



Gemini on Android : 打开移动设备的AI新世界

叶楠

谷歌开发者专家 (Android方向)



分享内容

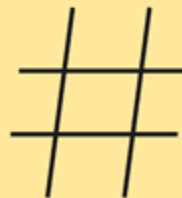
- （1） 什么是Gemini
- （2） Gemini在Android上的使用
- （3） Gemini在云端
- （4） Gemini在本地
- （5） Gemini应用 – 简小助

Mobile
@DevFest



Google
Developer
Groups

什么是Gemini?

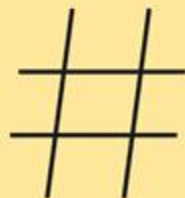


Gemini

Google 推出的大语言模型。旨在增强自然语言理解和生成的能力，结合了深度学习和多模态技术，使其不仅能够处理文本，还可以分析图像、视频等内容。



Google
Developer
Groups



Mobile
@DevFest

能用Gemini做什么？

内容总结

可以从大量信息中提取关键点，为用户提供精简的摘要。例如，它可以对长篇文章、报告、研究文献等进行高效总结，让用户快速了解主要内容。

内容生成

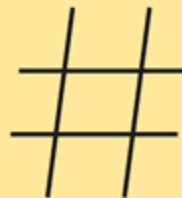
以基于输入内容生成各种类型的文本，包括对话、故事、文章、代码等。这种生成能力帮助用户在创作、内容填充以及想法拓展方面提高效率。

Mobile
@DevFest



Google
Developer
Groups

Gemini在 Android上的使用



Gemini云端API



Google AI Client SDK

帮助开发者在各种程序中集成 Google 的生成式 AI 模型，特别是 Gemini 系列模型。

Python

Node

Go

REST

Dart

Android

Swift

Web



Firebase SDK

为移动应用设计的开发工具包，用于在客户端中集成 Google 的生成式 AI 模型，特别是 Gemini 系列模型。

Dart

Android

Swift

Web

Gemini设备端API



Google AI Edge SDK

帮助开发者设备端本地运行Google生成式AI模型，主要是在Android设备上通过AI Core访问Gemini Nano模型

Pixel 9

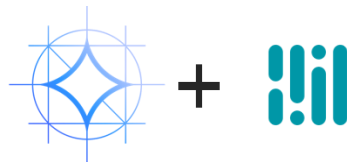
Pixel 8

Samsung

Motorola

Realme

Xiaomi



MediaPipe LLM

旨在帮助开发者在设备端运行大型语言模型（LLM），实现文本生成、信息检索和文档摘要等任务

Android

iOS

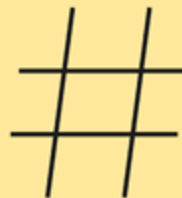
Web

Mobile
@DevFest



Google
Developer
Groups

Gemini在云端



Gemini云端API



Google AI Client SDK

帮助开发者在各种程序中集成 Google 的生成式 AI 模型，特别是 Gemini 系列模型。

Python

Node

Go

REST

Dart

Android

Swift

Web



Firebase SDK

为移动应用设计的开发工具包，用于在客户端中集成 Google 的生成式 AI 模型，特别是 Gemini 系列模型。

Dart

Android

Swift

Web

Google AI Client SDK使用清单

(1)

生成API密钥

(2)

测试你的提示词

(3)

添加Gradle依赖项

(4)

创建模型

(5)

生成内容

API keys

Cloud projects are subject to the [Google Cloud Platform Terms of Service](#), and use of Gemini API and Google AI Studio is subject to the [Gemini API Additional Terms of Service](#).

Remember to use API keys securely. Don't share or embed them in public code. Use of Gemini API from a billing-enabled project is subject to [pay-as-you-go pricing](#).

Quickly test the API by running a cURL command

API quickstart guide

```
curl \
-H "Content-Type: application/json" \
-d '{"contents":[{"parts":[{"text":"Explain how AI works"}]}]}' \
-X POST "https://generativelanguage.googleapis.com/v1beta/models/gemini-1.5-flash-latest:generateContent?key=YOUR_API_KEY"
```

Use code with caution.

Create API key

Your API keys are listed below. You can also view and manage your project and API keys in Google Cloud.

Project number	Project name	API key	Created	Plan
...0343	your1024 🔗	...GHUM	Oct 23, 2024	<div>Paid</div> <div>Go to billing</div> <div>View usage data</div> <div>🗑️</div>
...6305	Gemini API 🔗	...pU7M	Oct 16, 2024	<div>Paid</div> <div>Go to billing</div> <div>View usage data</div> <div>🗑️</div>

Google AI Client SDK使用清单

(1)

生成API秘钥

(2)

测试你的提示词

(3)

