

第一部分: 标准作业程序 (SOP) - 调用自部署 Gemini API

1. 目的

本SOP旨在标准化通过自定义API端点 (<https://api.teaforia.in>) 调用 Gemini 2.5 Pro/Flash 模型多模态功能的工作流程, 确保请求格式正确, 功能调用成功。

2. 核心要素

- **API 基础地址 (Base URL):** <https://api.teaforia.in>
- **认证凭证 (API Key):** gyhongyu (您的专属口令)
- **认证方式 (Authentication):** HTTP POST 请求 + x-goog-api-key 请求头。

3. API 请求的核心结构

所有对 Gemini 模型的调用都使用同一个端点:

`/v1beta/models/{model_name}:generateContent`。请求成功的关键在于构建正确的 请求体 (**Request Body**)。

请求体核心是一个名为 `contents` 的数组, 它包含了您要发送给模型的所有信息。

3.1 纯文本请求 (Text-only)

这是最简单的形式, `contents` 数组中只有一个对象, 该对象包含一个文本部分 (`parts`)。

```
{
  "contents": [
    {
      "parts": [
        { "text": "你好, 请你做个自我介绍。" }
      ]
    }
  ]
}
```

3.2 多模态请求 (图像或文件)

要发送图片或文件, 您需要做两件事:

1. 将文件转换为 **Base64** 编码: 这是将二进制文件(如图片、PDF)转换为文本字符串的标准方法, 以便能在JSON中传输。
2. 在 **parts** 数组中添加文件对象: 除了文本部分, 再增加一个 `inlineData` 对象, 其中包含文件的 `contentType` 和 Base64 编码后的数据。

请求结构 (文本 + 图像):

```
{
  "contents": [
    {
      "parts": [
        { "text": "请提取这张图片中的所有文字。" },
        {
          "inlineData": {
            "mimeType": "image/jpeg",
            "data": "iVBORwOKGgoAAAANSUhEUgA..."
          }
        }
      ]
    }
  ]
}
```

请求结构 (文本 + PDF文件):

```
{
  "contents": [
    {
      "parts": [
        { "text": "请总结这份PDF文档的核心内容。" },
        {
          "inlineData": {
            "mimeType": "application/pdf",
            "data": "JVBERiOxLjQKJ..."
          }
        }
      ]
    }
  ]
}
```

支持的 **mimeType** 包括: image/png, image/jpeg, application/pdf, text/plain,

text/x-python 等。

4. 操作流程

1. 准备数据:
 - 文本: 准备好您的提示词 (Prompt)。
 - 图像/文件: 准备好要上传的文件。
2. 构建请求:
 - 如果是图像/文件, 使用程序 (如下方HTML代码中的JavaScript) 将文件读取为 Base64字符串。
 - 根据 **3. API请求的核心结构** 中的模板, 构建包含文本和/或文件数据的JSON请求体。
3. 发送请求:
 - 向 <https://api.teaforia.in/v1beta/models/{您选择的模型}:generateContent> 发送一个 POST 请求。
 - 在请求头中必须包含:
 - Content-Type: application/json
 - x-goog-api-key: gyhongyu
 - 将构建好的JSON作为请求体发送。
4. 处理响应:
 - 解析返回的JSON数据, 提取 `candidates[0].content.parts[0].text` 的内容, 即为模型的回答。