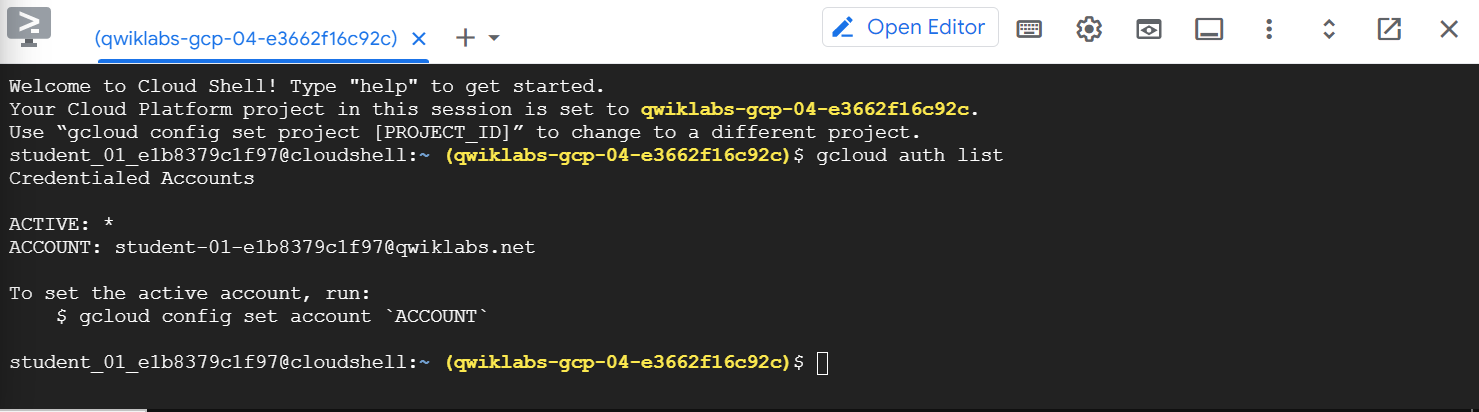
Graphical user interface, text, application, email

Description automatically generated

1. It takes a few moments to provision and connect to the environment. When you are connected, you are already authenticated, and the project is set to your PROJECT\_ID. For example:

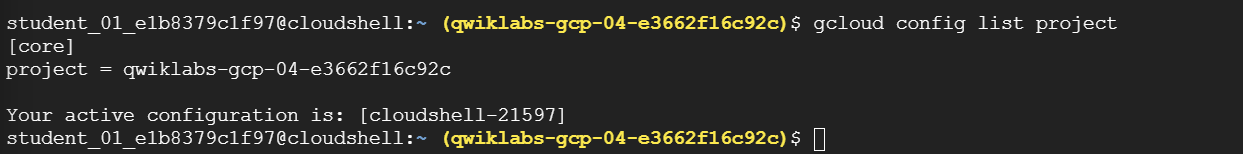
* You can list the active account name with this command:

à gcloud auth list

****

* You can list the project ID with this command:

à gcloud config list project

****

**Open the Apigee console**

The Apigee console is accessed on a page separate from the Google Cloud Console.

* In a different browser tab within the same window, open the Apigee console

You may also open the Apigee console from the Google Cloud Console by opening the **Navigation menu** (Navigation menu icon) and selecting **Tools > Apigee**.

**Graphical user interface, text, application

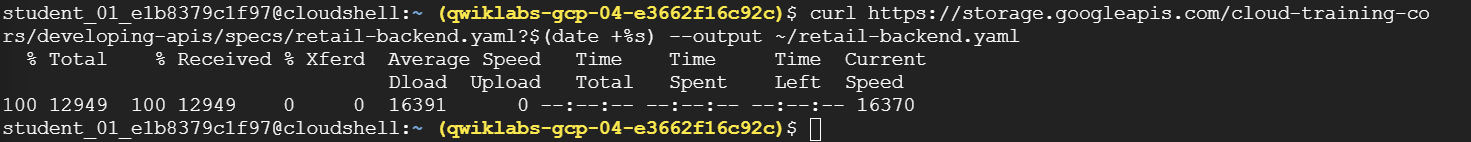
Description automatically generated**

**Examine the OpenAPI specification for the backend service**

Download the OpenAPI specification

* In **Cloud Shell**, download the OpenAPI specification for the backend service using this [curl](https://curl.se/docs/httpscripting.html) command:

curl https://storage.googleapis.com/cloud-training-cors/developing-apis/specs/retail-backend.yaml?$(date +%s) --output ~/retail-backend.yaml

****

This curl command downloads a file named “retail-backend.yaml” and stores it in a file with the same name in the home directory. Later, you will use this same specification when creating an API proxy.

**Graphical user interface, text

Description automatically generated**

The **openapi** field specifies the version of the OpenAPI specification. This is an OpenAPI Version 3 spec, as indicated by the version number at the top of the file:

openapi: "3.0.0"

The **info** object provides metadata about the API itself. The version shown is the version of the Retail Backend specification:

info:

  version: 1.0.0

  title: Retail Backend

  description: Retail backend service used for Developing APIs course

  contact:

    name: Google Cloud (Apigee)

    email: apigee@example.org

    url: https://cloud.google.com/apigee

  license:

    name: MIT

    url: <https://opensource.org/licenses/MIT>

The **servers** array contains a list of server objects that specify connectivity information for target servers. This specification contains a single backend service, which your API proxy will call:

servers:

  - url: "https://gcp-cs-training-01-test.apigee.net/training/db"

    description: Retail backend for Developing APIs course

The **tags** array adds metadata to tags used in the operations, which are shown below. Tags may be shared by multiple operations, and tags can be used to provide verbose descriptions or links to external documentation.

tags:

