Google Gemini API Reference Guide

What is the Gemini API?

The Gemini API allows you to send text prompts to Google's Gemini AI models and receive generated responses. It's a powerful tool for building applications that can understand and generate human-like text, answer questions, summarize content, create quizzes, and much more.

Getting an API Key

1. Go to Google AI Studio
2. Sign in with your Google account.
3. Click **Get API key** in the left sidebar.
4. Click **Create API key** – you can use an existing project or create a new one.
5. Copy the generated key. **Keep it secret** – don't share it or commit it to public repositories.

API Endpoint

The base URL for the Gemini API is:

```
https://generativelanguage.googleapis.com
```

To generate content, you'll use the following endpoint structure:

```
https://generativelanguage.googleapis.com/v1beta/models/{MODEL_NAME}:{ACTION}?key={YOUR_API_KEY}
```

- `{MODEL_NAME}` – The model you want to use. For this lesson, we use `gemini-2.5-flash` (fast, free-tier friendly).
- `{ACTION}` – The action to perform. For text generation, it's `generateContent`.
- `{YOUR_API_KEY}` – Your actual API key, passed as a query parameter.

Example endpoint:

```
https://generativelanguage.googleapis.com/v1beta/models/gemini-2.5-flash:generateContent?key=AIz
```

Request Format

You send a **POST** request with a JSON body. The body must contain a `contents` array, which holds the parts of the conversation. For a simple prompt, it looks like this:

```
{
  "contents": [
    {
      "parts": [
        {
          "text": "Your prompt here"
        }
      ]
    }
  ]
}
```

In Python, using the `requests` library:

```
body = {
    "contents": [
        {
            "parts": [
                {"text": "Explain Python lists in one sentence."}
            ]
        }
    ]
}

response = requests.post(url, json=body, timeout=30)
```

Response Format

A successful response will have a `200` status code. The generated text is nested inside the JSON:

```
{
  "candidates": [
    {
      "content": {
        "parts": [
          {
            "text": "A list in Python is an ordered, mutable collection that can hold items of d
          }
        ]
      }
    }
  ]
}
```


To extract the text in Python:

```
data = response.json()
text = data["candidates"][0]["content"]["parts"][0]["text"]
```

Error Handling

Always check the status code of the response. Common errors:

Status Code	Meaning	Possible Fix
400	Bad request – malformed JSON	Check your request body structure
403	Forbidden – invalid or missing key	Verify your API key
429	Too many requests – rate limit	Wait and try again, reduce frequency
500	Internal server error	Try again later

In your code, handle exceptions and non-200 responses:

```
try:
    response = requests.post(url, json=body, timeout=30)
    if response.status_code != 200:
        print(f"API error: {response.status_code} - {response.text}")
        return None
    # ... process response ...
except Exception as e:
    print("An error occurred:", e)
```

Prompting Tips

- **Be specific and clear** – Tell the model exactly what you want.
- **Request structured output** – Ask for JSON, CSV, or a specific format. Example:

  ```
  Respond with ONLY a JSON array in this format: [{"question": "...", "answer": "..."}]
  ```

- **Provide examples** – Few-shot prompting can improve accuracy.
- **Set a timeout** – Always include a timeout (e.g., 30 seconds) to avoid hanging.

Rate Limits and Quotas

The free tier typically allows:

- **60 requests per minute**
- **1,500 requests per day**

These limits may vary, so check the official documentation for the latest information.

Minimal Working Example

```python
import requests

API_KEY = "YOUR_API_KEY"
url = f"https://generativelanguage.googleapis.com/v1beta/models/gemini-2.5-flash:generateContent

body = {
    "contents": [{"parts": [{"text": "Say hello in 5 words or less"}]}]
}

response = requests.post(url, json=body, timeout=30)
data = response.json()
text = data["candidates"][0]["content"]["parts"][0]["text"]
print(text)
```

Additional Resources

- Official Gemini API Documentation
- Google AI Studio – test prompts interactively
- Python `requests` library documentation