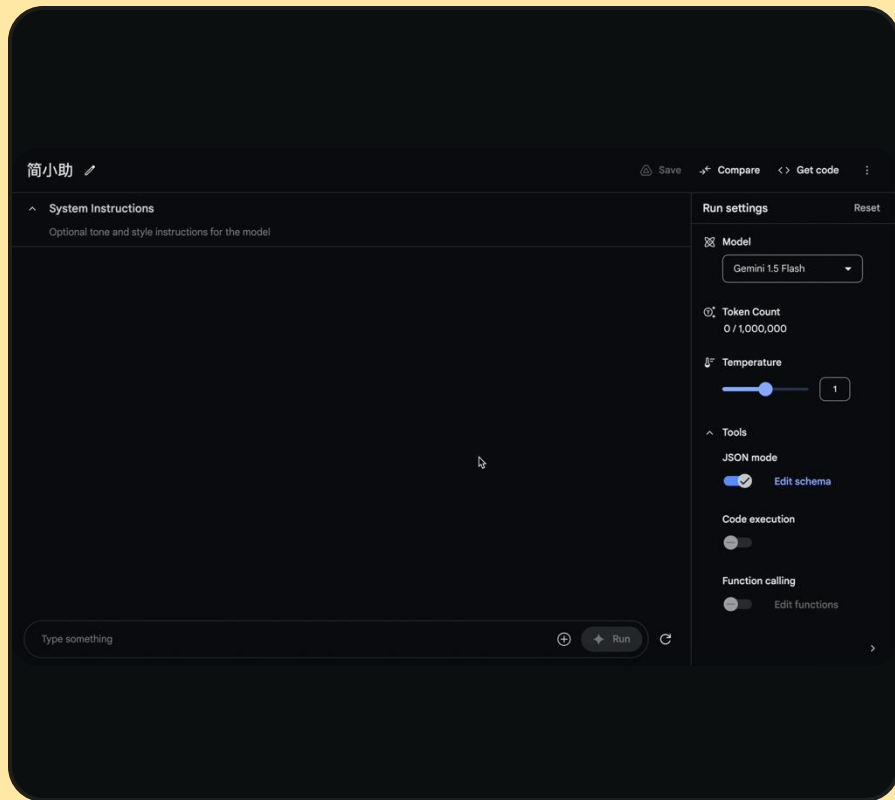
添加Gradle依赖项

(4)

创建模型

(5)

生成内容



Google AI Client SDK使用清单

(1)

生成API秘钥

(2)

测试你的提示词

(3)

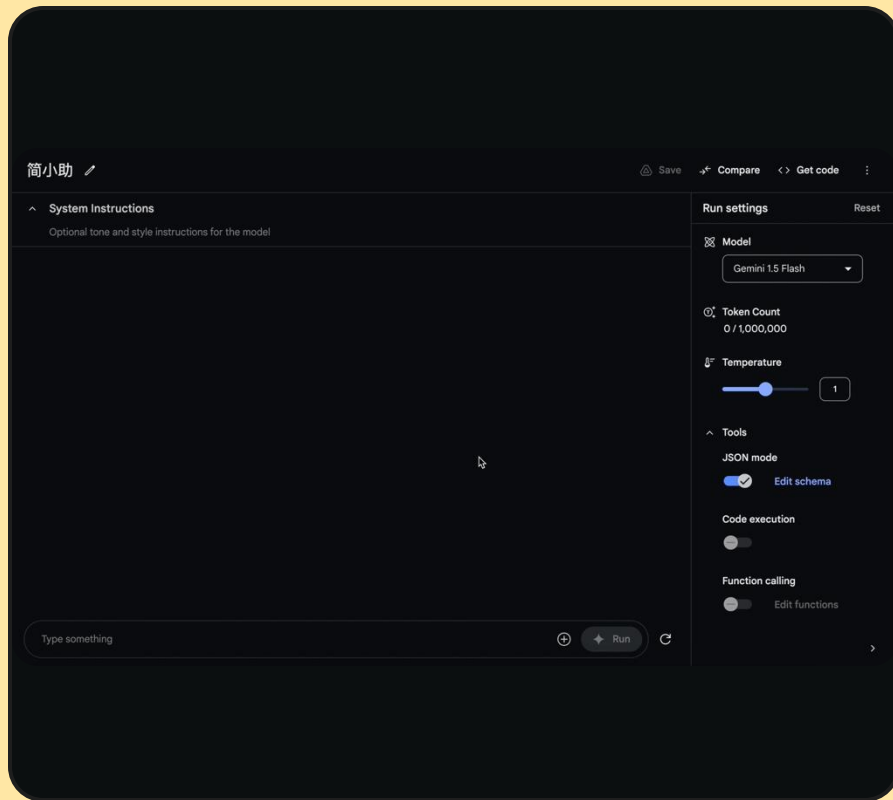
添加Gradle依赖项

(4)

创建模型

(5)

生成内容



Google AI Client SDK使用清单

(1)

生成API秘钥

(2)

测试你的提示词

(3)

添加Gradle依赖项

(4)

创建模型

(5)

生成内容

```
dependencies {  
    ...  
    implementation (  
        group: 'com.google.ai.client.generativeai',  
        name: 'generativeai',  
        version: '0.9.0'  
    )  
    ...  
}
```

Google AI Client SDK使用清单

(1)

生成API秘钥

(2)

测试你的提示词

(3)

添加Gradle依赖项

(4)

创建模型

(5)

生成内容

```
model = GenerativeModel(  
    modelName = 'gemini-1.5-flash-001',  
    apiKey = BuildConfig.GEMINI_API_KEY,  
    generationConfig = generationConfig {  
        temperature = 0.15  
        topK = 32  
        topP = 1f  
        maxOutputTokens = 8192  
    }  
)
```