  - name: categories

    description: Product Categories

  - name: products

    description: Products

  - name: orders

    description: Orders

  - name: stores

    description: Stores

The **paths** object holds the relative paths to individual endpoints and their operations. One such path, **/categories/{categoryId}**, is used to specify a single category. Shown here is a **get** operation specified to get a category by ID. The get object shows parameters and responses. For operations that contain a request body, like **PATCH /products/{productId}**, the request body will also be specified.

paths:

  /categories:

    get:

      summary: Get all categories

      operationId: getCategories

      tags:

        - categories

      responses:

        '200':

          description: Complete list of categories

          content:

            application/json:

              schema:

                $ref: "#/components/schemas/Categories"

        default:

          description: unexpected error

          content:

            application/json:

              schema:

                $ref: "#/components/schemas/Error"

  /categories/{categoryId}:

    get:

      summary: Get a specific category

      operationId: getCategoryById

      tags:

        - categories

      parameters:

        - name: categoryId

in: path

          required: true

          description: category id

          schema:

            type: integer

      responses:

        '200':

          description: Selected category

          content:

            application/json:

              schema:

                $ref: "#/components/schemas/Category"

        '404':

          description: Category not found

          content:

            application/json:

              schema:

                $ref: "#/components/schemas/Error"

        default:

          description: unexpected error

          content:

            application/json:

              schema:

                $ref: "#/components/schemas/Error"

  /stores:

    get:

      summary: Get all stores

      operationId: getStores

      tags:

        - stores

      responses:

        '200':

          description: Complete list of stores

          content:

            application/json:

              schema:

                $ref: "#/components/schemas/Stores"

        default:

          description: unexpected error

          content:

            application/json:

              schema:

                $ref: "#/components/schemas/Error"

  /stores/{storeId}:

    get:

      summary: Get a specific store

      operationId: getStoreById

      tags:

        - stores

      parameters:

        - name: storeId

in: path

          required: true

          description: store id

          schema:

            type: string

      responses:

        '200':

          description: Selected store

          content:

            application/json:

              schema:

                $ref: "#/components/schemas/Store"

        '404':

          description: Store not found

          content:

            application/json:

              schema:

                $ref: "#/components/schemas/Error"

        default:

          description: unexpected error

          content:

            application/json:

              schema:

                $ref: "#/components/schemas/Error"

  /products:

    get:

      summary: Get all products

      operationId: getProducts

      tags:

        - products

      responses:

        '200':

          description: Complete list of products

          content:

            application/json:

              schema:

                $ref: "#/components/schemas/Products"

        default:

          description: unexpected error

          content:

            application/json:

              schema:

                $ref: "#/components/schemas/Error"

  /products/{productId}:

    get:

      summary: Get a specific product

      operationId: getProductById

      tags:

        - products

      parameters:

        - name: productId

in: path

          required: true

          description: product id

          schema:

            type: string

      responses:

        '200':

          description: Selected product

          content:

            application/json:

              schema:

                $ref: "#/components/schemas/Product"

        '404':

          description: Product not found

          content:

            application/json:

              schema:

                $ref: "#/components/schemas/Error"

        default:

          description: unexpected error

          content:

            application/json:

              schema:

                $ref: "#/components/schemas/Error"

    patch:

      summary: Update a specific product

      description: Only overall\_rating can be changed, and it must be a number.

      operationId: updateProductById

      tags:

        - products

      parameters:

        - name: productId

in: path

          required: true

          description: product id

          schema:

            type: string

      requestBody:

        description: Product fields to update

        required: true

        content:

          application/json:

            schema:

              $ref: '#/components/schemas/ProductForUpdate'

      responses:

        '200':

          description: Returns the request product update fields

          content:

            application/json:

              schema:

                $ref: "#/components/schemas/ProductForUpdate"

        '401':

          description: Unauthorized

          content:

            application/json:

              schema:

                $ref: "#/components/schemas/Error"

        '404':

          description: Product not found

          content:

            application/json:

              schema:

                $ref: "#/components/schemas/Error"

        default:

          description: unexpected error

          content:

            application/json:

              schema:

                $ref: "#/components/schemas/Error"

      security:

        - basicAuth: []

  /orders:

    post:

      summary: Create a new order

      operationId: createOrder

      tags:

        - orders

      requestBody:

        description: New order fields (freeform)

        required: true

        content:

          application/json:

            schema:

              $ref: '#/components/schemas/Order'

      responses:

        '200':

          description: Order created successfully

          content:

            application/json:

              schema:

                $ref: "#/components/schemas/NewOrderResponse"

        default:

          description: unexpected error

          content:

            application/json:

              schema:

                $ref: "#/components/schemas/Error"

  /orders/{orderId}:

    get:

      summary: Get a specific order

      operationId: getOrderById

      tags:

        - orders

      parameters:

        - name: orderId

in: path

          required: true

          description: order id

          schema:

            type: string

      responses:

        '200':

          description: Selected order

          content:

            application/json:

              schema:

                $ref: "#/components/schemas/Order"

        '404':

          description: Order not found

          content:

            application/json:

              schema:

                $ref: "#/components/schemas/Error"

        default:

          description: unexpected error

          content:

            application/json:

              schema:

                $ref: "#/components/schemas/Error"

    delete:

      summary: Delete a specific order

      operationId: deleteOrderById

      tags:

        - orders

      parameters:

        - name: orderId

in: path

          required: true

          description: order id

          schema:

            type: string

      responses:

        '200':

          description: Null response

          content:

            application/json:

              schema:

                $ref: "#/components/schemas/DeleteOrderResponse"

        '404':

          description: Order not found

          content:

            application/json:

              schema:

                $ref: "#/components/schemas/Error"

        default:

          description: unexpected error

          content:

            application/json:

              schema:

                $ref: "#/components/schemas/Error"

