Android

# Develop faster with Gemini Agents in Android Studio


**Adarsh Fernando**

Group Product Manager
Android Studio

@adarshfernando

**Manda Edling**

Sr. UX Designer
Android Studio

@manda_edling

# Our guiding principles

### Transparency and Control

Be transparent in what data we use and how we use it. Make sure the users have full control over the data they want to share.

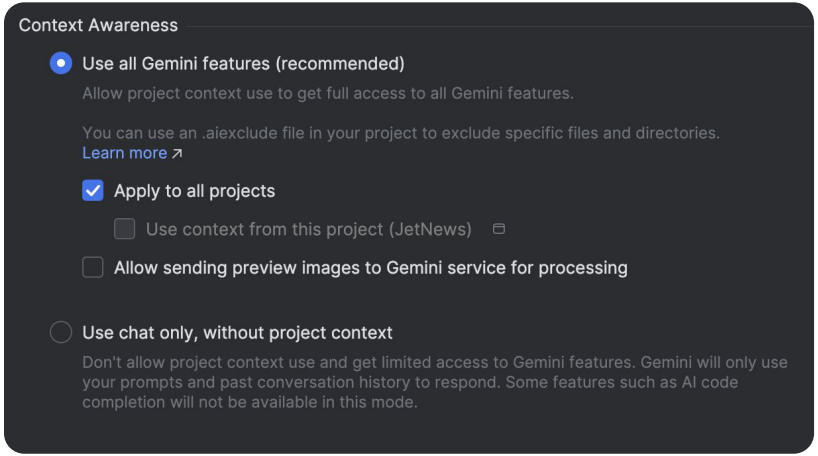
### Accelerate productivity

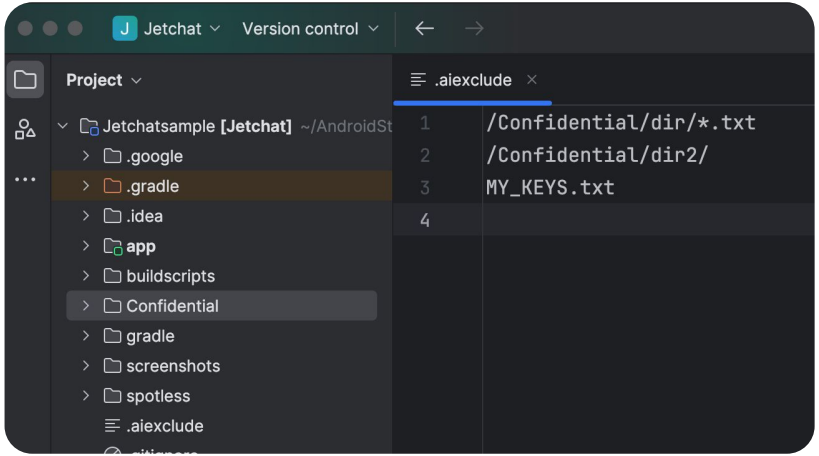
Provide assistive tools that meet developers where they are, to accelerate their current workflow. These can be a number of smaller improvements, or large ones.

### Maximize creativity

Provide experiences that allow developers to explore new ideas, experiment, iterate, and make developing with Gemini in Android Studio rewarding, productive, and fun.

# Transparency and Control



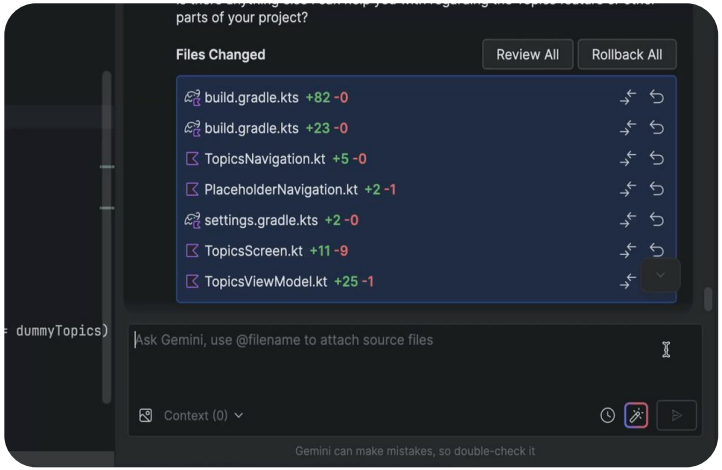

### Clear and transparent options

You control how much or how little access to your code to give Gemini. Providing Gemini with your code context provides the maximum capabilities and response quality.

