



Sistemas Distribuidos - Podcast

Neste trabalho nós iremos criar um Bucket no Google Cloud Storage associado ao gerenciamento de podcasts por uma RESTful API.

Vídeo de apresentação, disponível em: [YouTube](#)

APIs

As APIs são conjuntos de métodos que permitem que programas se comuniquem de forma eficaz seguindo um protocolo claro para transferência e interpretação de dados. Elas utilizam a Internet como canal de comunicação padrão, onde as solicitações e respostas entre programas são transmitidas. As APIs web seguem a arquitetura cliente-servidor, em que o cliente, um dispositivo de computação, solicita recursos ou dados, e o servidor, que contém esses recursos, interpreta e atende às solicitações do cliente.

HTTP

As APIs usam a Web como canal de comunicação e muitas delas seguem o protocolo HTTP, que estabelece regras e métodos para trocar dados entre clientes e servidores pela Internet. Aquelas que utilizam o protocolo HTTP empregam métodos de solicitação HTTP, para enviar pedidos do cliente aos servidores. Os métodos HTTP mais comuns

incluem GET (para buscar dados do servidor), PUT (para atualizar ou criar dados), POST (principalmente para criar novos recursos) e DELETE (para remover dados ou recursos especificados pelo cliente no servidor).

Endpoints

As APIs utilizam os métodos HTTP para interagir com dados ou serviços hospedados em um servidor. No entanto, para que essa interação ocorra de forma consistente, é necessário um meio de acesso aos recursos de maneira organizada. As APIs fazem uso de canais de comunicação chamados de endpoints, que permitem aos clientes acessar recursos de forma simples e consistente. Esses endpoints são pontos de acesso a dados ou recursos no servidor, apresentados na forma de um URI HTTP. Eles são anexados ao URL base da API, criando um caminho para recursos específicos ou contêineres de recursos. Além disso, é possível adicionar strings de consulta aos endpoints para passar variáveis que possam ser necessárias para concluir uma solicitação na API.

Cloud API Storage

O Google disponibiliza uma variedade de APIs que podem ser aplicadas em diversos campos e setores. Essas APIs são amplamente utilizadas em diversas áreas, como desenvolvimento web, aprendizado de máquina, ciência de dados e administração de sistemas, entre outros. No entanto, esses são apenas exemplos de suas possíveis aplicações.

Passo a passo

No Cloud Shell, execute o seguinte comando para criar e editar um arquivo chamado values.json:

```
nano values.json
```

No editor de texto nano, copie e cole o seguinte. Como o bucket deve ter um nome de bucket exclusivo, o ID do projeto é usado no nome do bucket:

```
{
  "name": "qwiklabs-gcp-03-ba96171b9a13",
  "location": "us",
  "storageClass": "multi_regional"
}
```

OAuth

Abra o playground do OAuth 2.0 em uma nova guia.

Selecione API de armazenamento em nuvem V1.

Selecione o escopo https://www.googleapis.com/auth/devstorage.full_control.

Clique na caixa azul que diz Authorize APIs (Autorizar APIs). Isso abre a página de login do Google.

Selecione seu nome de usuário e clique em permitir quando as permissões forem solicitadas.

Clique em Exchange authorization code for tokens (Trocar código de autorização para tokens).

Copie o token de acesso para usá-lo na próxima etapa.

Bucket

No prompt CLI, digite ls e pressione Enter. Você deverá ver o arquivo values.json que criou anteriormente e um arquivo README-cloudshell.txt:

Execute o seguinte comando para definir seu token OAuth2 como uma variável de ambiente, substituindo <YOUR_TOKEN> pelo token de acesso que você gerou:

```
export OAUTH2_TOKEN=<YOUR_TOKEN>
```

Execute o seguinte comando para definir o ID do projeto como uma variável de ambiente:

```
export PROJECT_ID=$(gcloud config get-value project)
```

Agora, execute o seguinte comando para criar um bucket do Cloud Storage:

```
curl -X POST --data-binary @values.json \  
-H "Authorization: Bearer $OAUTH2_TOKEN" \  
-H "Content-Type: application/json" \  
"https://www.googleapis.com/storage/v1/b?project=$PROJECT_ID"
```

Pronto, o Bucket foi criado, agora é hora de fazer as requisições e mostrar as funcionalidades.

Salve os áudios dos podcasts disponíveis no repositório em seu computador.

Em sua sessão do Cloud Shell, clique no ícone de menu com três pontos no canto superior direito. Clique em Upload > Choose File (Carregar > Escolher arquivo). Selecione e carregue os arquivos dos podcasts.

Execute o seguinte comando para obter o caminho para o arquivo de imagem:

```
realpath demo-podcast.mp3
```

Defina o caminho do arquivo como uma variável de ambiente executando o seguinte comando, substituindo <DEMO_IMAGE_PATH> pela saída do comando anterior:

```
export OBJECT=<DEMO_IMAGE_PATH>
```

Defina o nome do seu bucket como uma variável de ambiente executando o seguinte comando:

```
export BUCKET_NAME=Project_ID-bucket
```

Agora, execute o seguinte comando para carregar o podcast de demonstração em seu bucket do Cloud Storage:

```
curl -X POST --data-binary @$OBJECT \  
-H "Authorization: Bearer $OAUTH2_TOKEN" \  
-H "Content-Type: audio/mpeg3" \  
"https://www.googleapis.com/upload/storage/v1/b/$BUCKET_NAME/o?uploadType=media&name=pod1"
```

Para ver a imagem que foi adicionada ao seu compartimento, abra o menu de navegação e selecione Cloud Storage > Compartimentos.

Agora, na página principal é possível filtrar, agrupar e classificar os podcasts de acordo com o que foi solicitado.