The **components** object contains reusable objects for different portions of the OpenAPI spec. The **security Schemes** component object contains definitions of different types of security schemes used by the operations. This spec defines a single Basic Authentication scheme, which is referenced in the **PATCH /products/{productId}** operation. The **schemas** component object contains input and output data types. Shown here is the **Category** object, which is returned when the **GET /categories/{categoryId}** operation returns successfully:

components:

  securitySchemes:

    basicAuth:

      type: http

      description: basic auth

      scheme: basic

  schemas:

    Category:

      type: object

      description: product category

      required:

        - color

        - id

        - name

      properties:

        color:

          description: use this color for displaying this category

          type: string

        id:

          description: integer id (used for access)

          type: integer

          format: int32

          minimum: 0

        name:

          description: category name

          type: string

    Categories:

      type: array

      description: array of product categories

      items:

        $ref: "#/components/schemas/Category"

    Store:

      type: object

      description: store

      required:

        - id

        - name

        - site\_type

      properties:

        adr:

          $ref: "#/components/schemas/Address"

        affiliate:

          description: affiliate name

          type: string

        id:

          description: integer id (NOT used for access)

          type: integer

          format: int32

          minimum: 0

        location:

          $ref: "#/components/schemas/Location"

        name:

          description: store name

          type: string

        phone:

          description: store phone number

          type: string

        site\_type:

          description: type of site

          type: string

          enum:

            - store

            - warehouse

    Stores:

      type: object

      description: list of stores

      additionalProperties:

        $ref: "#/components/schemas/Store"

    Address:

      type: object

      description: store address

      required:

        - addr1

        - city

        - country

        - state

        - zip

      properties:

        addr1:

          description: street address 1

          type: string

        addr2:

          description: street address 2

          type: string

        city:

          description: city

          type: string

        country:

          description: country

          type: string

        state:

          description: state

          type: string

        zip:

          description: zip code

          type: string

    Location:

      type: object

      description: store location

      required:

        - latitude

        - longitude

      properties:

        latitude:

          description: latitude in decimal degrees

          type: number

          format: float

          minimum: -90

          maximum: 90

        longitude:

          description: longitude in decimal degrees

          type: number

          format: float

          minimum: -180

          maximum: 180

    Product:

      type: object

      description: product

      required:

        - category

        - image

        - name

        - overall\_rating

        - product\_name

        - short\_description

      properties:

        category:

          description: category name

          type: string

        image:

          description: image url

          type: string

        name:

          description: product id

          type: string

        overall\_rating:

          description: rating of product

          type: number

          format: float

        product\_name:

          description: descriptive product name

          type: string

        short\_description:

          description: short paragraph describing the product

          type: string

    ProductForUpdate:

      type: object

      description: product (updateable fields)

      required:

        - overall\_rating

      properties:

        overall\_rating:

          description: rating of product

          type: number

          format: float

    Products:

      type: object

      description: list of products

      additionalProperties:

        $ref: "#/components/schemas/Product"

    Order:

      type: object

      description: order request

      additionalProperties: true

    NewOrderResponse:

      type: object

      description: order response

      required:

        - name

      properties:

        name:

          description: string id (used for access)

          type: string

    DeleteOrderResponse:

      type: object

      nullable: true

    Error:

      type: object

      description: error response

      required:

        - error

        - error\_description

      properties:

        error:

          type: string

        error\_description:

          type: string

**Create an API proxy using the OpenAPI spec**

**Graphical user interface, text

Description automatically generated**

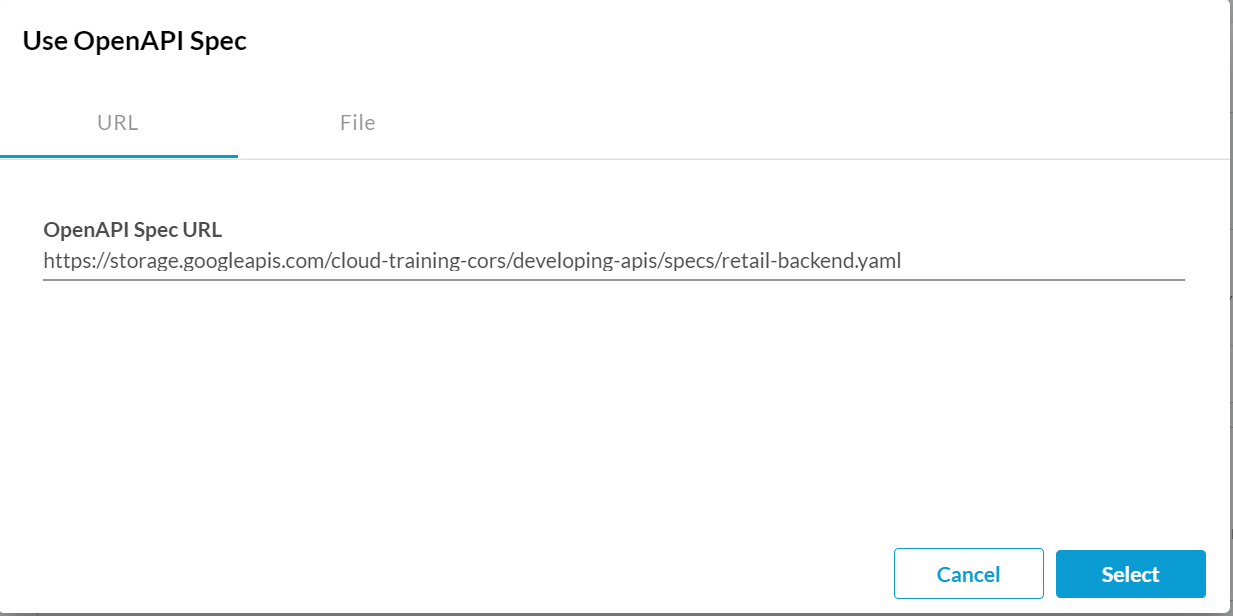
**Use the proxy wizard to create a proxy**

1. Select the Apigee console tab in your browser window.
2. On the left navigation menu, select **Develop > API Proxies**.
3. To start the proxy wizard, click **Create New**.
4. In the **Reverse proxy** box, click the **Use OpenAPI Spec** link.

