Представляем Android Sunflower

Садовое приложение, иллюстрирующее лучшие практики разработки Android Jetpack

[](https://medium.com/@tiembo?source=post_page-----e421b43fe0c2----------------------)

[Tiem Song](https://medium.com/@tiembo?source=post_page-----e421b43fe0c2----------------------)

следить

[27 августа 2018 г.](https://medium.com/androiddevelopers/introducing-android-sunflower-e421b43fe0c2?source=post_page-----e421b43fe0c2----------------------) · 4 минуты чтения

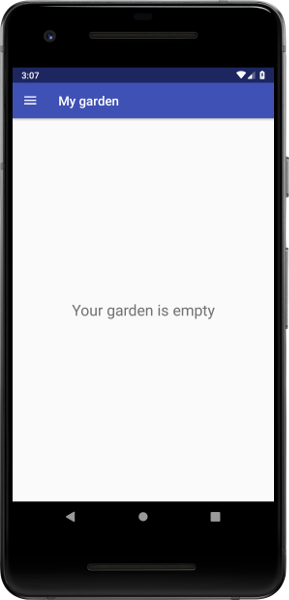
Google анонсировал [Android Jetpack](https://developer.android.com/jetpack/) на Google I / O 2018. Jetpack - это набор программных компонентов Android, облегчающих разработку великолепных приложений для Android.  
Эти компоненты помогают вам следовать рекомендациям, освобождают вас от написания стандартного кода и упрощают сложные задачи, поэтому вы можете сосредоточиться на коде, который вам небезразличен.

Чтобы помочь разработчикам научиться интегрировать компоненты Jetpack в свои приложения, мы создали [Sunflower](https://github.com/googlesamples/android-sunflower) , приложение для садоводства, иллюстрирующее лучшие практики разработки Android с помощью Android Jetpack.  
Если вы новичок в Jetpack, неплохо было бы проверить [содержимое Jetpack на developer.android.com](https://developer.android.com/jetpack/) и [создать свое первое простое приложение на основе Jetpack,](https://developer.android.com/jetpack/docs/getting-started) прежде чем продолжить.

Эта статья о Подсолнухе является первой в серии. Он проведет вас по Sunflower и кратко объяснит, какие компоненты Jetpack включены в приложение.  
В следующих статьях мы расскажем о разработке Jetpack и Android и углубимся в конкретные темы, такие как навигация и планирование повторяющихся задач с помощью WorkManager.

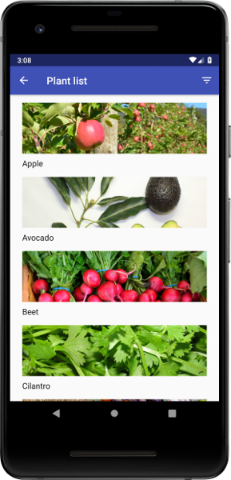
**Быстрый тур по саду**

Когда приложение запускается впервые, пользователь видит экран «Мой сад», который изначально пуст.



Экран My Garden с пустым садом

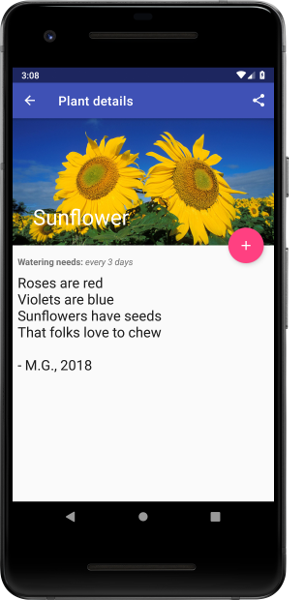
Давайте добавим растение! Перейдите к списку растений, нажав в меню гамбургера и открыв ящик для навигации.



Список растений

Каждое растение связано с Зоной выращивания, которая определяется широтой и указывает регион, наиболее подходящий для роста растения. Вы можете отфильтровать список растений, нажав на элемент меню фильтра, который выбирает Зону роста.