Secrets Gradle 插件

Google AI Client SDK使用清单

(1)

生成API密钥

(2)

测试你的提示词

(3)

添加Gradle依赖项

(4)

创建模型

(5)

生成内容

```
/* 同步生成内容 */
model.generateContent (
    content {
        image(bitmap)
        blob(blob, stream.readBytes())
        text(prompt)
    }
).text

/* 流式生成内容 */
model.generateContentStream(
    content {
        image(bitmap)
        blob(blob, stream.readBytes())
        text(prompt)
    }
).collect{ ... }
```



Google AI Client SDK使用清单

(1)

生成API密钥

(2)

测试你的提示词

(3)

添加Gradle依赖项

(4)

创建模型

(5)

生成内容

```
/* 同步生成内容 */
model.generateContent (
    content {
        image(bitmap)
        blob(blob, stream.readBytes())
        text(prompt)
    }
).text

/* 流式生成内容 */
model.generateContentStream(
    content {
        image(bitmap)
        blob(blob, stream.readBytes())
        text(prompt)
    }
).collect{ ... }
```



Gemini云端API



Google AI Client SDK

帮助开发者在各种程序中集成 Google 的生成式 AI 模型，特别是 Gemini 系列模型。

Python

Node

Go

REST

Dart

Android

Swift

Web



Firebase SDK

为移动应用设计的开发工具包，用于在客户端中集成 Google 的生成式 AI 模型，特别是 Gemini 系列模型。

Dart

Android

Swift

Web

Firebase SDK使用清单

(1)

项目绑定Vertex AI

(2)

确保API已经启用

(3)

测试你的提示词

(4)

添加Gradle依赖项

(5)

创建模型



Vertex AI in Firebase

使用 Gemini API 和客户端 SDK 构建具备生成式 AI 功能的客户端 Web 应用和移动应用

主要特性:

- 适用于 Swift、Android、Web 和 Flutter 的 SDK
- 使用 Gemini 多模态模型
- Vertex AI Gemini API 需要关联结算账号
- 直接从移动应用和 Web 应用中调用 Vertex AI Gemini API
- 使用 Firebase App Check 帮助保护 Gemini API 调用

[了解详情](#)

[开始使用](#)

Firebase SDK使用清单

(1)

项目绑定Vertex AI

(2)

确保API已经启用

(3)

测试你的提示词

(4)

添加Gradle依赖项

(5)

创建模型



Vertex AI API

[Google Enterprise API](#)

Train high-quality custom machine learning models with minimal machine learning expertise and...

管理

[试用此 API](#)

✓ API 已启用



Vertex AI in Firebase API

[Google](#)

firebasevertexai.googleapis.com API.

管理

✓ API 已启用

Firestore SDK使用清单

(1)

项目绑定Vertex AI

(2)

确保API已经启用

(3)

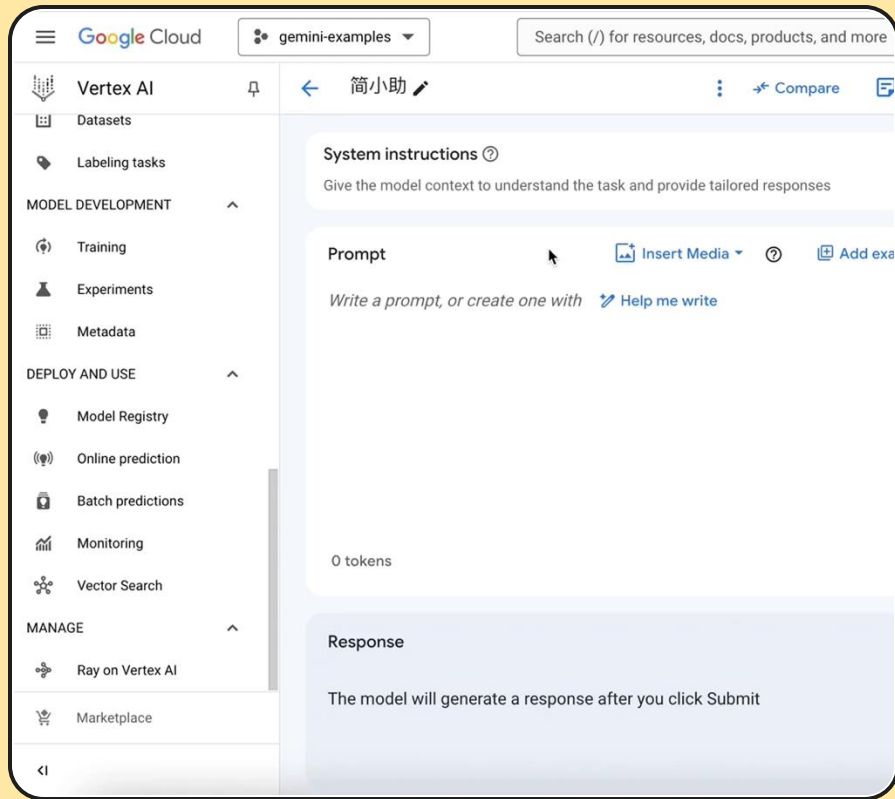
测试你的提示词

(4)