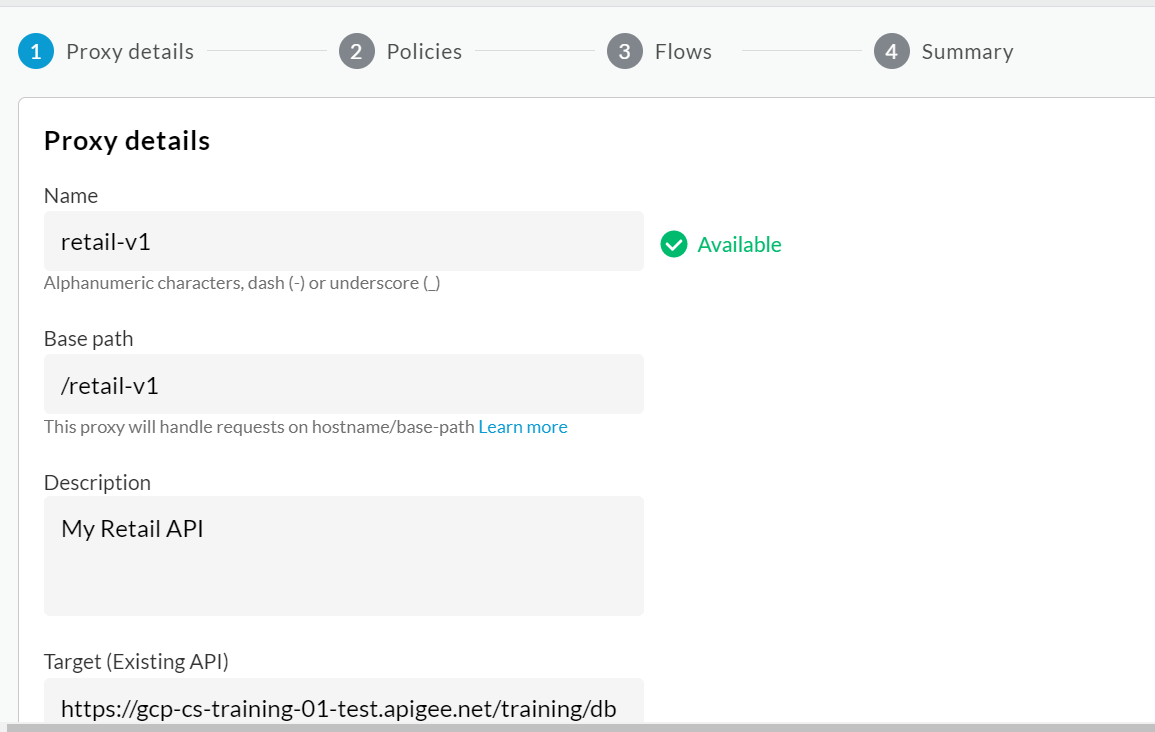
**Note:**Confirm that you are using the link in the "Reverse proxy" box, not the "No target" box.

1. Specify the following URL for the **OpenAPI Spec URL**:

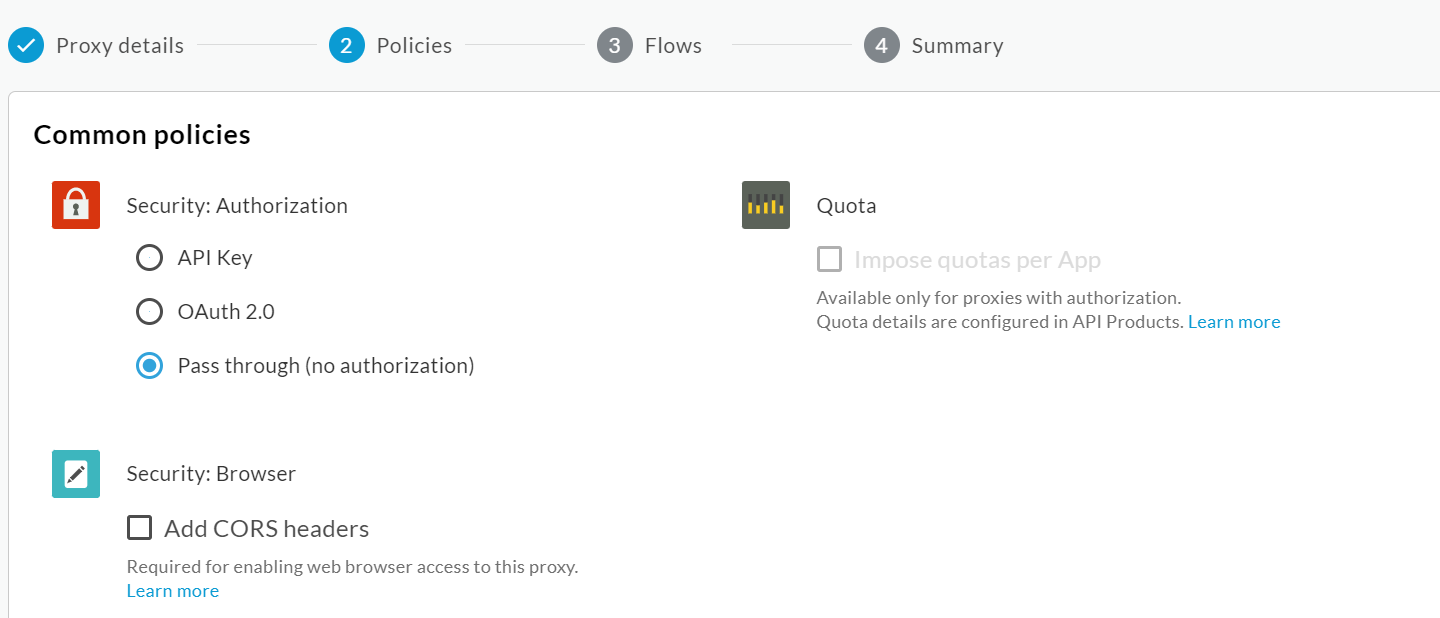
<https://storage.googleapis.com/cloud-training-cors/developing-apis/specs/retail-backend.yaml>