### Access control with `*.aiexclude`

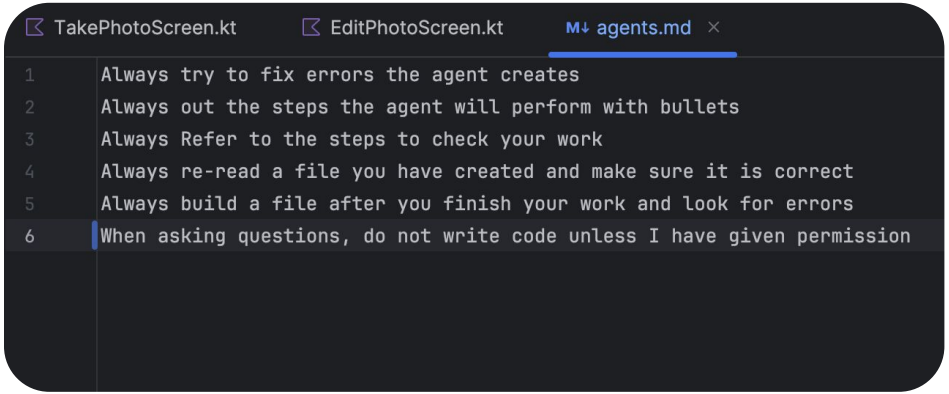
For more granular control, use aiexclude files (`.gitignore` syntax) to exclude specific files or subdirectories from sharing with Gemini.

# Accelerate productivity




### (new) Agent mode and MCP

By accessing tools, such as build/sync, code search/analysis, and refactoring, the agent reproduce common developer tasks to resolve much more complex tasks. With support to configure MCP servers, the possibilities are endless.




### (new) `AGENTS.md`

Define project-specific instructions, coding style rules, and other guidance as context when prompting Gemini. These files, like `.aiexclude`, can be checked in to VCS to easily share across your team.