添加Gradle依赖项

(5)

创建模型



Firestore SDK使用清单

- (1) 项目绑定Vertex AI
- (2) 确保API已经启用
- (3) 测试你的提示词
- (4) 添加Gradle依赖项
- (5) 创建模型

```
dependencies {  
    ...  
    implementation (  
        group: 'com.google.firebase',  
        name: 'firebase-vertexai',  
        version: '16.0.1'  
    )  
    ...  
}
```

Firestore SDK使用清单

(1)

项目绑定Vertex AI

(2)

确保API已经启用

(3)

测试你的提示词

(4)

添加Gradle依赖项

(5)

创建模型

```
model = GenerativeModel(  
    modelName = 'gemini-1.5-flash-001',  
    generationConfig = generationConfig {  
        temperature = 0.15  
        topK = 32  
        topP = 1f  
        maxOutputTokens = 8192  
    }  
)
```

Firebase SDK使用清单

(2)

确保API已经启用

(3)

测试你的提示词

(4)

添加Gradle依赖项

(5)

创建模型

(6)

生成内容

```
/* 同步生成内容 */
model.generateContent (
    content {
        image(bitmap)
        inlineData(stream.readBytes(),
            mimeType)
        text(prompt)
    }
).text

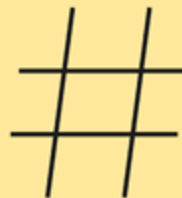
/*流式生成内容 */
model.generateContentStream(
    content {
        image(bitmap)
        inlineData(stream.readBytes(),
            mimeType)
        text(prompt)
    }
).collect{ ... }
```


Mobile
@DevFest



Google
Developer
Groups

Gemini在本地



Gemini设备端API



Google AI Edge SDK

帮助开发者设备端本地运行Google生成式AI模型，主要是在Android设备上通过AI Core访问Gemini Nano模型

Pixel 9

Pixel 8

Samsung

Motorola

Realme

Xiaomi



MediaPipe LLM

旨在帮助开发者在设备端运行大型语言模型（LLM），实现文本生成、信息检索和文档摘要等任务

Android

iOS

Web

Google AI Edge SDK使用清单

(1)

确定运行环境

(2)

安装系统组件

(3)

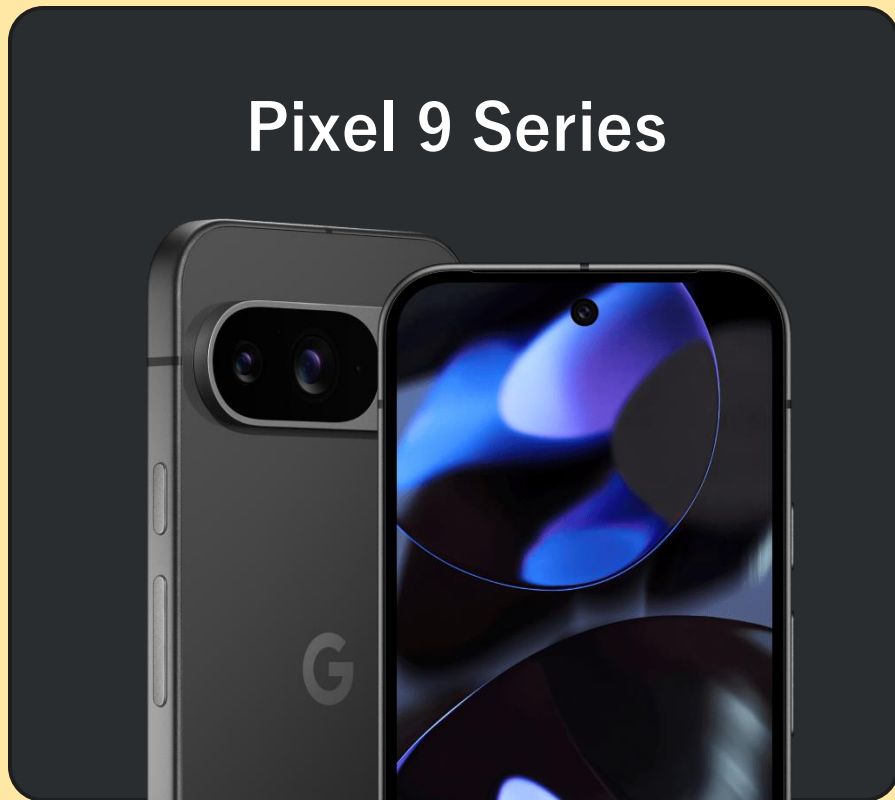
测试你的提示词

(4)

添加Gradle依赖项

(5)

创建模型



Google AI Edge SDK使用清单

(1)

确定运行环境

(2)

安装系统组件

(3)

测试你的提示词

(4)



添加Gradle依赖项

(5)

创建模型



Firebase Device Streaming

	Manufacturer	Name	API	Width	Height	dpi
<input type="checkbox"/>	 Google	Pixel 8	34	1080	2400	420
<input type="checkbox"/>	 Google	Pixel 8 Pro	34	1008	2244	390
<input type="checkbox"/>	 Google	Pixel 8a	34	1080	2400	420
<input type="checkbox"/>	 Google	Pixel 9	34	1080	2424	420
<input type="checkbox"/>	 Google	Pixel 9 Pro	34	960	2142	360
<input type="checkbox"/>	 Google	Pixel 9 Pro Fold	34	2076	2152	390
<input type="checkbox"/>	 Google	Pixel 9 Pro XL	34	1008	2244	360

