CatCol Setup Instructions (For Android Devices and Developed on MacOS Only)

# Create React Native Project

* Open the terminal inside of a directory that you’d like to create the project folder in. Also, install visual studio code from this website <https://code.visualstudio.com> as the code editor to make it easier to edit the code in the future.
* Check if you have node and npm installed using the ‘node -v’ and ‘npm -v’ commands, this should return a version number. If you don’t have them installed, follow this link <https://nodejs.org/en/download/package-manager> to install node.js and then use the ‘npm -g install npm’ command to install the corresponding npm package. (I worked with node version 18.18.0 and npm version 9.8.1)
* The ‘npx react-native init CatCol –-version 0.73.2’ command needs to be called to create a new project folder for the CatCol application containing all the required files for React Native applications. (possible to upgrade to newer version if so desired) <https://reactnative.dev/docs/0.70/environment-setup?guide=native>. (If you are missing any other tools, follow this tutorial to get all of them set up)
* If the ‘node\_modules’ folder already exist inside of the ‘CatCol’ folder, remove it. Then, copy the ‘package.json’ and ‘package-lock.json’ files from the ‘CatCol Source Code’ folder provided from the developer and copy them into the ‘CatCol‘ folder replacing the files with the same names.
* Then, navigate to the ‘CatCol’ folder and call the ‘npm install’ command inside of the terminal. This will create a new ‘node\_modules’ folder that includes all the third-party dependencies required for the application to run.

# Set up the backend on AWS Amplify

* Before the backend could be set up and connected to the frontend React Native application, the Amplify CLI needs to be install and set up on your development computer.
* To do that, follow this tutorial <https://docs.amplify.aws/gen1/react-native/start/getting-started/installation/> (Consult this documentation whenever you need help with interacting with AWS Amplify as well). Follow the ‘npm’ option for the ‘Install the Amplify CLI’ section, and the ‘Configure the Amplify CLI’ section. Skip the ‘Manually configure the Amplify CLI’ section.
* After the Amplify CLI is set up on your machine, log in to your AWS account and navigate to the AWS Amplify service using the search bar.

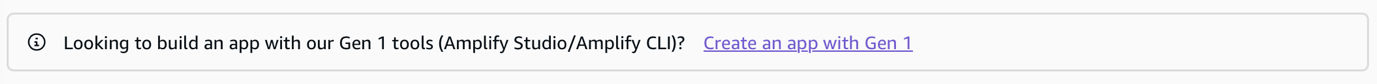
A screenshot of a computer

Description automatically generated

* Press the ‘Create new app’ button after navigating to the AWS Amplify service.



* Then, scroll down to the bottom of the ‘Choose source code provider’ page, and click on the ‘Create an app with Gen 1’ link.



* This should take you to the ‘Get started with Amplify Studio’ page, where you’ll be able to provide the name of the application, which should be ‘CatCol’.

A screenshot of a computer

Description automatically generated

* Then, press the ‘Confirm Deployment’ button and wait for AWS to deploy the application to their servers.
* After this is done, press the ‘Launch Studio’ button to be taken to the ‘Amplify Studio’ where all the services for the mobile application can be managed from.

A screenshot of a computer

Description automatically generated

* After the studio is launched, navigate to the menu bar on the left side of the screen and press the ‘Authentication’ button, and then select the ‘Start from scratch’ option.

A screenshot of a computer

Description automatically generated

* For the ‘Configure login’ setup, an ‘Email’ configuration should be chosen. And the ‘Multi-factor authentication’ should be turned off.

A screenshot of a computer

Description automatically generated

* Then, for the ‘Configure sign up’ setup, three attributes should be chosen, using the ‘Add attribute button’: ‘Email’, ‘Name’, and ‘Nickname’.

A blue lines on a white background

Description automatically generated

* Then, the password policy needs to be set up. All the tick boxes should be selected to maximise the security of the application.

A screenshot of a computer

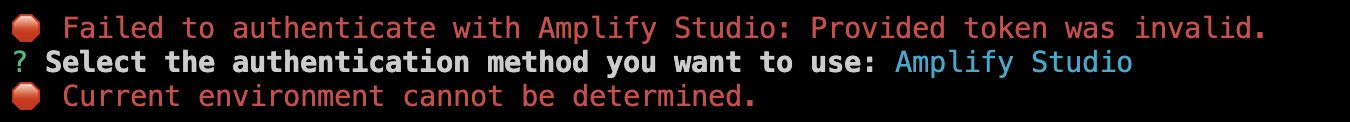
Description automatically generated

* Finally, for the verification mechanism, the Email option should be chosen, and then the email subject and email body can be left as default or can be customised as required.