# Maximize creativity

# Getting Started

01 Stitch to Figma

02 Configure `.aiexclude`

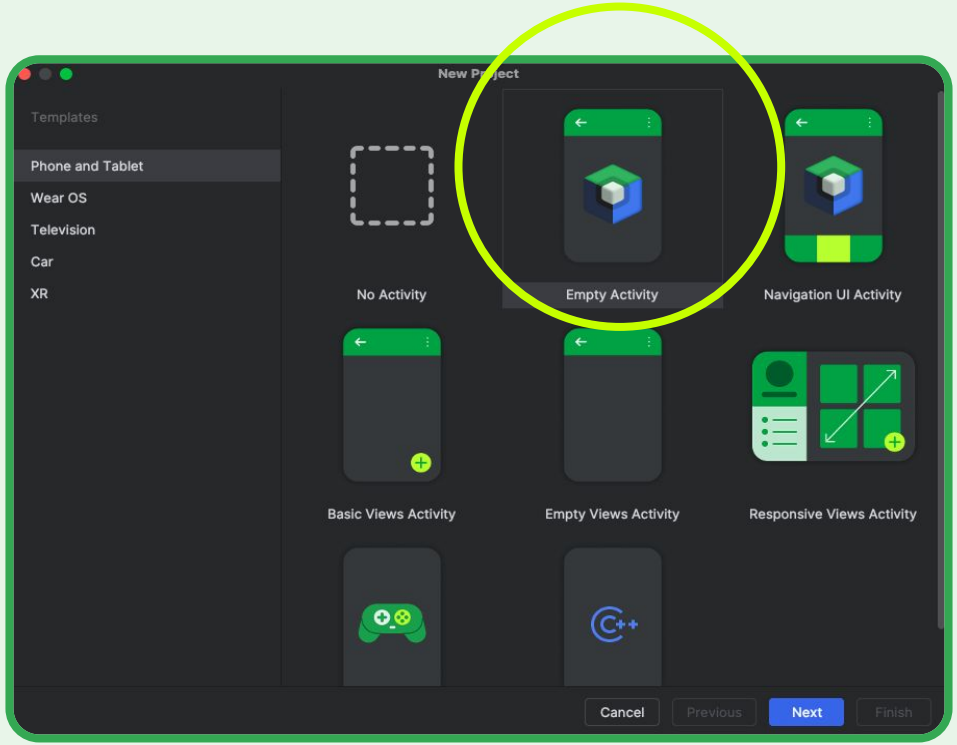
03 Configure `AGENTS.md`

04 Configure MCP

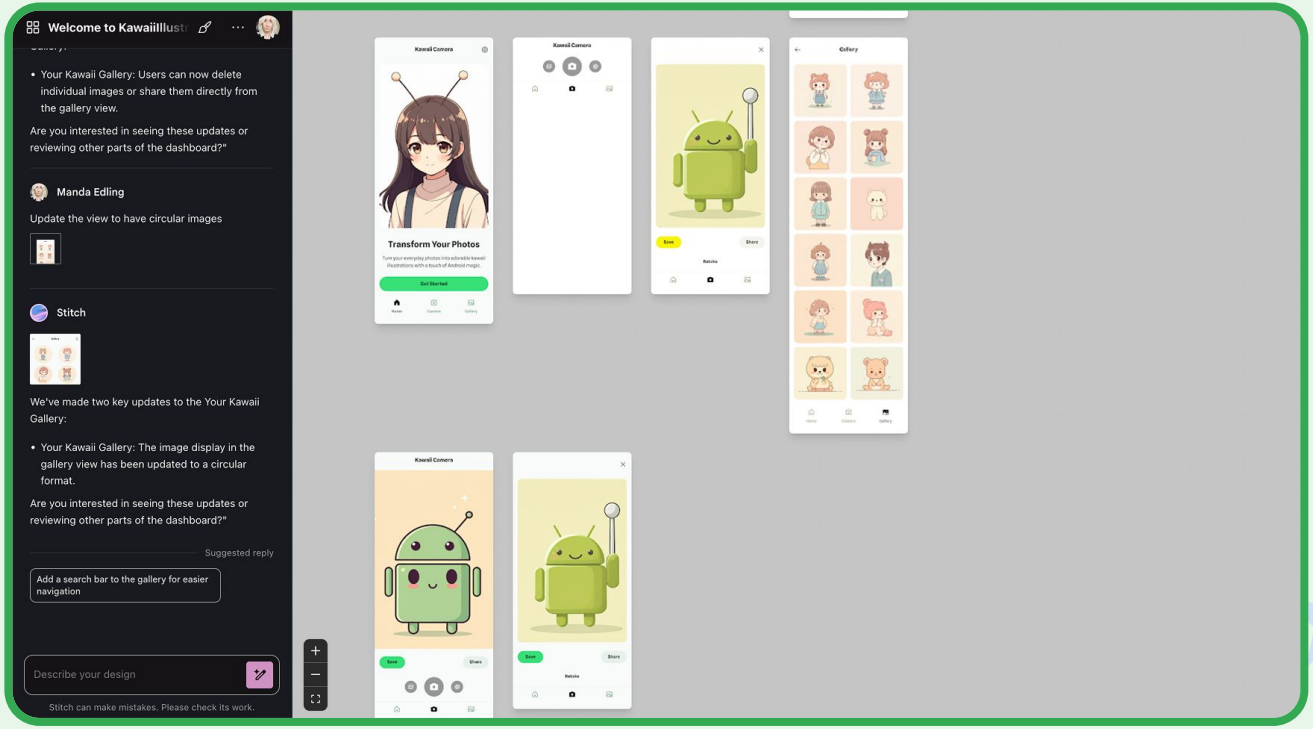
05 Figma to Code

Android Developers

# Getting Started

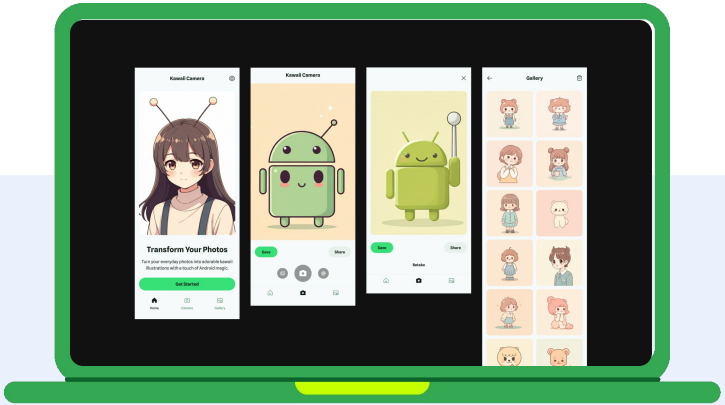


# Stitch


Welcome to KawaiiIllustr
• Your Kawaii Gallery: Users can now delete individual images or share them directly from the gallery view.
Are you interested in seeing these updates or reviewing other parts of the dashboard?"
Manda Edling
Update the view to have circular images
Stitch
We've made two key updates to the Your Kawaii Gallery:
