**Introducing Appwrite for Android: Best and Easy way to Self-Host**

Hello to all the readers, I’m glad to see you here for reading about my Blog post on [#appwrite](https://appwrite.io/) for Android. Today I am thrilled to introduce Appwrite to the Android Developer Community. Appwrite is basically open-source and self-hosted Firebase alternative that is extremely easy to deploy. Developers use the Android SDK to integrate the app with the Appwrite server to easily start interacting with all of Appwrite backend APIs and tools.



# ****What is Appwrite?****

Appwrite is a self-hosted open source backend as a service server that abstract and simplify complex and repetitive development tasks behind a very simple to use REST API. Appwrite aims to provide you with all the core APIs needed to build a modern application and helps you develop your apps faster and in a more secure way. It also helps to abstract all the boiler plate code to allow you to focus on creating awesome and innovative solutions.

The best part about Appwrite is we can build applications using Appwrite without even writing any backend code at all! But ya this is not the only way to use it. It’s just that the Appwrite provides you a better starting point for your project and a great developer experience. You can directly integrate Appwrite with your client app, use it behind your custom backend or alongside your custom backend server.

Although as of now, Appwrite is still in beta but the majority is of BaaS features, and it just getting better all the time. Backend by an amazing community of thousands of developers, over 250+ contributors and even now just recently over 10,000 GitHub stars⭐

# ****Getting Started for Android****

Appwrite is a development platform providing you an easy yet powerful API and management console to get your next project up and running quickly. In order to get started, first you need to have an Appwrite server instance up and running on your host machine or server.

# ****Create your First Appwrite Project****

Firstly Create your first Appwrite project and then Go to your Appwrite console. Once inside, click the **+** icon in the top navigation header or on the “Create Project” button on your console homepage. Then choose a name for your project and click create to get started.

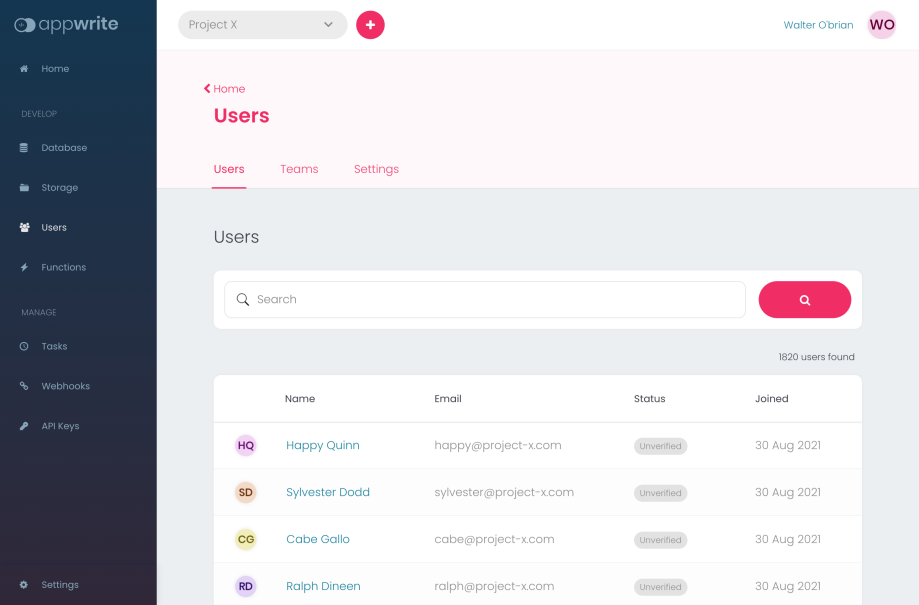
# ****Add your Android Platform****

In order to init your **SDK** and start interacting with Appwrite services, All you need is to add a new Android platform to your project. For this, just go to your App write console, then choose the project you created and then click on **Add Platform**button.

Now Add a new **Android** platform and then add your app name & package name. you can find your application id in your app-level **build.gradle** file. Now by registering your new app platform, you are allowing your app to communicate with the Appwrite API.