****

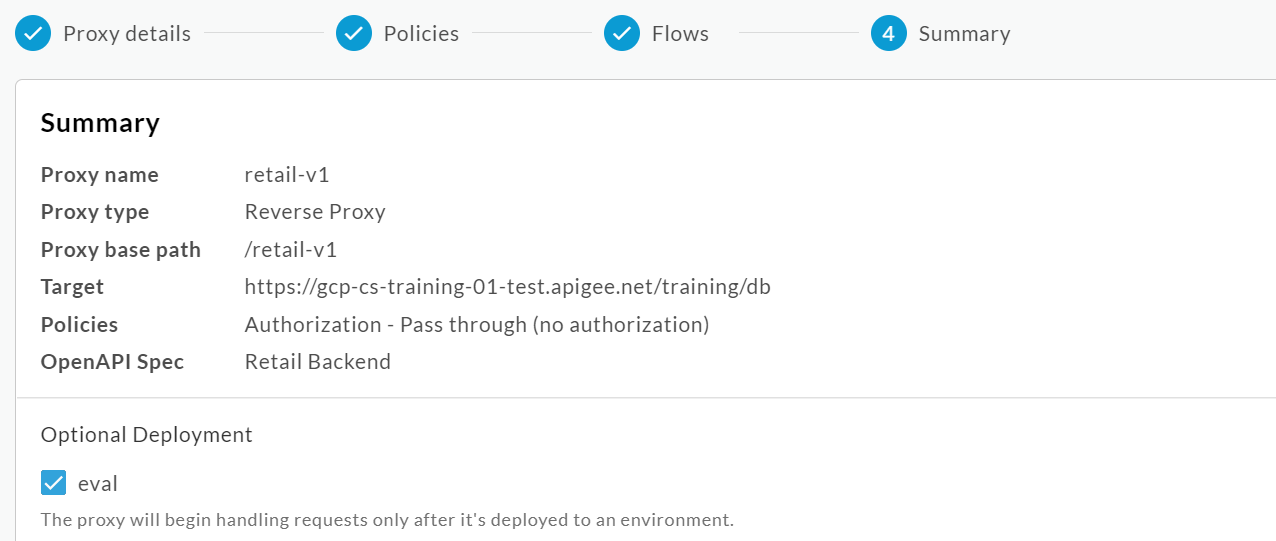
1. Click **Select**.
2. Specify the following for the **Proxy details**:

****

1. The target was taken from the **server's** array in the OpenAPI spec. Leave the target unchanged.
2. **Note:**Confirm that you are using "/retail/v1" for the base path, and not "/retail-v1".
3. Click **Next**.
4. Leave the Common Policies settings as their defaults. Click **Next**.

****

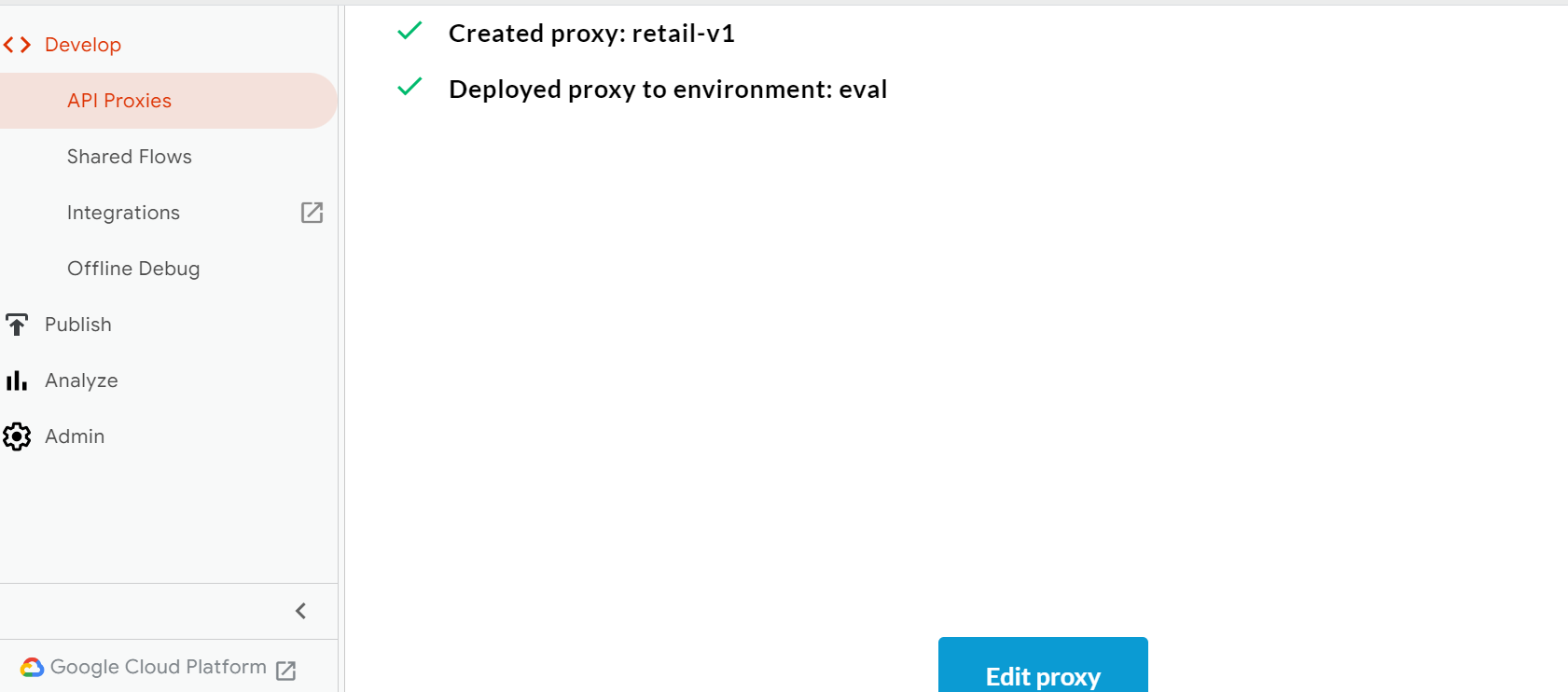
1. Operations that were found in the OpenAPI specification are listed. Confirm that all operations are selected and click **Next**.
2. On the Summary page, select the **eval** environment.

