

Integrate NodeJS with Gemini API's.

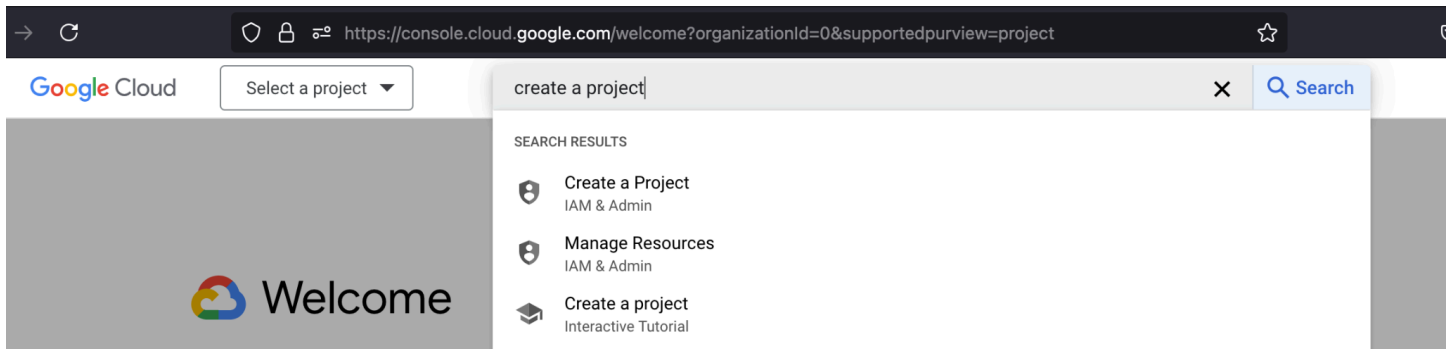
Source: <https://ai.google.dev/gemini-api/docs/text-generation?lang=node>

Get Gemini API Key

Source: <https://aistudio.google.com/>

How to Create a Project in Google

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



Generate text using the Gemini API

The simplest way to generate text using the Gemini API is to provide the model with a single text-only input, as shown in this example:

```
const gemini_api_key = process.env.API_KEY;
const googleAI = new GoogleGenerativeAI(gemini_api_key);
const geminiModel = googleAI.getGenerativeModel({
  model: "gemini-1.5-flash",
});

var question = "what is the value of pie in maths ?";
const generate = async (question) => {
  try {
    const prompt = question;
    const result = await geminiModel.generateContent(prompt);
    const response = result.response;
    console.log(response.text());
  }
}
```

```

    return response.text();
  } catch (error) {
    console.log("response error", error);
  }
};

app.post('/api/content', async (req, res) => {
  let data = req.body.question;
  var result = await generate(data);
  console.log(result);
  res.json({"result" : result});
})

```

- Dependencies we need in our projects

```

const { GoogleGenerativeAI } = require("@google/generative-ai");
require('dotenv').config();
const express = require('express')
const axios = require('axios')
const multer = require('multer');
const bodyParser = require('body-parser');
const fs = require('fs')
const path = require('path');
const app = express()
const port = process.env.PORT || 3000
app.use(bodyParser.json());
app.use(express.json())
app.get('/', (req, res) => {
  res.send('Hello World!')
})

```