• Your Kawaii Gallery: The image display in the gallery view has been updated to a circular format.
Are you interested in seeing these updates or reviewing other parts of the dashboard?"
Suggested reply
Add a search bar to the gallery for easier navigation
Describe your design
Stitch can make mistakes. Please check its work.
Kawaii Camera
Transform Your Photos
Get Started
Gallery

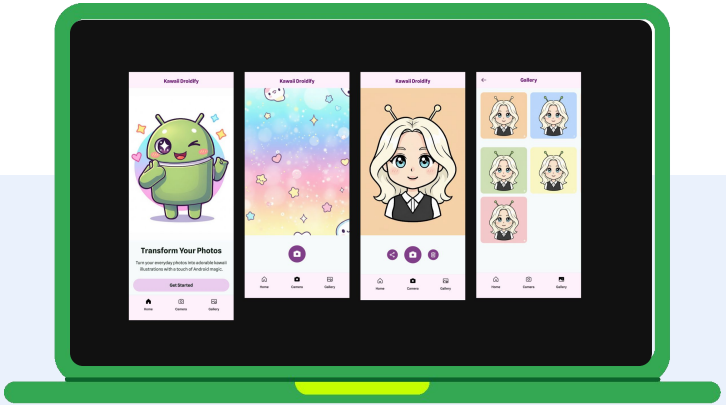
# Easily copy Stitch designs into Figma

## Stitch Designs

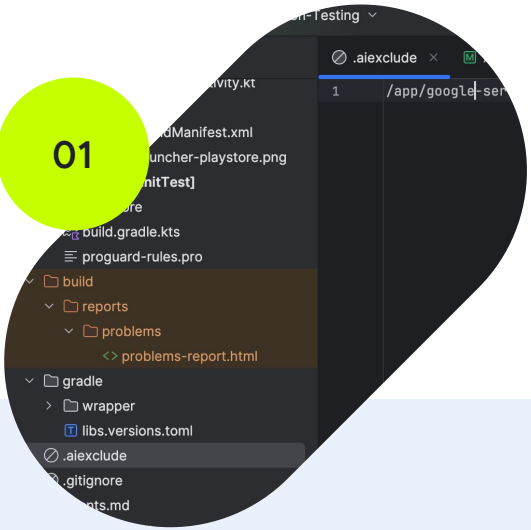



These images have layers that can be easily edited and manipulated in Figma

## Modified in Figma




I used Figma to tweak button designs, theme colors and add cute Kawaii images generated by Gemini