Google AI Edge SDK使用清单

(1)

确定运行环境

(2)

安装系统组件

(3)

测试你的提示词

(4)

添加Gradle依赖项

(5)

创建模型



Android AI Core
(Beta)

0.thirdparty:ear开头



Private Compute
Services

1.0.release.658389993或者更高

Google AI Edge SDK使用清单

(1)

确定运行环境

(2)

安装系统组件

(3)

测试你的提示词

(4)

添加Gradle依赖项

(5)

创建模型

仅供参考



Google AI Studio



Vertex AI Studio

Google AI Edge SDK使用清单

(1)

确定运行环境

(2)

安装系统组件

(3)

测试你的提示词

(4)

添加Gradle依赖项

(5)

创建模型

```
dependencies {  
    ...  
    implementation (  
        group: 'com.google.ai.edge.aicore',  
        name: 'aicore',  
        version: '0.0.1-exp01'  
    )  
    ...  
}
```

Google AI Edge SDK使用清单

(1)

确定运行环境

(2)

安装系统组件

(3)

测试你的提示词

(4)

添加Gradle依赖项

(5)

创建模型

```
model = GenerativeModel(  
    generationConfig = generationConfig {  
        context = applicationContext  
        temperature = 0.15  
        topK = 32  
        maxOutputTokens = 8192  
    }  
)
```


Google AI Edge SDK使用清单

(2)

安装系统组件

(3)

测试你的提示词

(4)

添加Gradle依赖项

(5)

创建模型

(6)

生成内容

```
/* 同步生成内容 */
model.generateContent(
    content {
        text(prompt)
    }
).text

/* 流式生成内容 */
model.generateContentStream(
    content {
        text(prompt)
    }
).collect{ ... }
```

Google AI Edge SDK使用清单

(3)

测试你的提示词

(4)

添加Gradle依赖项

(5)

创建模型

(6)

生成内容

(7)

关闭会话

```
/* 不关闭会导致AICore问题, 甚至系统崩溃*/  
model.close()
```

Gemini设备端API



Google AI Edge SDK

帮助开发者设备端本地运行Google生成式AI模型，主要是在Android设备上通过AI Core访问Gemini Nano模型

Pixel 9

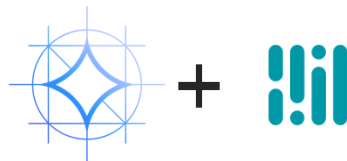
Pixel 8

Samsung

Motorola

Realme

Xiaomi



MediaPipe LLM

旨在帮助开发者在设备端运行大型语言模型（LLM），实现文本生成、信息检索和文档摘要等任务

Android

iOS

Web

Mediapipe LLM使用清单

(1)

下载模型

(2)

测试你的提示词

(3)

添加Gradle依赖项

(4)

添加库引用依赖

(5)

创建模型

- **Gemma-2 2B**: 最新版本的 Gemma 系列模型。是一系列先进的轻量级开放模型的一部分，这些模型采用与 **Gemini** 模型相同的研究成果和技术构建而成。

推荐

- **Gemma 2B**: 是一系列先进的轻量级开放式模型的一部分，其开发采用了与 **Gemini** 模型相同的研究成果和技术。非常适合用于处理各种文本生成任务，包括问答、摘要和推理。
- **Phi-2**: 一个拥有 27 亿参数的 Transformer 模型，最适合问答、聊天和代码格式。
- **Falcon-RW-1B**: 一个参数数为 10 亿的仅解码器因果模型，基于 **RefinedWeb** 的 3500 亿个词元进行训练。
- **StableLM-3B**: 一个拥有 30 亿参数的 decoder-only 语言模型，基于多样化的英语和代码数据集内的 1 万亿个词元进行了预训练。

https://ai.google.dev/edge/mediapipe/solutions/genai/llm_inference/android

Mediapipe LLM使用清单

(1)

下载模型

(2)

测试你的提示词

(3)

添加Gradle依赖项

(4)

添加库引用依赖

(5)

创建模型

仅供参考



Google AI Studio



Vertex AI Studio

Mediapipe LLM使用清单

(1)

下载模型

(2)

测试你的提示词

(3)

添加Gradle依赖项

(4)

添加库引用依赖

(5)

创建模型

```
dependencies {  
    ...  
    implementation (  
        group: 'com.google.mediapipe',  
        name: 'tasks-genai',  
        version: '0.10.16'  
    )  
    ...  
}
```

Mediapipe LLM使用清单

(1)

下载模型

(2)

测试你的提示词

(3)

添加Gradle依赖项

(4)

添加库引用依赖

(5)

创建模型

```
<application>
  <uses-native-library
    android:name="libOpenCL.so"
    android:required="false" />
  <uses-native-library
    android:name="libOpenCL-car.so"
    android:required="false" />
  <uses-native-library
    android:name="libOpenCL-pixel.so"
    android:required="false" />
</application>
```

