

Structure

- **model** contains code deals with data such as local database using **Room**, remote that send audio files to api developed using **flask** and retrieve corresponding text.
- **feature** screens that appear to user (home activity, SuralList, Listen Fragment, ..etc).
- **util**
utilities classes(**date operation**, **parser for json**, **some text processing**).

1. model

contains sub-packages

- **local** :- deals with local data using Room, contains Entities, Dao, Database class some entities
 - **AyahItem**, **SuraItem**.
- **remote** :- connect to api developed by **flask** using **Retrofit**
 - use **model** classes used to convert Responce of networking calls into Java Objects such as **MLResponce.java**.
 - use **MLModelService** to define **GET** and **POST** calls.
 - **ApiClient** create **singleton** instance from **Retrofit** with defing **baseUrl**.
- **repository** it uses **reposotory-like** pattern to **provide access to all data sources** (**local**, **remote**) from only one place.

2. feature.

each feature represent a screen to user, contains logic and ui for it.

- **home_Activity**: **home** & **launcher** activity for app. control fragments used with **PageSliding** such as (ListenFragment,SuralListFragment, .. etc). it include these fatures
 - **ListenFragment**: allow user to **select ayahs** and **listen** to them, with text is appeared on screen, user can control playback.
 - **SuralListFragment**: **display list of surahs** of quran, user click on sura to open, read it.
 - **TafseerDetails**: display tafseer of ayahs, allow user to select sura & ayah num to show its tafseer.
 - **TestQuran** : test save of user by two main methods:-
 - **Text**: which allow user to **write quran** and app will **evalute it** and get results with **score**
 - **Sound**: which allow user to **record ayahs** and upload it, then get results.
 - **test_sound**: test quran iwth sound contails
 - **RecordFragment**:- allow user to **record**.
 - **RecordList**: **show saved records** with score & **upload audio** to be **evaluated**.

- **ReadLog**: that store pages user read. Display every day with pages user read, its count.
- **Bookmark**: **display bookmarks** user saved while reading and allow **deleting** & **navigate** to page saved with bookmark.
- **ayahs_search**: search of ayahs with display **count** of results, get with some options:-
 - **show tafseer** of ayah.
 - **play sound** of it.
 - **open page** that contains that ayah.
- **Download** :- download **Quran** & **Tafseer** and **ayahs audio** files.
- **Setting** :- select modes of read:-
 - **Night mode**.
 - **Normal mode**:- here user **select background** color for ayahs.
- **scores** :- **display score** of user, **collect** from quran test tries.

3. Util

some util classes used as **helper classes**.

- **Constants** : contain **constants** used in app such as **KEYS** for **intents**, **bundle**.
- **Data** : contains **list of surahs** names.
- **DateOperation** : some date operation such as **getCurrentDate** formatted, get current day, month, year.
- **TestText** :- used to **evalute texts** and provide score.
- **Util** : some **util functions** such as (parse json, create dialoge, span texts, check internet connection, ...etc).