.aiexclude

## Add .aiexclude to a project


Agents.md

## Add AGENTS.md file to refine agentic responses


Model Context Protocol (MCP)

## Use Figma MCP to inform Agent on App design

Android Developers

# Demo: Getting Started

Getting Started Video

# Iterate & refine

01 Update dependencies

02 Bring Your Own Model (BYOM)

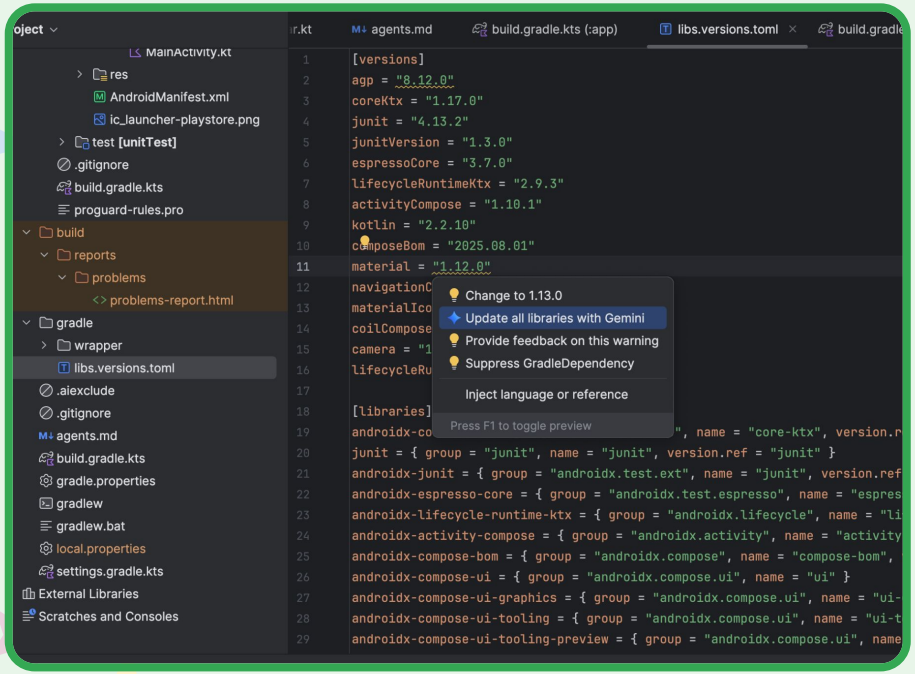
03 Configure local model

04 Multimodal: Attach image, @file

05 UI Transforms from Compose Preview

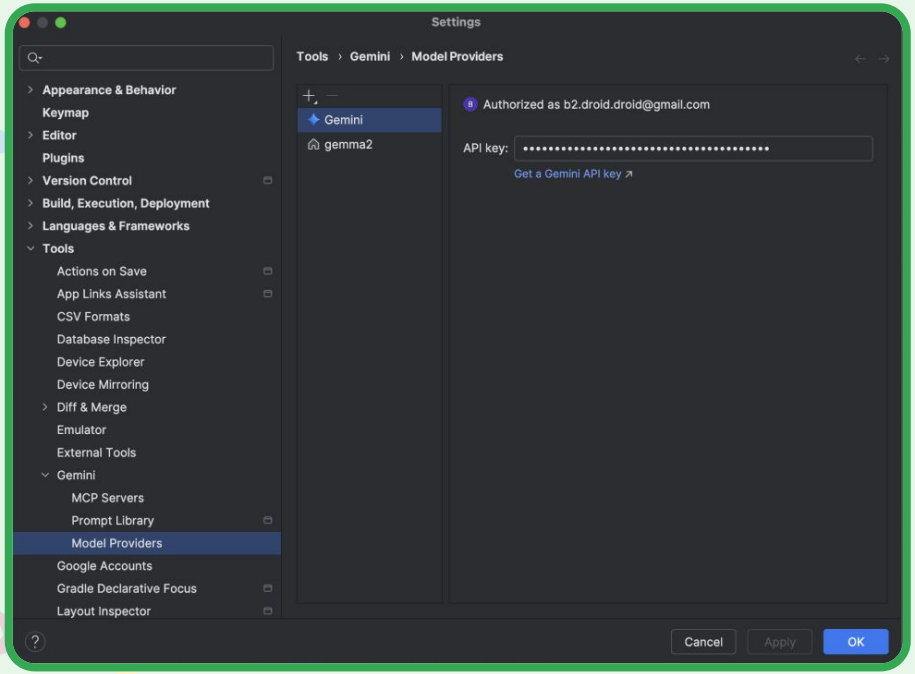
Android Developers

# Update Dependencies



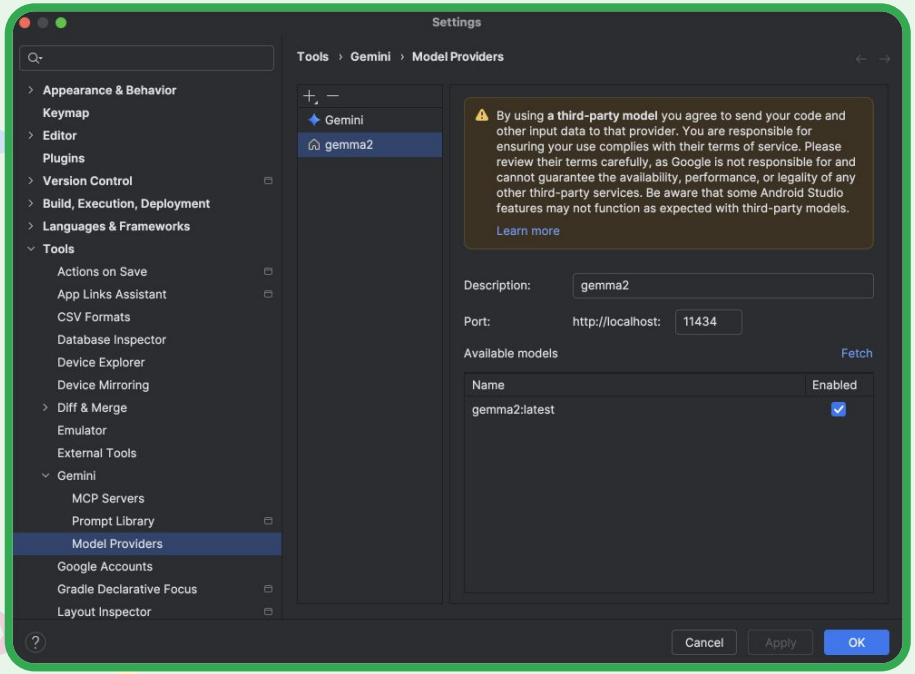

**Update you app dependencies in libs.versions.toml**

# Bring your own Model




You can add your own Gemini model to your work flow and switch between your models and Gemini's default model.