Mediapipe LLM使用清单

(1)

下载模型

(2)

测试你的提示词

(3)

添加Gradle依赖项

(4)

添加库引用依赖

(5)

创建模型

```
val options = LlmInference
    .LlmInferenceOptions.builder()
    .setModelPath(MODEL_PATH)
    .setTemperature(0.15f)
    .setTopK(32)
    .setResultListener { partial, done ->
        /* 流式生成式，异步获取回复内容 */
    }
    .build()

llmInference = LlmInference
    .createFromOptions(context, options)
```


Mediapipe LLM使用清单

(2)

测试你的提示词

(3)

添加Gradle依赖项

(4)

添加库引用依赖

(5)

创建模型

(6)

生成内容

```
/* 同步生成内容 */  
llmInference.generateResponse(text)  
  
/*流式生成内容 */  
llmInference  
    .generateResponseAsync(text)
```

Mediapipe LLM使用清单

(3)

添加Gradle依赖项

(4)

添加库引用依赖

(5)

创建模型

(6)

生成内容

(7)

关闭会话

```
/* 关闭会话应用会崩溃，不关闭会导致内存泄漏*/  
llmInference.close()
```

目前已知的问题：

<https://github.com/google-ai-edge/mediapipe/issues/5740>

Mobile
@DevFest

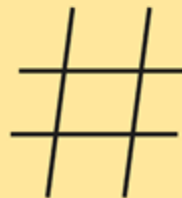


Google
Developer
Groups

Gemini应用



简小助





简小助

借助谷歌的Gemini 技术，简小助可以帮助用户生成个性化的职业画像，提供简历修改建议，面试指导，等等。让用户在竞争激烈的就业市场中脱颖而出。

求职路上的每一步都将更高效、更简单、更自信！

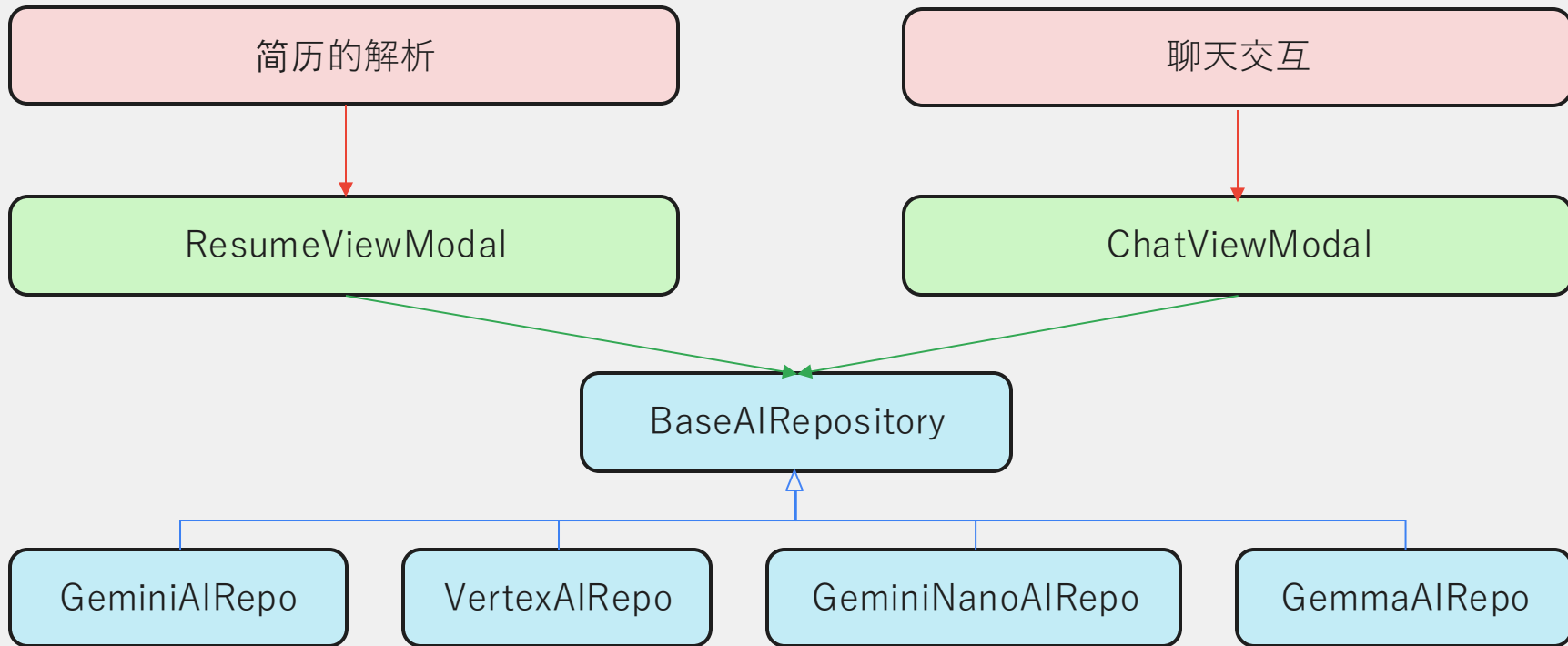
 简小助

让你求职道路上的每一步都更

高效、更简单、更自信！



软件架构



简历的解析

个人信息

姓名: 张三

电话: 13800138000
邮箱: zhangsan@example.com

教育背景

2018.09 - 2022.06 清华大学 计算机科学与技术专业