A white square with black lines

Description automatically generated

* Then, press the ‘Deploy’ button and wait for the AWS servers to deploy the changes to the backend.
* After that is finished, follow the instructions provided by the ‘pull the latest client config’ section, by copying the piece of code that starts with the words ‘amplify pull’ followed by the unique details of your backend application. Paste this command into the terminal inside of the ‘CatCol’ folder of the React Native application.
* If you get the ‘failed to authenticate’ and ‘current environment cannot be determined’ errors and you press the ‘Amplify Studio option’, try and call the ‘amplify pull’ you copied earlier again.



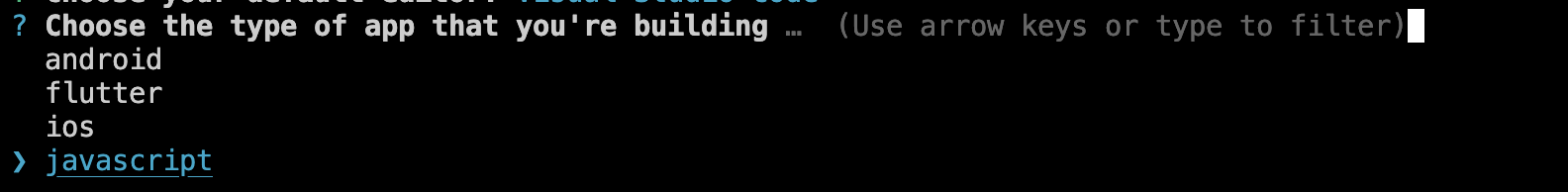
* This should open an AWS window in your default browser, where you’ll be able to copy a CLI key from.
* Paste this key into the terminal when prompted, and this should authenticate you and your computer to make changes to the Amplify backend through the terminal.



* You will then be able to continue with the rest of the set up for connecting the backend to the front end.
* As a continuation you will be prompted with a set of questions about the project setup, so answer them in the following way:

A black screen with white text

Description automatically generated



A black screen with white text

Description automatically generated











* After the whole process finishes, it will create an ‘amplify’ folder inside of the ‘CatCol’ project folder. Inside of it you will find the ‘backend’ folder, which will contain the ‘auth’ folder that contains all the authentication configurations, that we previously specified on the backend in the ‘Amplify Studio’.
* After the authentication features were introduced to the application, file storage needs to be set up as well. To do that, you need to navigate to the left-hand side menu of the ‘Amplify Studio’ again, and then press on the ‘Storage’ option.
* Then, you will be taken to the ‘File Storage’ page, where you should choose the ‘Create a new S3 bucket’ option and tick all the boxes for the ‘Signed-in users’ section.

A screenshot of a computer

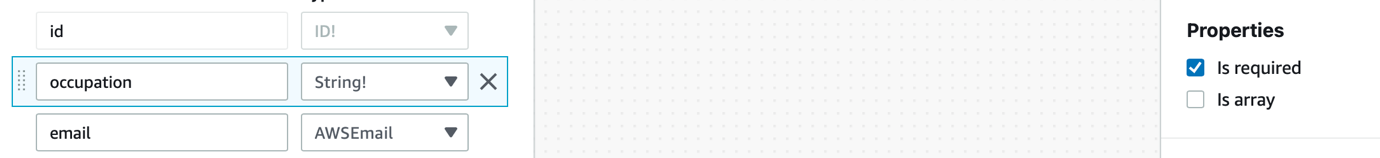
Description automatically generated

* Then, press the ‘Create bucket’ button, and wait for the AWS servers to deploy the changes.
* After that is done, go back to your terminal at the ‘CatCol’ React Native project folder, and call the ‘amplify pull’ command. Since, this is the second time you are pulling the changes from the backend, you don’t need to include any of the backend credentials when you’re calling this method, just call ‘amplify pull’ and that’s it. Calling this command will create a new folder called ‘storage’ inside of the ‘backend’ folder. (Use ‘sudo’ keyword before the command if you ever get permission errors)
* The last part of the backend setup in ‘Amplify Studio’ is to create data models and set up the database. To do that click on the ‘Data’ option in the left-hand side menu.
* This will take you to the ‘Data Modelling’ page, where you need to press on the ‘Add model’ button, and create a ‘User’ model as shown below:

A screenshot of a computer

Description automatically generated

* The ‘!’ symbol next to some of the types symbolises that it’s a required variable, so the ‘is required’ checkbox should be ticked.



* Similarly, the ‘[]’ symbols surrounding some types symbolises that they have to be arrays, so the ‘is array‘ and ‘allow null arrays’ checkboxes should be ticked for these variables.

A white background with black dots

Description automatically generated

* The next data model that needs to be made by clicking on the ‘Add model’ button is the ‘Chat’ model:

A screenshot of a computer

Description automatically generated

* The one after that is the ‘Message’ model:

A screenshot of a computer

Description automatically generated

* After the models are created, some relationships between them also need to be specified. To do this you need to click the ‘Add relationship’ button found at the bottom of each of the models.
* So, the first relationship to be made is going to be done by clicking the ‘add relationship’ button inside of the ‘User’ model, and choosing the related model to be ‘Chat’ and the model relationship to be ‘Many User to many Chat’.

