Incorporating LLMs into Your Flutter Application

LLMs and Flutter

- Currently Flutter actively supports most major LLMs
  - OpenAI ChatGPT & DALL-E, Google Gemini, Meta LLama, Mistral AI, …
- Since most LLMs are very large models, they are accessed from your app via API calls
- All LLM vendors require that you sign up for their service in order to obtain an API key to use the respective service
- Most API keys come with limited or no free-tier usage and most importantly, they require credit-card signup
- A modest exception to the above rule is Google :)

Getting a Gemini API Key

- Follow this link to obtain a Gemini API Key: https://aistudio.google.com/app/apikey
  - Select the *Develop in your own environment* option and accept license terms if asked
- This key is not directly linked to a Google Billing Account and can be used up until it exhausts free tier resources
  - After which you get an 429: RESOURCE EXHAUSTED API Response
- See https://ai.google.dev/pricing for rate limit details

Incorporate the API Key into Your Flutter Project

- Create a new flutter project via Visual Studio Code as usual
- Create file **api-keys.json** with the following content

```
{
    "GOOGLE_AI_KEY": "_your_api_key_here"
}
```

Tell VS Code to LAUNCH Debugger Reading Your Keys

- Create **.vscode/launch.json** and put the following content in it

```
{
  "version": "0.2.0",
  "configurations": [
      {
      "name": "Launch",
      "request": "launch",
      "type": "dart",
      "program": "lib/main.dart",
      "args": [
      "--dart-define-from-file",
      "api-keys.json"
      ]
      }
  ]
}
```

Set GRADLE Version

- Find your Java version by typing **flutter doctor -v**. Then consult the Gradle Compatibility Matrix to identify the corresponding gradle version (3rd column). Finally set the *distributionUrl* parameter of **android/gradle/wrapper/gradle-wrapper.properties** accordingly

```
distributionUrl=https\://services.gradle.org/distributions/gradle-8.5-all.zip
```

Add Google Generative AI dependency

- We will be using the google generative ai plugin
  - We will go through its flutter example, available here
- Add the following in **pubspec.yaml** under flutter dependencies

```
google_generative_ai: ^0.4.7

flutter_markdown: ^0.7.7+1
```

- Set the version of plugin id "*com.android.application*" to "*8.3.2*" in **android/settings.gradle**
- See the example to be presented on screen

Second Example: Use LLMs To Enhance UX

- We will extend the 2D2D functionality, using an LLM for creating tasks
- We will mostly follow the example code available here
- We will create **api-keys.json** and **.vscode/launch.json** as before
- We will using the same dependencies in **pubspec.yaml** as before
  - Adding url_launcher: ^6.3.2

Second Example: Other Tweeks

- Upgrade gradle to 8.5 in **android/gradle/wrapper/gradle-wrapper.properties**
- Set *plugin id "com.android.application"* to version *8.3.2* in **android/settings.gradle**