GPA: 3.8/4.0 主要课程: 数据结构、操作系统、计算机网络、数据库系统原理

工作经历

2022.07 - 2023.06 腾讯科技有限公司 后端开发工程师

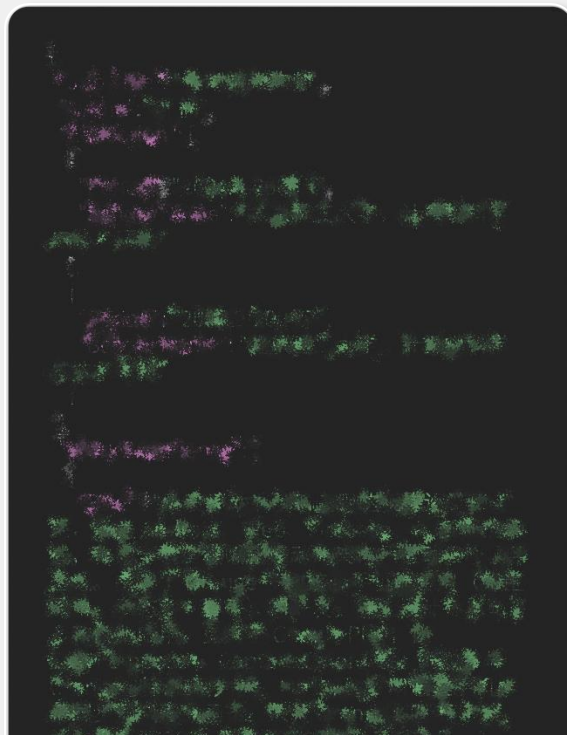
主要职责:

负责后端接口开发与维护, 参与微服务架构改造, 优化系统性能, 提升用户体验。主导了XX模块的设计与开发, 成功上线并稳定运行。

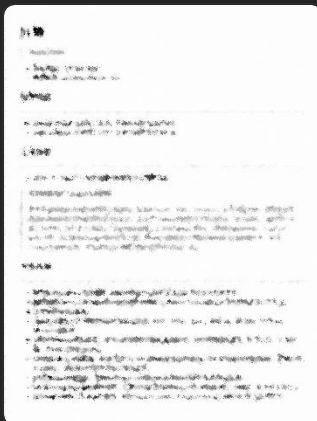
项目经历

2021.03 - 2021.06 校园管理系统升级项目
担任后端开发, 负责用户认证、权限管理等模块。通过引入Spring Security, 提升了系统的安全性。项目获得校级优秀项目称号。

2020.09 - 2020.12 电商促销系统开发
参与秒杀活动的后端开发, 通过Redis缓存与分布式锁, 有效防止了超卖, 保障了活动的顺利进行。

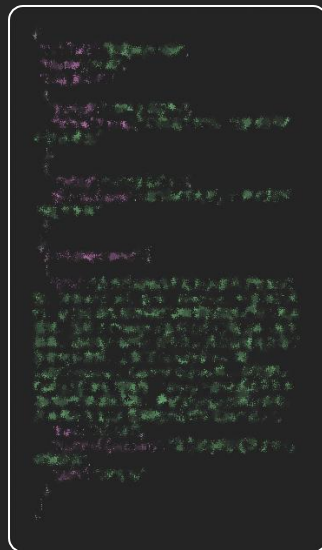


简历的解析



帮我总结一下这个人的信息,
用下面的JSON格式描述:

```
{  
  "name": string,  
  "gender": string,  
  "age": string,  
  "title": string,  
  ...  
}
```



接口的定义

generateContent (

```
/**
 * 生成内容
 * @param {string} topic 主题
 * @param {string} style 风格
 * @param {string} length 长度
 * @return {string} 生成的内容
 */
function generateContent(topic, style, length) {
  // 生成内容逻辑
  return '生成的内容';
}
```

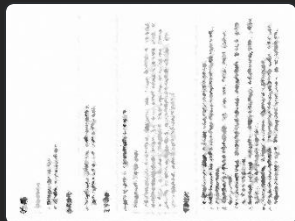
帮我总结一下这个人的信息，
用下面的JSON格式描述：

```
{
  "name": string,
  "gender": string,
  "age": string,
  "title": string,
  ...
}
```

```
{
  "name": "张三",
  "gender": "男",
  "age": "30",
  "title": "工程师"
}
```

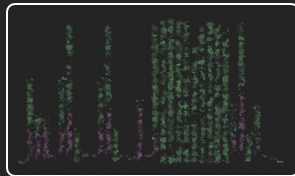

接口的定义

generateContent (String?, String?, prompt: String, string, "title": string, ...) : ?