A screenshot of a chat

Description automatically generated

* Click on the ‘add relationship button’ inside of the ‘User’ model again and choose the related model to be ‘Message’ and the model relationship to be ‘One User to many Message’.

A screenshot of a computer

Description automatically generated

* The last relationship needs to be made by clicking the ‘add relationship’ button inside of the ‘Chat’ model and choose the related model to be ‘Message‘ and the model relationship to be ‘One Chat to many Message’.

A screenshot of a chat

Description automatically generated

* After all the changes are made all the models should look in the following way:

Screens screenshot of a chat

Description automatically generated

A screenshot of a computer

Description automatically generated

* After all the models are set up correctly, you need to press the ‘save and deploy’ button and wait till the AWS servers deploy the changes. After that’s done, you need to go back to the terminal inside of the ‘CatCol’ React Native project and call the ‘amplify pull’ command again. This time it will create an ‘api’ folder inside of the ‘backend’ folder.
* After the ‘api’ folder is set up, all the mutations, queries, and subscriptions need to be pulled from the backend. To do that you have to call the ‘amplify codegen add’ command inside of the terminal from the ‘CatCol’ folder (use ‘sudo’ if needed) and answer the prompts that will be provided in the following way:

A black background with white text

Description automatically generated  






* After all of this is set up, the ‘schema.graphql’ file should be present in the ‘catcol’ folder that is inside of the ‘api’ folder. This file needs to be altered, since this change is not allowed to be made directly from ‘Amplify Studio’.
* So, you need to open the ‘schema.graphql’ file and later the ‘Message’ model to look like this: (If there are permission problems, then an alert should pop up asking if you want to make this changes as a super user, so choose yes.)

A screen shot of a computer

Description automatically generated

* Then, this change is only made on the front end so far, so it needs to be pushed to the backend too. So, to do that call the ‘amplify push’ command in the terminal from the ‘CatCol’ project folder. (Use ‘sudo’ if necessary)
* When prompted with some questions after calling the command, answer ‘yes’ or ‘y’ to all of them and wait until the AWS servers deploy the changes to the backend.

# Integrating the Source Code

* Replace the existing ‘App.js’ file inside of the ‘CatCol’ project folder with the file of the same name provided by the developer.
* Then, insert the ‘assets’, ‘components’, ‘navigation’, ‘pages’, and ‘customGraphQL’ folders into the ‘src’ folder, which is found inside of the ‘CatCol’ project folder.

# Run the Application

* Set up an android emulator <https://developer.android.com/studio/run/managing-avds.html> or plug in an android device <https://reactnative.dev/docs/0.73/running-on-device>.
* Navigate to the ‘CatCol’ React Native project folder and run the ‘npm start’ command and then press ‘a‘ to run the android version of the application.
* If you run into the ‘./gradlew app:installDebug -PreactNativeDevServerPort=8081 FAILURE: Build failed with an exception.’ error navigate to the ‘android’ folder inside of the ‘CatCol’ folder and run the ‘sudo chown -R $USER:$(id -gn) /Volumes/Macintosh\ HD\ -\ Data/CatCol’ command. Then go back to the ‘CatCol’ folder and try running the ‘npm start’ command again.

# Create an APK File

* The method of running the application described in the previous section only allows you to run the application locally, so on an emulator or on a device that’s plugged in to the computer.
* However, if you’d like to have an application that can be installed on any android device, you will have to generate an ‘apk’ file first, which then can be used for installing the app.
* To do this then, you first need to make a new folder called ‘assets’ inside of the ‘CatCol’ 🡪 ‘android’ 🡪 ‘app’ 🡪 ‘src’ 🡪 ‘main’ folder.
* Then, you need to call the ‘npx react-native bundle --platform android --dev false --entry-file index.js --bundle-output android/app/src/main/assets/index.android.bundle --assets-dest android/app/src/main/res’ command in the terminal inside of the ‘CatCol’ project folder and wait until all the necessary files have been copied to the ‘assets’ folder.
* After the command finishes executing, you need to navigate to the ‘android’ folder inside of the terminal by using the ‘cd android’ command.
* Then, inside of the ‘android’ folder from the terminal, the ‘./gradlew assembleDebug’ command must be called to generate the ‘apk’ file. This file will then be found inside of the ‘CatCol’ 🡪 ‘android’ 🡪 ‘app’ 🡪 ‘build’ 🡪 ‘outputs’ 🡪 ‘apk’ 🡪 ‘debug’ folder and the file will be called ‘app-debug.apk’, which could then be copied and used to install the application on any android device.
* However, this file can only be used to install the application in this manual way. If you’d like to obtain am ‘apk’ file that could be uploaded to the ‘Google Play Store’, you’d have to create a ‘signed apk’ file, which can be done by following the ‘Release APK’ section of the following tutorial <https://medium.com/geekculture/react-native-generate-apk-debug-and-release-apk-4e9981a2ea51>.