****

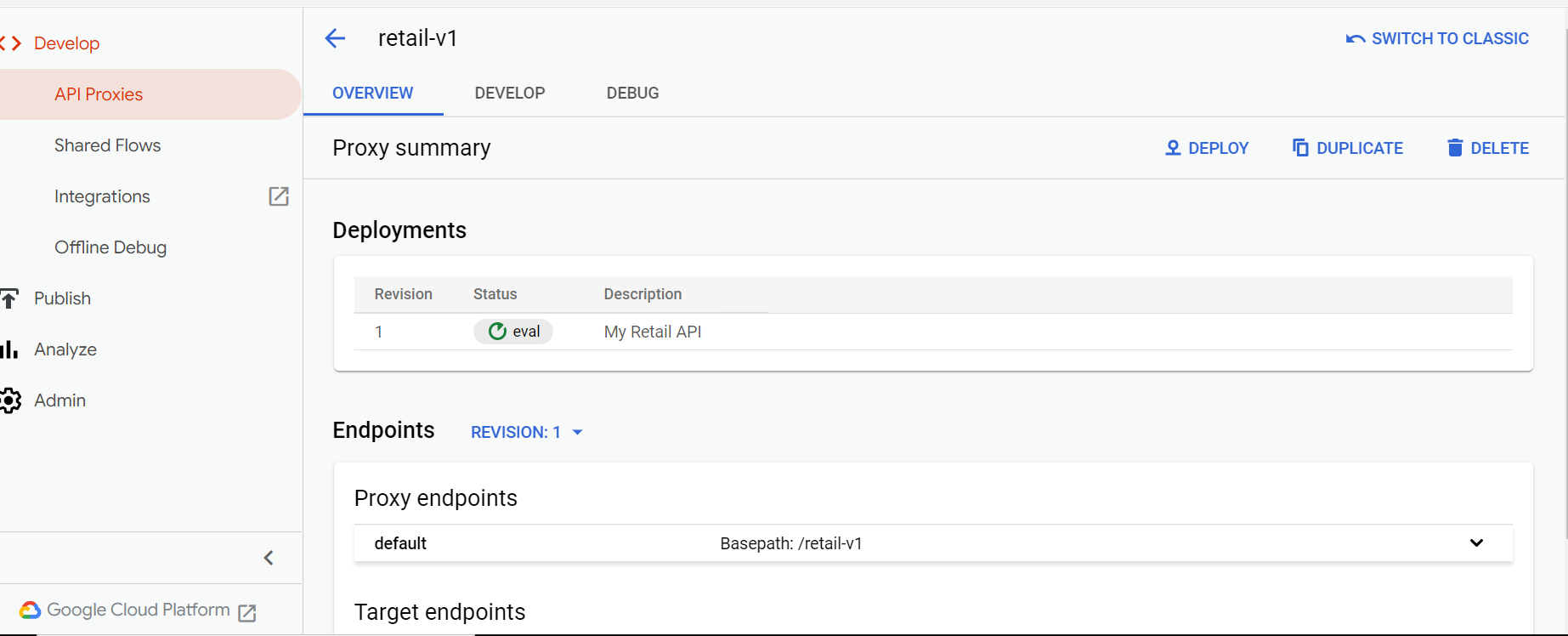
1. Click **Create and deploy**.