帮我总结一下这个人的信息，用下面的JSON格式描述：

Prompt: String



接口的定义

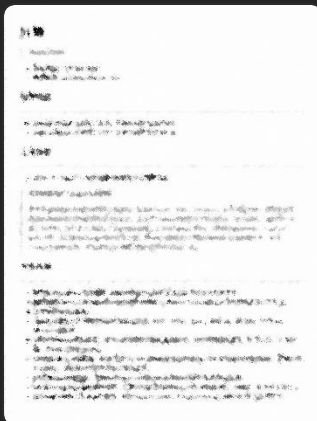
```
generateContent(file: String?,  
                mime: String?,  
                prompt: String  
                ): String?
```

```
generateContentStream( file: String?,  
                       mime: String?,  
                       prompt: String  
                       ): Flow<String?>
```

BaseAIRepository

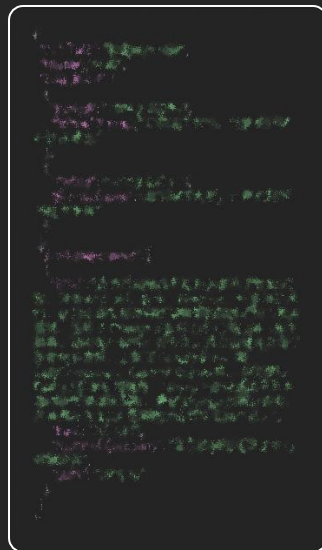
[illegible]

简历的解析



帮我总结一下这个人的信息,
用下面的JSON格式描述:

```
{  
  "name": string,  
  "gender": string,  
  "age": string,  
  "title": string,  
  ...  
}
```



简历的解析

```
{  
  "name": string,  
  "gender": string,  
  "age": string,  
  "title": string,  
  ...  
}
```

JSON结构化输出

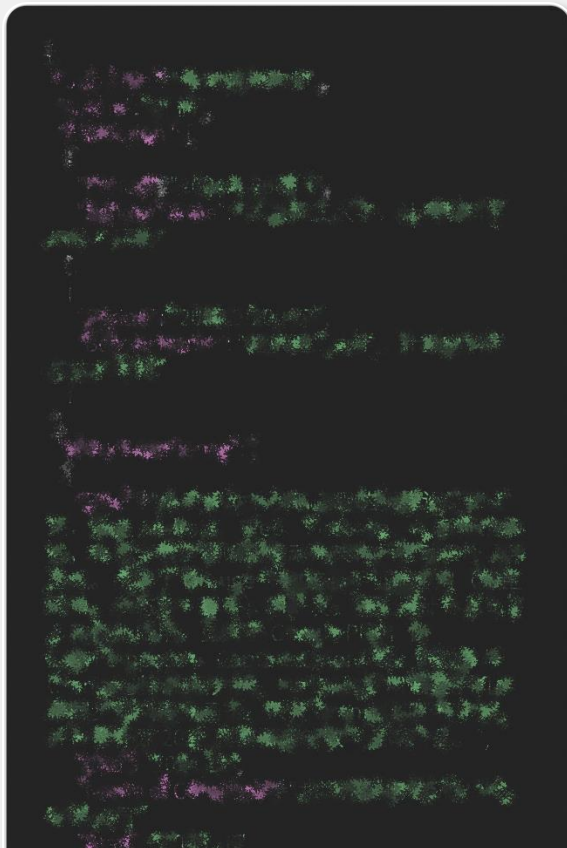
```
generationConfig = generationConfig {  
    responseMimeType = "application/json"  
    responseSchema = Schema(  
        name = "resume",  
        type = FunctionType.OBJECT,  
        description = "personal information",  
        properties = mapOf(  
            "name" to Schema(  
                name = "name",  
                type = FunctionType.STRING,  
                description = "Name of the person",  
                type = FunctionType.STRING),  
            "title" to Schema(  
                name = "title",  
                description = "Title of the person",  
                type = FunctionType.STRING)  
        )  
    )  
}
```

同步生成

Vertex

Gemini

Schema类配置输出



JSON结构化输出

```
model.generateContent(  
  content {  
    text(  
      """  
      帮我总结一下这个人的信息，用下面的JSON格式描述：  
      {  
        "name": string,  
        "name": string,  
        "gender": string,  
        "gender": string,  
        "age": string,  
        "age": string,  
        "title": string,  
        ...  
      }  
      """  
    ).trimIndent()  
  }  
)
```

推荐使用

流式生成

同步生成

MediaPipe

Gemini Nano

Vertex

Gemini

Schema提示词限制输出

4种方式的使用限制

	Gemini	Vertex	Gemini Nano	MediaPipe
API Level	21	21	31	24
图像输入	✓	✓	✗	✗
文件输入	✓	✓	✗	✗
JSON结构化输出	✓	✓	✓	✓
最大输出Token	<u>8192</u>	<u>8192</u>	~250	<u>不大于8192</u>
库尺寸	143KB	476KB	316KB	3.5MB / arm64-v8a

4种方式的性能比较

生成一篇1000字的童话故事



4种方式的Gemini性能比较

Vertex

Nano

Gemma2





谢谢！

Gemini API

<https://ai.google.dev/gemini-api/docs?hl=zh-cn>

Firebase Vertex AI

<https://firebase.google.com/docs/vertex-ai?hl=zh-cn>

Gemini Nano

<https://developer.android.com/ai/gemini-nano/experimental>

MediaPipe

LLM

<https://developers.googleblog.com/en/large-language-models-on-device-with-mediapipe-and-tensorflow-lite/>



Google
Developer
Groups