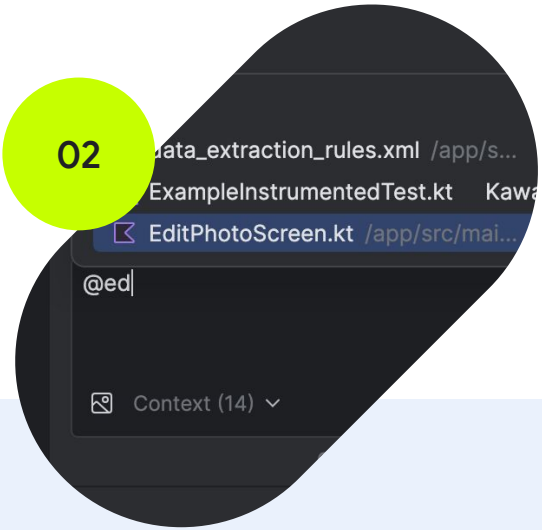
# Configure a Local Model




**Add and run your favorite local model**

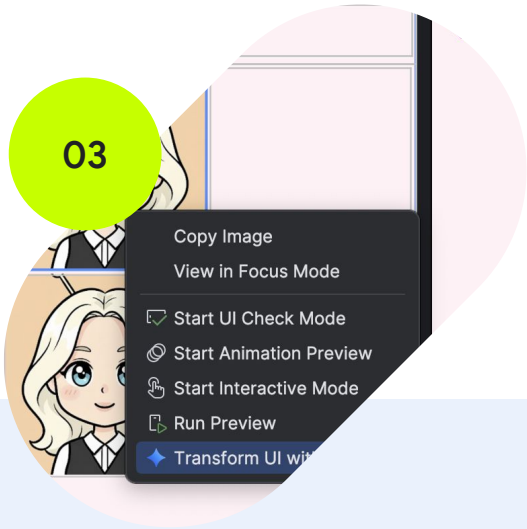

Attach Image

## Attach an image and use it to developer your UI




@file

## Attach specific project file context to a prompt




UI Transform in Preview

## Use Gemini to transform UI elements from preview

Android Developers

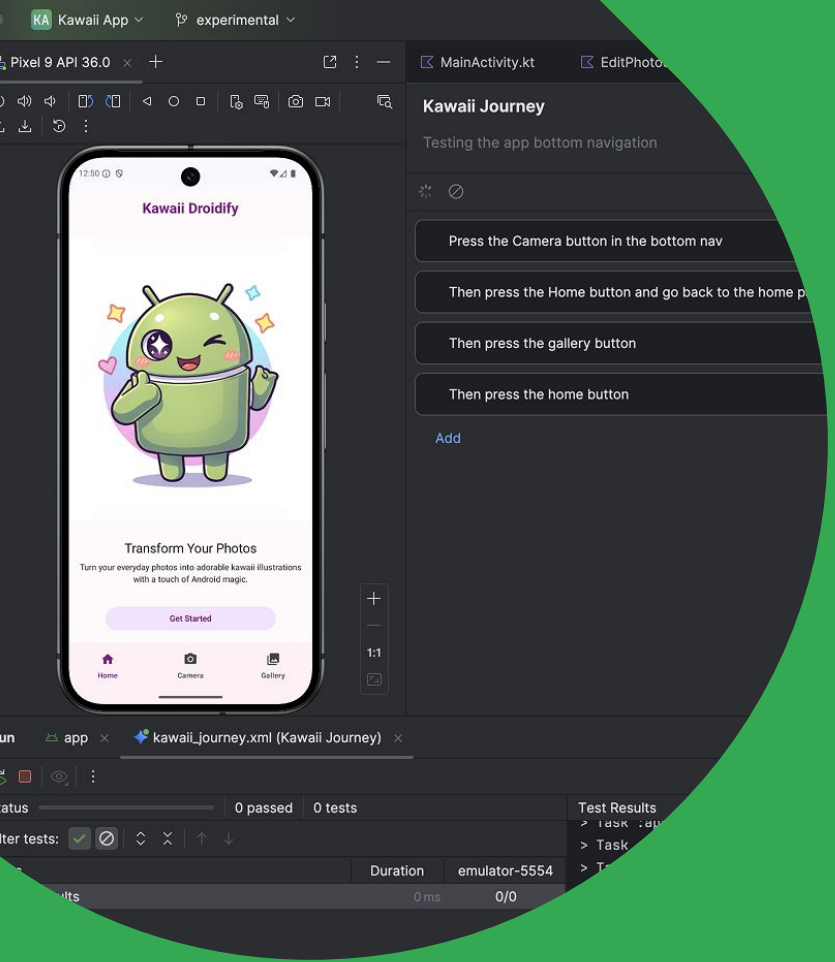
# Demo: Iterate & Refine

Iterate & Refine Video

# Testing

01 Generating Unit tests

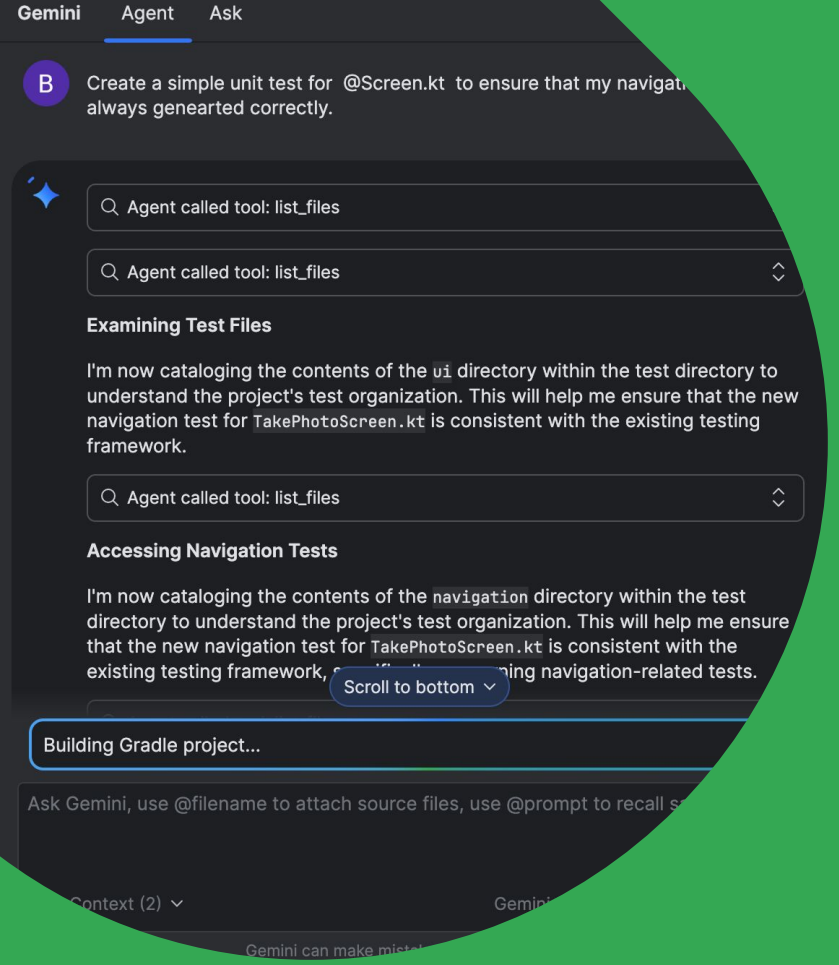
02 Testing with Journeys




# Journeys

Creating journeys for Android Studio helps make end-to-end tests easy to write and maintain by letting you use natural language to describe the steps and assertions for each test

Android Developers




# Generate Unit tests

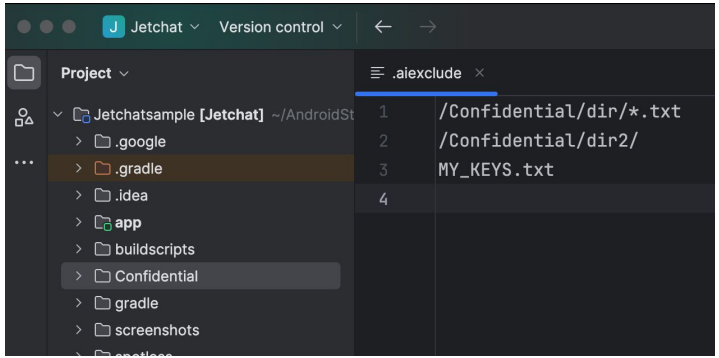
Agent mode can create unit tests using the context of the code you want to test. When generating unit test scenarios, Gemini includes detailed names and descriptions for your tests, so that you better understand the intention for each test.