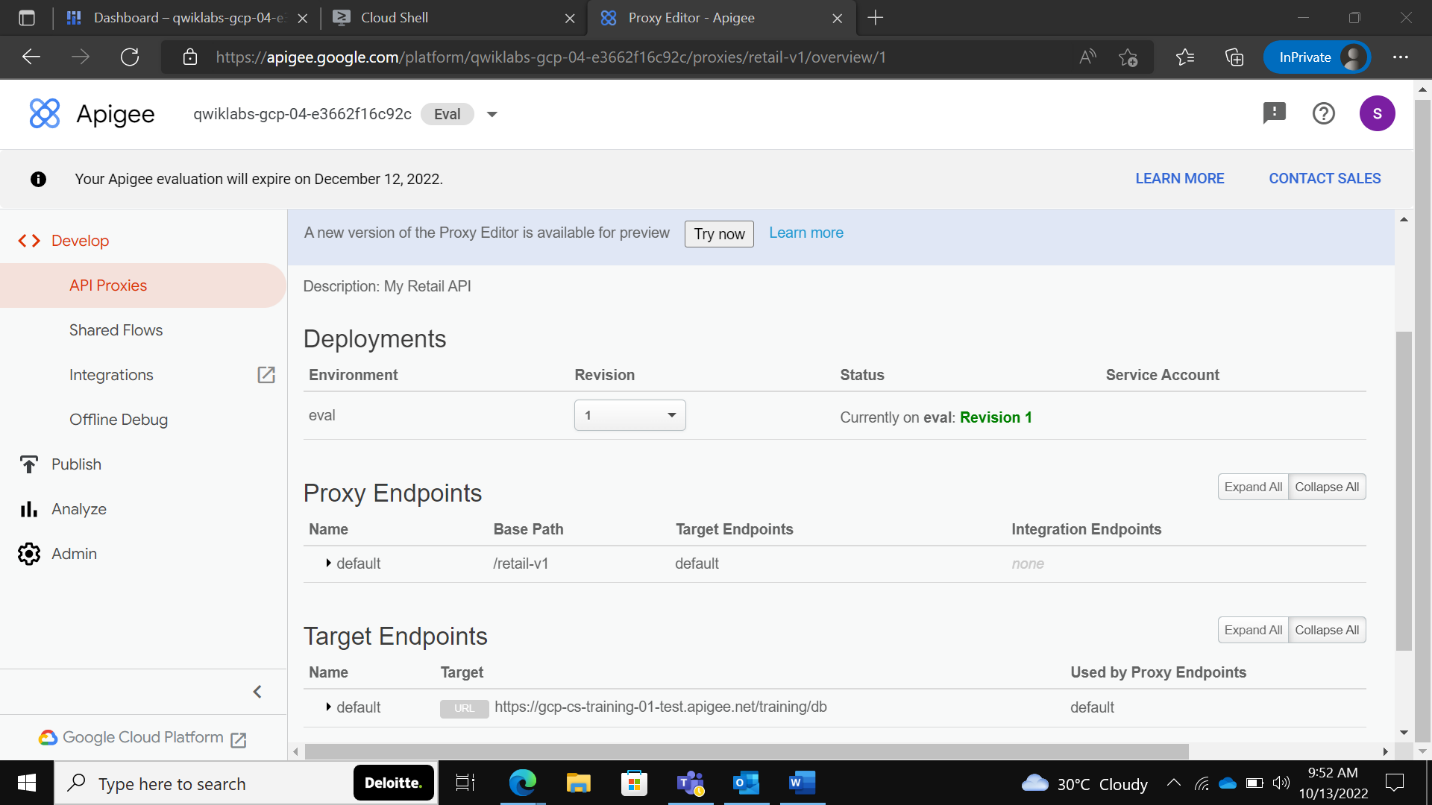
Your proxy will be generated and marked for deployment.

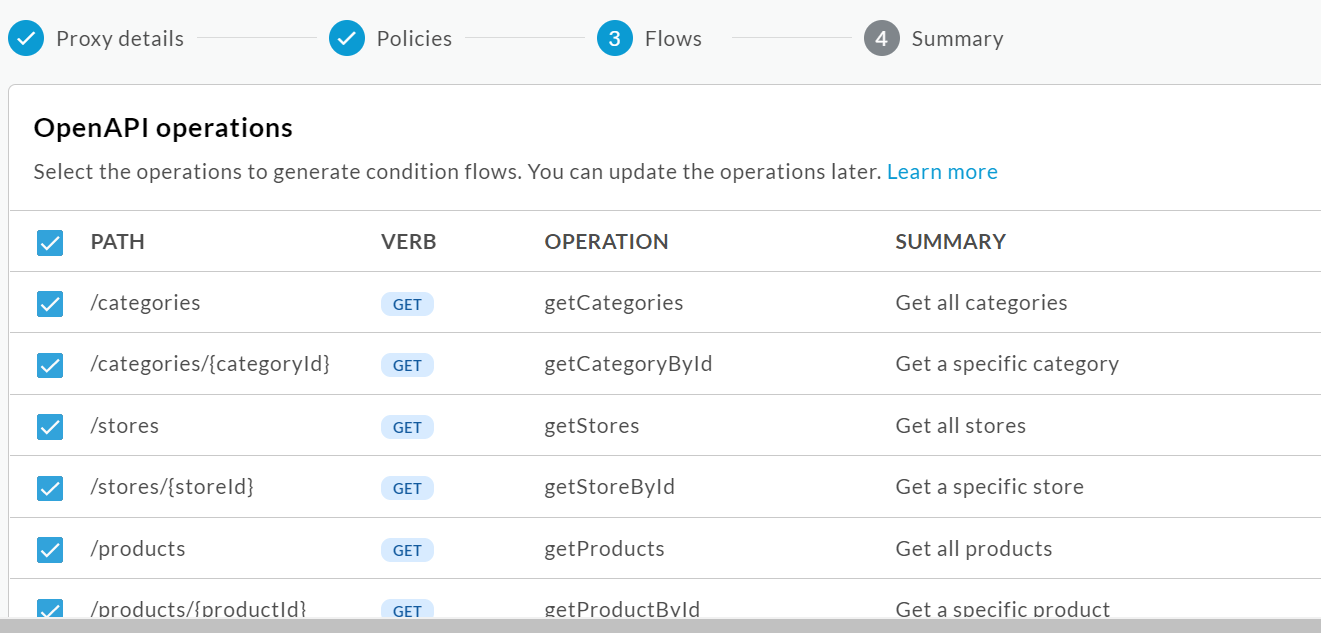
1. Click **Edit proxy**.

****

1. If the **Switch to Classic** link exists in the upper right corner, click that link.

****

****

****

**Check Provisioning dashboard**

1. In the Google Cloud Console, navigate to **Compute Engine >** **VM (Virtual Machine) instances**.
2. Click on the **External IP** for the **lab-startup** VM.