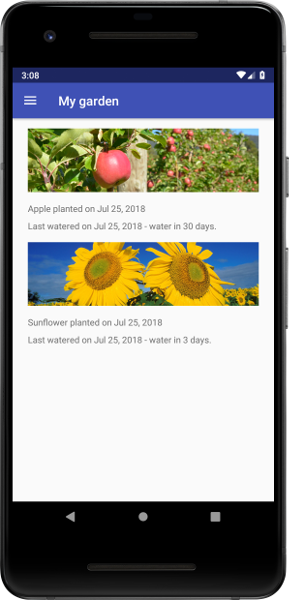
Нажав на конкретное растение, вы перейдете к экрану Plant Detail.



Детали завода

Экран сведений о растении показывает название растения, описание, потребности в поливе и изображение растения. Пользователь может добавить растение в свой сад с этого экрана, нажав [кнопку](https://developer.android.com/guide/topics/ui/floating-action-button) с [плавающим действием](https://developer.android.com/guide/topics/ui/floating-action-button) . Они также могут [поделиться](https://developer.android.com/training/sharing/) заводом через пункт меню.

После добавления одного или нескольких растений каждое новое растение появляется на экране «Мой сад» вместе с датой, когда оно было посажено, и когда его необходимо поливать. Давайте вернемся к «Моему саду», чтобы увидеть экран теперь, когда мы добавили растение или два.



Мой сад с добавленными растениями

Экран «Мой сад» теперь показывает добавленные растения, а также дату их посадки и последнего полива.

**Использование компонентов Jetpack**

Подсолнух использует много компонентов Jetpack. Вот краткий обзор того, как каждый компонент играет свою роль:

* Подсолнух написан полностью на [Kotlin](https://developer.android.com/kotlin) , используя [Android KTX](https://developer.android.com/kotlin/ktx) . Android KTX - это набор [расширений Kotlin,](https://kotlinlang.org/docs/reference/extensions.html) который оптимизирует API Jetpack и платформы Android для более краткого и идиоматического кода Kotlin.
* Приложение использует одну активность с несколькими [фрагментами](https://developer.android.com/guide/components/fragments) . Переходы между фрагментами используют [компонент Навигация](https://developer.android.com/topic/libraries/architecture/navigation/) и [действия анимации перехода](https://developer.android.com/topic/libraries/architecture/navigation/navigation-implementing#Create-transition) .
* На экранах используются макеты фрагментов, созданные с использованием [ConstraintLayout](https://developer.android.com/training/constraint-layout/" \t "_blank) и [Data Binding](https://developer.android.com/topic/libraries/data-binding/" \t "_blank)
* Хранение данных на устройстве списка растений и записей моего сада управляется с помощью [Room](https://developer.android.com/topic/libraries/architecture/room" \t "_blank) на уровне базы данных и [отображается](https://developer.android.com/topic/libraries/architecture/livedata) в пользовательском интерфейсе через [ViewModels](https://developer.android.com/topic/libraries/architecture/viewmodel" \t "_blank) через [LiveData](https://developer.android.com/topic/libraries/architecture/livedata" \t "_blank)
* [AppCompat](https://developer.android.com/topic/libraries/support-library/packages#v7-appcompat) используется для сохранения ключевой функциональности приложения на старых версиях Android
* Фоновые задачи обрабатываются [WorkManager](https://developer.android.com/topic/libraries/architecture/workmanager)
* Сведения о заводе могут быть [переданы](https://developer.android.com/training/sharing/shareaction) другим приложениям на устройстве или просто скопированы в буфер обмена
* [Тестирование](https://developer.android.com/training/testing/) осуществляется как [локальных тестов JUnit](https://developer.android.com/training/testing/unit-testing/) и [Espresso](https://developer.android.com/training/testing/espresso/" \t "_blank) тестов Android UI

**Будущие улучшения**

Sunflower в настоящее время выпускается в виде альфа-версии и находится в стадии интенсивного развития, с новыми функциями и более глубокими интеграциями компонентов. Некоторые из этих предстоящих улучшений включают фильтрацию списка растений в зависимости от местоположения пользователя, [переход](https://developer.android.com/topic/libraries/support-library/androidx-overview) на [AndroidX](https://developer.android.com/topic/libraries/support-library/androidx-overview" \t "_blank) и уведомление о необходимости полива растения.

Создайте и проголосуйте за [проблемы](https://github.com/googlesamples/android-sunflower/issues) на GitHub, которые вы хотели бы видеть реализованными или изученными.