Android Developers

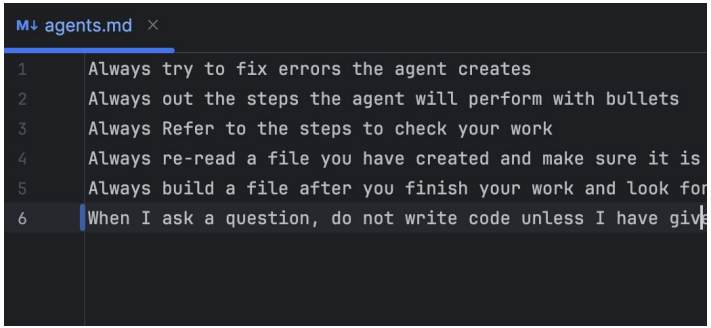
# Demo: Testing

Testing Video

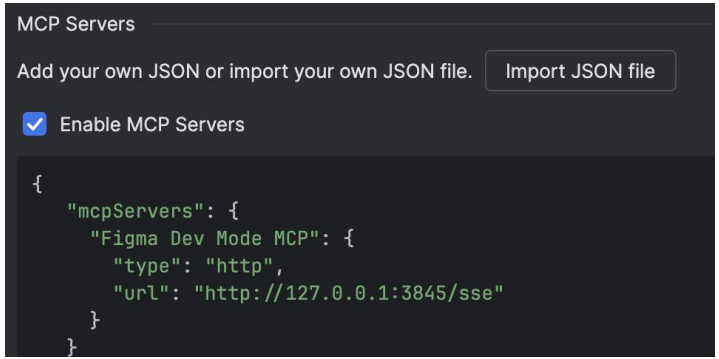
# Recap: Getting started with Agents



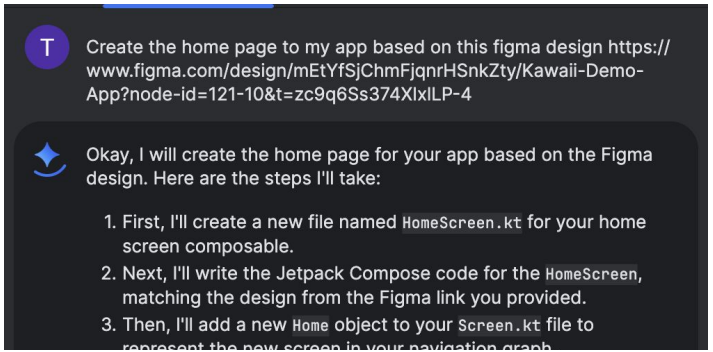

[Control code access with `*.aiexclude`]




[Customize model behavior with `AGENTS.md`]

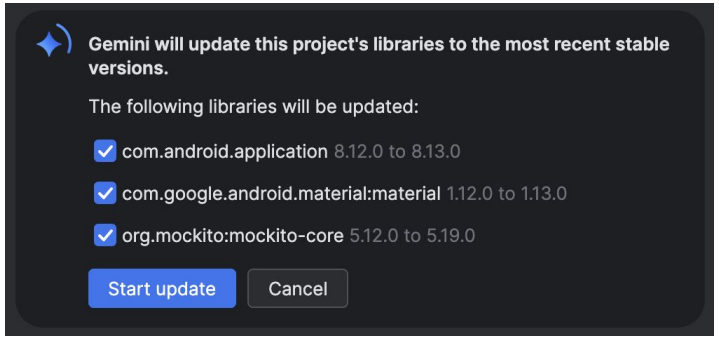



[Configure MCP servers]

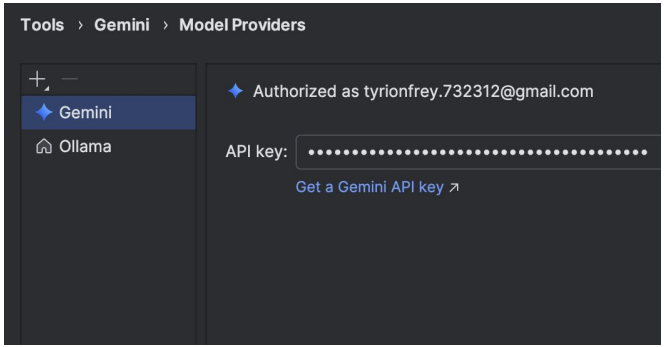



[Figma to Code]

