--

AlWeather is an Android application which shows weather based on user location.

It has options for forecasting weather for the next five days. The user can switch to different unit systems (European or American). The weather data is provided by World Weather Online API

---

Features

Display weather based on user location.

Weather forecast for 5 days

Supports European and American unit system

----

Installation

To fork this project you should have the latest Android Sdk and build tools installed.

If you don’t have Hugo plugin installed then you should gradle sync and a notification for installing Hugo plugin will appear. This plugin is used for writing debug logs. Then everything should work like a charm

Documentation

The application has a MainActivity which is responsible for opening two fragments, setuping Toolbar and Navigation Drawer. For the navigation drawer is used Material Drawer from Mike Penz. For the usage details check: <https://github.com/mikepenz/MaterialDrawer>.

TodayFragment serves for showing the weather today based on user current location. For retrieving user location is used FusedLocationProviderApi introduced in Google Play Services 7.0. <http://android-developers.blogspot.com/2015/03/google-play-services-70-places-everyone.html>

It gives optimal location, also here is introduced a standard mechanism to check that the necessary location settings are enabled for a given LocationRequest to succeed. If there are possible improvements, you can display a one touch control for the user to change their settings without leaving your app. This service is used also for reverse geocoding the user location to his address.

The communication with World weather online API is done using retrofit library from square. It makes developing the communication with the API nice and tight. Check this site for the usage http://square.github.io/retrofit/

For passing data through classes fragments activity is used EventBus from GreenRobot. Check this site for the usage <https://github.com/greenrobot/EventBus>

For the settings activity is used ActionBarActivty with PrefenceFragment.

For the about Fragment is used MaterialDialog library from <https://github.com/afollestad/material-dialogs>

For loading Images from the server is used Universal Image Loader (the best in town ☺), for its usage and implementation check <https://github.com/nostra13/Android-Universal-Image-Loader>

For writing debug logs only in debug mode without doing a lot of work (simple and clean ) is used Hugo a library from Jake Wharton. For its usage and installation check <https://github.com/JakeWharton/hugo>

For the future release I have saved Android Annotation to simplify the code, for more info check https://github.com/excilys/androidannotations

----

Building project

-----------

Dependecies

AppCompat

Support-v4

RecycleView-v7

Play-Services-Location-7.0

Retrofit

EventBus(GreenRobot)

Universal-Image-Loader

Crouton-Toast

MaterialDrawer

RobotoTextView

Material-Dialogs

**Publishing**

* Set proper versions in the main build script
* Set key hash of [World Weather Online API](http://www.worldweatheronline.com/)

Check build config fields in the main build script

Update text info in changelog/about

Developed by

Armando Shkurti

License

Bla bla

AlWeather

===============

AlWeather is an Android application which shows weather based on user location.

It has options for forecasting weather for the next five days. The user can switch to different unit systems (European or American). The weather data is provided by World [Weather Online API](http://www.worldweatheronline.com/api/local-city-town-weather-api.aspx)

## Fetaures

- Display weather based on user location

- Weather forecast for 5 days

- Supports European and American unit system

## Instalaltion

To fork this project you should have the latest Android Sdk and build tools installed.

If you don’t have Hugo plugin installed then you should gradle sync and a notification for installing Hugo plugin will appear. This plugin is used for writing debug logs. Then everything should work like a charm.

## Documentation

The application has a MainActivity which is responsible for opening two fragments, setuping Toolbar and Navigation Drawer. For the navigation drawer is used Material Drawer from Mike Penz. For the usage details check: https://github.com/mikepenz/MaterialDrawer.

TodayFragment serves for showing the weather today based on user current location. For retrieving user location is used FusedLocationProviderApi introduced in Google Play Services 7.0:

http://android-developers.blogspot.com/2015/03/google-play-services-70-places-everyone.html

It gives optimal location, also here is introduced a standard mechanism to check that the necessary location settings are enabled for a given LocationRequest to succeed. If there are possible improvements, you can display a one touch control for the user to change their settings without leaving your app. This service is used also for reverse geocoding the user location to his address.

The communication with World weather online API is done using retrofit library from square. It makes developing the communication with the API nice and tight. Check this site for the usage:

http://square.github.io/retrofit/

For passing data through classes fragments activity is used EventBus from GreenRobot. Check this site for the usage https://github.com/greenrobot/EventBus

For the settings activity is used ActionBarActivty with PrefenceFragment.

For the about Fragment is used MaterialDialog library from: https://github.com/afollestad/material-dialogs

For loading Images from the server is used Universal Image Loader (the best in town ), for its usage and implementation check:

https://github.com/nostra13/Android-Universal-Image-Loader

For writing debug logs only in debug mode without doing a lot of work (simple and clean) is used Hugo a library from Jake Wharton. For its usage and installation check: https://github.com/JakeWharton/hugo

For the future release I have saved Android Annotation to simplify the code, for more info check:

https://github.com/excilys/androidannotations

## Building Project

This chapter describes how to build APK with Gradle and prepare app for publishing.

You don't need to install Gradle on your system, because there is a Gradle Wrapper. The wrapper is a batch script on Windows, and a shell script for other operating systems. When you start a Gradle build via the wrapper, Gradle will be automatically downloaded and used to run the build.

Clone this repository

Open main build script /mobile/build.gradle and set constants and config fields as required (see below for more info)

Run gradlew assemble in console

APK should be available in /mobile/build/outputs/apk directory

Note: You will also need a "local.properties" file to set the location of the SDK in the same way that the existing SDK requires, using the "sdk.dir" property. Example of "local.properties" on Windows: sdk.dir=C:\\adt-bundle-windows\\sdk. Alternatively, you can set an environment variable called "ANDROID\_HOME".

Tip: Command gradlew assemble builds both - debug and release APK. You can use gradlew assembleDebug to build debug APK. You can use gradlew assembleRelease to build release APK. Debug APK is signed by debug keystore. Release APK is signed by own keystore, stored in /extras/keystore directory.

Signing process: Keystore passwords are automatically loaded from property file during building the release APK. Path to this file is defined in "keystore.properties" property in "gradle.properties" file. If this property or the file does not exist, user is asked for passwords explicitly.

## Dependecies

- AppCompat

- Support-v4

- RecycleView-v7

- Play-Services-Location-7.0

- Retrofit

- EventBus(GreenRobot)

- Universal-Image-Loader

- Crouton-Toast

- MaterialDrawer

- RobotoTextView

- Material-Dialogs

## Publishing

- Set proper versions in the main build script

- Set key hash of World Weather Online API

- Check build config fields in the main build script

- Update text info in changelog/about

## Special Thanks

- Romain Guy

- GreenRobot

- Sergey Tarasevich

- Jake Wharton

- Lucas Urbas

Loading...

## Developed by:

- Armando Shkurti - shkurtiarmando@gmail.com

## License

Feel free to use it in your Android apps and also include the license in your app. I would be grateful if you would inform me about its usage: shkurtiarmando(at)gmail(dot)com

Copyright 2015 Armando Shkurti

Licensed under the Apache License, Version 2.0 (the "License");

you may not use this file except in compliance with the License.

You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software

distributed under the License is distributed on an "AS IS" BASIS,

WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and

limitations under the License.