В следующих статьях мы рассмотрим использование других компонентов Jetpack в Sunflower, таких как навигация и повторяющиеся задачи, так что следите за обновлениями. Спасибо за чтение!

Introducing Android Sunflower

A gardening app illustrating Android Jetpack development best practices

[](https://medium.com/@tiembo?source=post_page-----e421b43fe0c2----------------------)

[Tiem Song](https://medium.com/@tiembo?source=post_page-----e421b43fe0c2----------------------)

Follow

[Aug 27, 2018](https://medium.com/androiddevelopers/introducing-android-sunflower-e421b43fe0c2?source=post_page-----e421b43fe0c2----------------------) · 4 min read

(This article is also available in Chinese at [WeChat](https://mp.weixin.qq.com/s/k-zQVB5EhsKXIR1H3hbGYQ) / 中文版请参考 [WeChat](https://mp.weixin.qq.com/s/k-zQVB5EhsKXIR1H3hbGYQ))

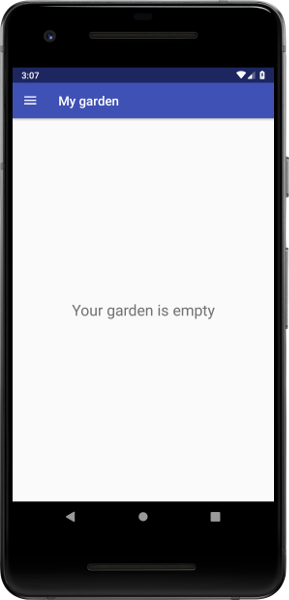
Google announced [Android Jetpack](https://developer.android.com/jetpack/) at Google I/O 2018. Jetpack is a collection of Android software components to make it easier for you to develop great Android apps. These components help you follow best practices, free you from writing boilerplate code, and simplify complex tasks, so you can focus on the code you care about.

To help developers learn how to integrate Jetpack components into their own apps, we created [Sunflower](https://github.com/googlesamples/android-sunflower), a gardening app illustrating Android development best practices with Android Jetpack. If you’re new to Jetpack, it’s a good idea to check out the [Jetpack content on developer.android.com](https://developer.android.com/jetpack/) and [build your first simple, Jetpack-based app](https://developer.android.com/jetpack/docs/getting-started) before continuing.

This article about Sunflower is the first in a series. It takes you on a tour of Sunflower and briefly explains which Jetpack components are included in the app. Future articles will discuss Jetpack and Android development and dive deeper into specific topics like Navigation and scheduling recurring tasks with WorkManager.

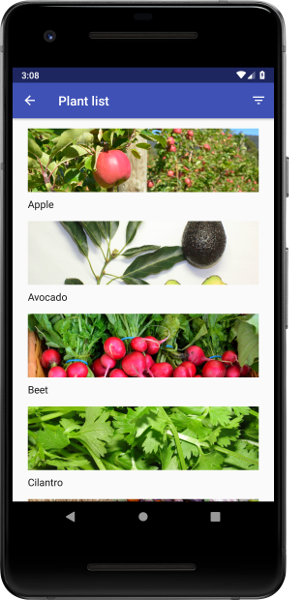
**A quick garden tour**

When the app is first launched, the user sees the “My Garden” screen, which is initially empty.



My Garden screen with an empty garden

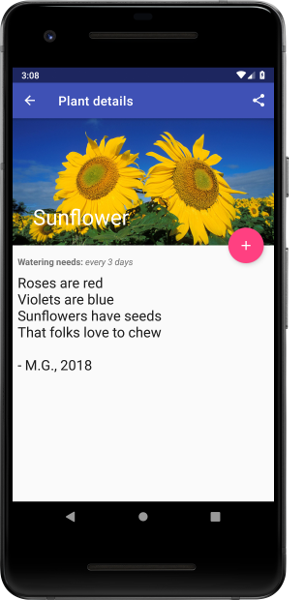
Let’s add a plant! Navigate to a list of plants by tapping on the hamburger menu and opening the navigation drawer.



Plant list

Each plant is associated with a Grow Zone, which is determined by latitude and indicates the region best suited for the plant to thrive. You can filter the list of plants by tapping the filter menu item which selects a Grow Zone.