# ****Authentication and Users****

The Appwrite **Accounts** and **Users** services helps to expose your client to an API that allow you to interact with you app’s logged-in accounts and even easily manage your app users from the **Appwrite Android SDK** or the **Appwrite Console.**



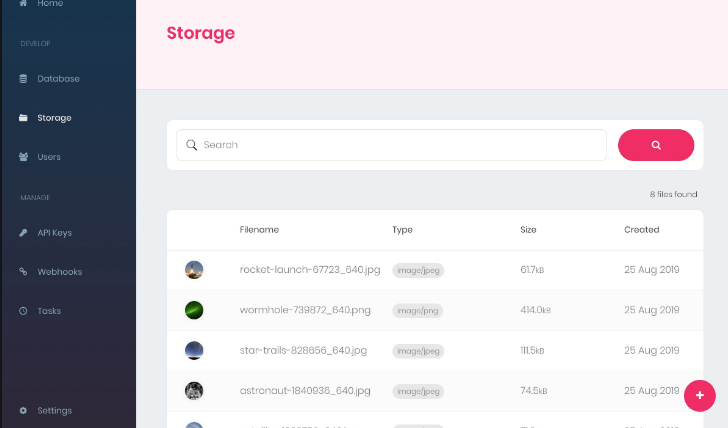
Updation of the Account related information are allowed for users by the **Account Service** and even save their private preferences like their **UI Language, Timezone** or **Favourite theme**. In addition, you can allow your users to securely store all their preferences.

You can also use the API to get a list of all active user sessions which include the information about the **session location, device, operating system and user-agent.**Here you can even let your users review their latest logins, password recoveries, and other security-sensitive events.

# ****Storage****

The Appwrite storage service is the easiest way to let your app users upload and manage their files securely. The API takes advantage of the same simple **read and write permissions** mechanism that the database uses. This allows you to easily grant read and write access to all users.

One advantage is also there ie. Appwrite also takes care of all your file compression, encryption and even provides you with built-in anti-virus scanning. **Security and Privacy** are the key points for which you would love to work with Appwrite.

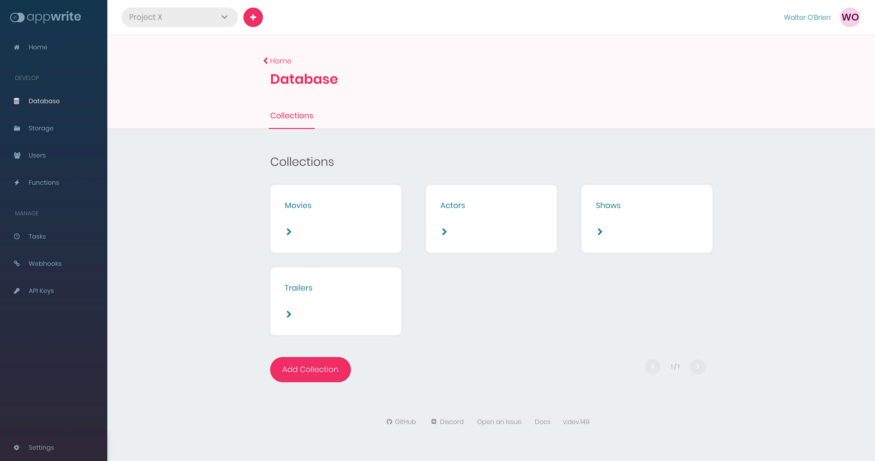


# ****Database****

The Database of the Appwrite API is a simple **REST API** that allows you to add persistency to your application. Lets take an example- you can create collections of documents and handle for which user has both read and write access. Currently, Appwrite uses MariaDB as the database. Appwrite offers an easy-to-use , document-based database API for storing your app’s data.