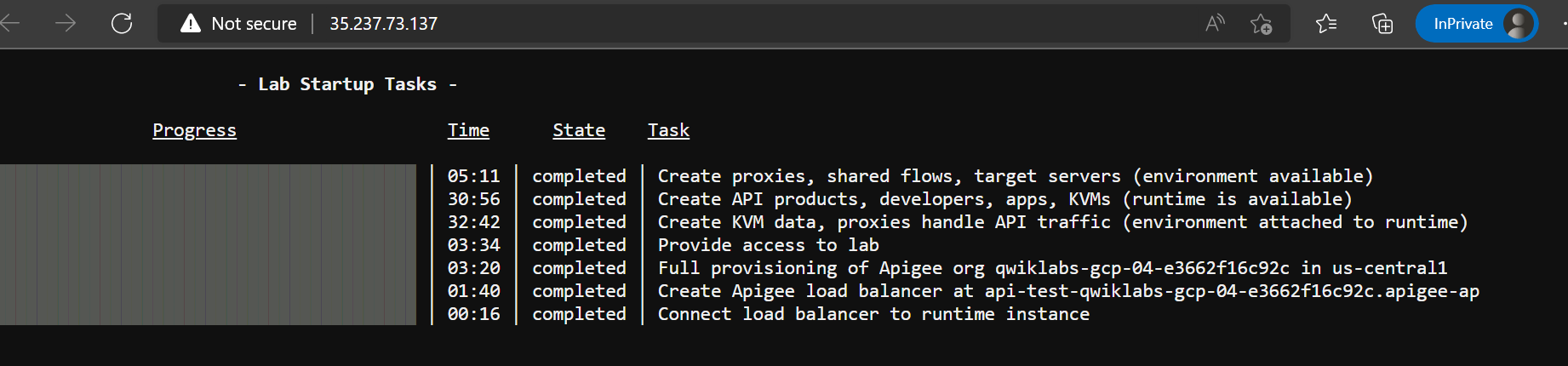
**A screenshot of a computer

Description automatically generated**

If you see a redirect notice page, click the link to the external IP address.

A new browser window will open. Lab startup tasks are shown with their progress.

* *Create proxies, shared flows, target servers* should be complete, allowing you to use the Apigee console for tasks like proxy editing.
* *Create API products, developers, apps, KVMs* indicates when the runtime is available, and those assets may be saved.
* *Create KVM data, proxies handle API traffic* indicates when the eval environment has been attached to the runtime and the deployed proxies can take runtime traffic.

****

**Explore and trace the API proxy**

1. Click the **Develop** tab.

This tab is used to edit the proxy that has been generated. Conditional flows have been created for each of the operations in the OpenAPI specification. These conditional flows appear in the proxy endpoint in the Navigator on the left. When you click on a conditional flow, it is selected in the **Code** pane on the right.

**A screenshot of a computer

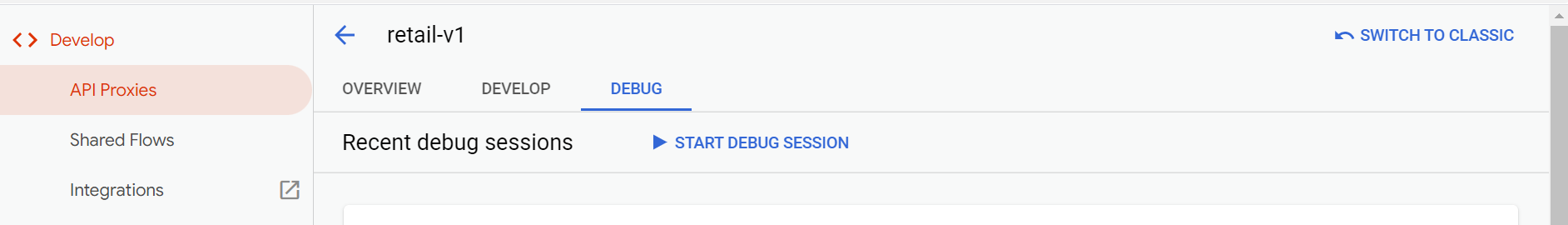
Description automatically generated**

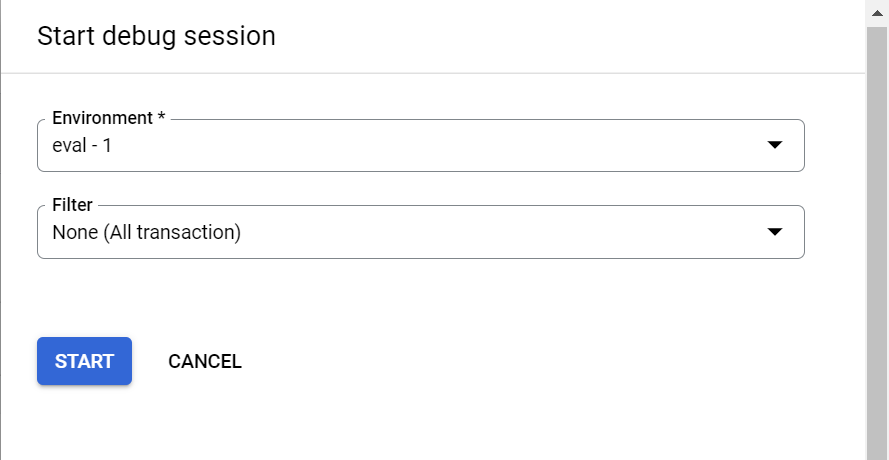
1. Click the **Debug** tab. The debug tool is used to trace API requests that are handled by the proxy.
2. In the **Start a debug session** pane, on the environment (Env.) dropdown, select **eval**.

The deployed revision number will also show in the dropdown.

1. Click **Start Debug Session**.

The debug session will run for 10 minutes.

****

****

1. To send a "get categories" request to your proxy, replace the URL in the **Send Requests** pane with:

<https://api-test-ORG.apigee-apijam.dev/retail/v1/categories>

1. Replace **ORG** in the URL with your Qwiklabs project ID. The project ID is also the name of your Apigee organization.
2. Click **Send**.

**Graphical user interface, text

Description automatically generated**