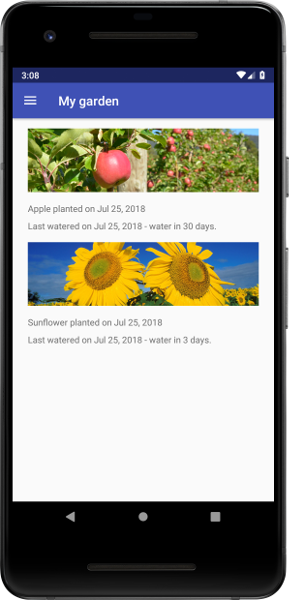
Tapping on a specific plant will navigate to the Plant Detail screen.



Plant details

The plant detail screen shows the plant’s name, description, watering needs, and a picture of the plant. The user can add the plant to their garden from this screen by tapping on the [Floating Action Button](https://developer.android.com/guide/topics/ui/floating-action-button). They can also [Share](https://developer.android.com/training/sharing/) the plant via the menu option.

After one or more plants are added, each new plant appears in the “My Garden” screen, along with the date it was planted and when it next needs to be watered. Let’s navigate back to “My Garden” to see the screen now that we’ve added a plant or two.



My garden with added plants

The “My Garden” screen now shows the added plants, along with the date they were planted and last watered.

**Using Jetpack components**

Sunflower uses many Jetpack components. Here’s a brief look at how each component plays its part:

* Sunflower is written entirely in [Kotlin](https://developer.android.com/kotlin), using [Android KTX](https://developer.android.com/kotlin/ktx). Android KTX is a set of [Kotlin extensions](https://kotlinlang.org/docs/reference/extensions.html) that optimizes Jetpack and Android platform APIs for more concise and idiomatic Kotlin code.
* The app uses a single Activity with multiple [Fragments](https://developer.android.com/guide/components/fragments). Transitions between fragments use the [Navigation component](https://developer.android.com/topic/libraries/architecture/navigation/) and [transition animation actions](https://developer.android.com/topic/libraries/architecture/navigation/navigation-implementing#Create-transition).
* The screens use fragment layouts, created using [ConstraintLayout](https://developer.android.com/training/constraint-layout/" \t "_blank) and [Data Binding](https://developer.android.com/topic/libraries/data-binding/)
* On-device data storage of the plant list and my garden entries are managed with [Room](https://developer.android.com/topic/libraries/architecture/room) at the database level, and surfaced to the UI through [ViewModels](https://developer.android.com/topic/libraries/architecture/viewmodel" \t "_blank) via [LiveData](https://developer.android.com/topic/libraries/architecture/livedata" \t "_blank)
* [AppCompat](https://developer.android.com/topic/libraries/support-library/packages#v7-appcompat) is used to preserve key app functionality on older versions of Android
* Background tasks are handled by [WorkManager](https://developer.android.com/topic/libraries/architecture/workmanager)
* Plant details can be [shared](https://developer.android.com/training/sharing/shareaction) with other applications on the device, or simply copied to the clipboard
* [Testing](https://developer.android.com/training/testing/) is performed by both [local JUnit tests](https://developer.android.com/training/testing/unit-testing/) and [Espresso](https://developer.android.com/training/testing/espresso/) Android UI tests

**Future improvements**

Sunflower is currently released as an alpha and is undergoing heavy development, with new features and deeper component integrations landing regularly. Some of these upcoming improvements include filtering the plant list based on the user’s location, migrating to [AndroidX](https://developer.android.com/topic/libraries/support-library/androidx-overview" \t "_blank), and notifying when a plant needs watering.

Create and vote for [issues](https://github.com/googlesamples/android-sunflower/issues) on GitHub that you’d like to see implemented or explored.

In future articles we’ll explore using other Jetpack components in Sunflower such as navigation and recurring tasks, so stay tuned. Thanks for reading!



A sunflower in a Google garden

**Continue Exploring**

* Check out the code for Sunflower at <https://github.com/googlesamples/android-sunflower>.
* Learn about [Jetpack](https://developer.android.com/jetpack/) and [build your first Jetpack app](https://developer.android.com/jetpack/docs/getting-started).