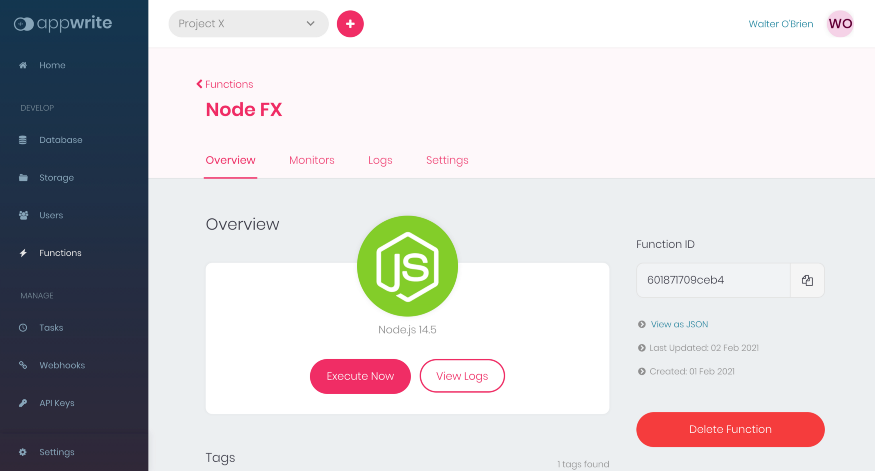
The Database service allows you to create structured collections of documents, query and filter lists of documents, and manage an advanced set of **read and write access permissions**.

All the data in the database service is stored in structured **JSON** documents. The Appwrite database service also allows you to nest child documents in parent documents and use deep filters to both search and query your data.



# ****Appwrite Functions****

The Appwrite functions API allows you to easily extend your Appwrite API behaviour and even capabilities in any programming language including Java and Kotlin. You can even trigger your custom functions using the Appwrite **API, Console or using a CRON Schedule**. You also get the facility to trigger your functions on any system event like user creation or when a new file is uploaded.



**Get Appwrite Android SDK**

1. Firstly, add this code to your root level **build.gradle** file:**-**

Repositories {

mavenCentral()

}

2. Now add this code to your project’s **build.gradle:-**

‘implementation(“io.appwrite:sdk-for-android:0.2.0”)’

3. Now Just find your **Appwrite PROJECT ID** in your **project Setting Screen of Appwrite Console:**-

<activity android:name="io.appwrite.views.CallbackActivity" >

<intent-filter android:label="android\_web\_auth">

<action android:name="android.intent.action.VIEW" />

<category android:name="android.intent.category.DEFAULT" />

<category android:name="android.intent.category.BROWSABLE" />

<data android:scheme="appwrite-callback-[PROJECT\_ID]" />

</intent-filter>

</activity>

4. Initialize your **SDK code with your Project ID:-**

import io.appwrite.Client

import io.appwrite.services.Account

val client = Client(context)

.setEndpoint("https://[HOSTNAME\_OR\_IP]/v1") // Your API Endpoint

.setProject("5df5acd0d48c2") // Your project ID

.setSelfSigned(true) // For self signed certificates, only use for development

# ****Get Started****

Appwrite is packaged as a set of Docker microservices that can run anywhere the Docker can run. This allows to deploy the Appwrite fairly easily using a single CLI Command :-

**CMD Command :-**

docker run -it — rm ^  
— volume //var/run/docker.sock:/var/run/docker.sock ^  
— volume “%cd%”/appwrite:/usr/src/code/appwrite:rw ^  
— entrypoint=”install” ^  
appwrite/appwrite:0.9.0

**PowerShell Command :-**

docker run -it — rm ,  
— volume /var/run/docker.sock:/var/run/docker.sock ,  
— volume ${pwd}/appwrite:/usr/src/code/appwrite:rw ,  
— entrypoint=”install” ,  
appwrite/appwrite:0.9.0

# ****What’s Next?****

So, Thank you for reading this blog post. If you wanted to know more about Appwrite and How does it works and what is the workflow of this. Just visit the [#appwrite](https://appwrite.io/) for the detailed information.