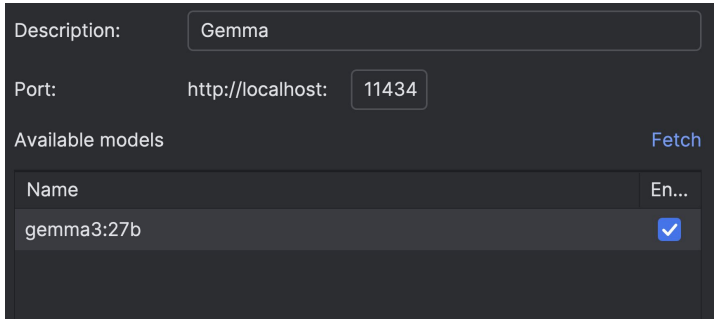
# Recap: Iterate and refine with Agents



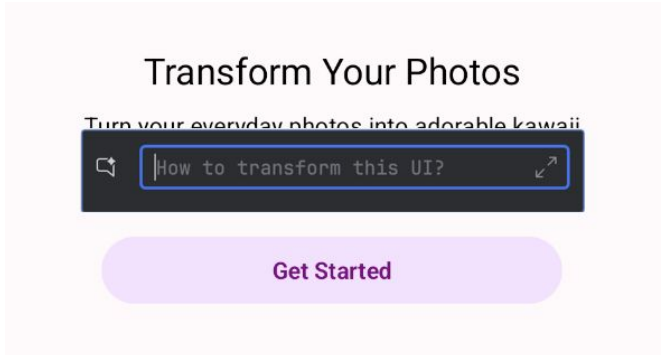

Update dependencies (coming soon!)




Add a Gemini API key



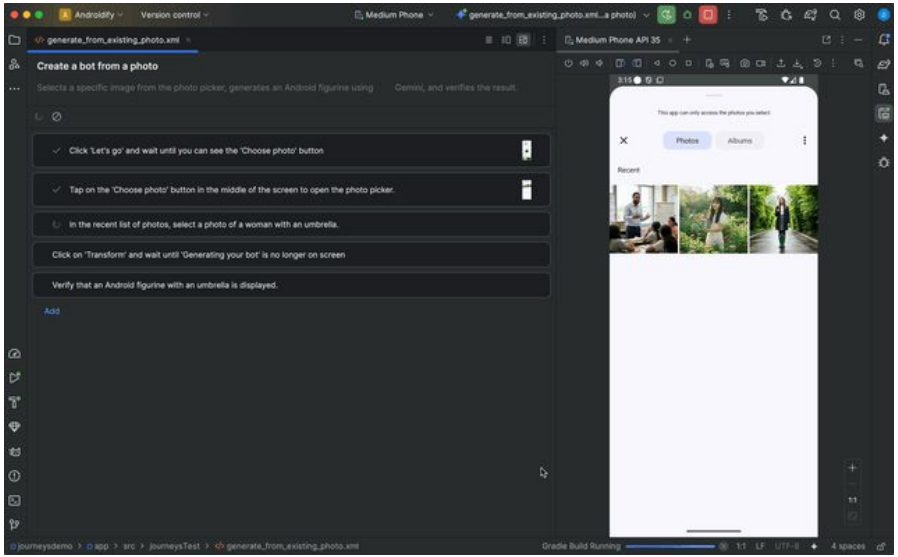

Use a local model



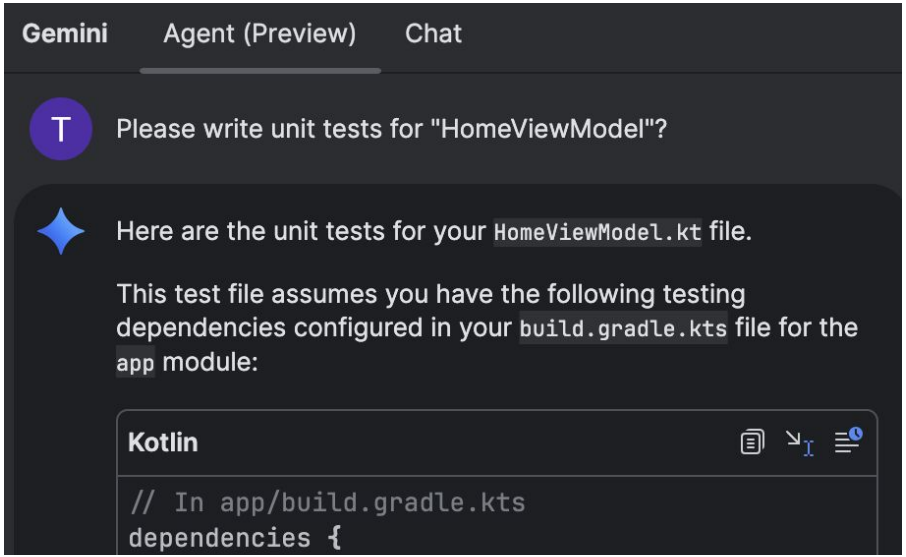

UI Transforms from Compose Previews

# Recap: Test your app with Agents






[Journeys]




[Generate unit and Instrumented tests with the Agent]

# The End