*Thanks to the wonderful individuals on the Android Developer Relations and Jetpack teams for their contributions to Sunflower and this article.*

**[Android Developers](https://medium.com/androiddevelopers?source=post_sidebar--------------------------post_sidebar-)**

The official Android Developers publication on Medium

–[](https://medium.com/@tiembo?source=follow_footer--------------------------follow_footer-)

WRITTEN BY

[**Tiem Song**](https://medium.com/@tiembo?source=follow_footer--------------------------follow_footer-)

Follow

Dad, husband, Longhorn, cyclist. Android DevRel @ Google. Gardening is joyfully fighting a losing battle with squirrels.

[](https://medium.com/androiddevelopers?source=follow_footer--------------------------follow_footer-)

[**Android Developers**](https://medium.com/androiddevelopers?source=follow_footer--------------------------follow_footer-)

Follow

The official Android Developers publication on Medium

**Showing responses for:**

[Introducing Android Sunflower](https://medium.com/androiddevelopers/introducing-android-sunflower-e421b43fe0c2)

[Tiem Song](https://medium.com/androiddevelopers/introducing-android-sunflower-e421b43fe0c2)

[2.5K5](https://medium.com/androiddevelopers/introducing-android-sunflower-e421b43fe0c2)

**Responses**

Write a response…

[Go to the profile of Sujin Shrestha](https://medium.com/@sujinshrestha)

[Sujin Shrestha](https://medium.com/@sujinshrestha?source=responses---------0-----------------------)

[Nov 26, 2018](https://medium.com/@sujinshrestha/would-be-great-experiment-if-the-application-is-built-on-feature-wise-modular-approach-cef4c9a4a54?source=responses---------0-----------------------)

[Would be great experiment if the application is built on feature-wise modular approach. With including the Dagger and other stuffs too.](https://medium.com/@sujinshrestha/would-be-great-experiment-if-the-application-is-built-on-feature-wise-modular-approach-cef4c9a4a54?source=responses---------0-----------------------)

21

[Go to the profile of Jorge Ayala](https://medium.com/@ayalasaenzjorge)

[Jorge Ayala](https://medium.com/@ayalasaenzjorge?source=responses---------1-----------------------)

[Oct 26, 2018](https://medium.com/@ayalasaenzjorge/hello-im-trying-to-compile-this-project-in-android-studio-3-2-and-3-3-but-i-get-this-error-9873b7790e90?source=responses---------1-----------------------)

[Hello, I’m trying to compile this project in Android Studio (3.2 and 3.3) but I get this error](https://medium.com/@ayalasaenzjorge/hello-im-trying-to-compile-this-project-in-android-studio-3-2-and-3-3-but-i-get-this-error-9873b7790e90?source=responses---------1-----------------------)

[Thanks in advance.](https://medium.com/@ayalasaenzjorge/hello-im-trying-to-compile-this-project-in-android-studio-3-2-and-3-3-but-i-get-this-error-9873b7790e90?source=responses---------1-----------------------)

Read more…

5

[Go to the profile of prathamesh kesarkar](https://medium.com/@pratham.kesarkar)

[prathamesh kesarkar](https://medium.com/@pratham.kesarkar?source=responses---------2-----------------------)

[Oct 28, 2018](https://medium.com/@pratham.kesarkar/it-would-be-great-if-we-can-have-same-sample-written-in-mvi-or-redux-android-1e7e59c3af00?source=responses---------2-----------------------)

[It would be great if we can have Same sample written in MVI or Redux Android](https://medium.com/@pratham.kesarkar/it-would-be-great-if-we-can-have-same-sample-written-in-mvi-or-redux-android-1e7e59c3af00?source=responses---------2-----------------------)

1

[Go to the profile of Mark Chiu](https://medium.com/@markchiu)

[Mark Chiu](https://medium.com/@markchiu?source=responses---------3-----------------------)

[Mar 26](https://medium.com/@markchiu/i-am-very-angry-why-are-these-examples-written-in-kotlin-instead-its-very-unfriendly-to-java-b8aba4569105?source=responses---------3-----------------------)

[I am very angry, why are these examples written in Kotlin instead,  
it’s very unfriendly to java users,  
Obviously Android app developers are approaching 90% using Java.  
Now suddenly changed to Kotlin, so that make java users can not understand it.](https://medium.com/@markchiu/i-am-very-angry-why-are-these-examples-written-in-kotlin-instead-its-very-unfriendly-to-java-b8aba4569105?source=responses---------3-----------------------)