**Hope Android Project Structure**

**1 Project structure**

**a) HorizontalListView**

This is the library code that used to display Assignment icons on 教案界面

**b) CalendarLibrary**

Calendar Widget display on 首页. I have customized the source to adjust the size and color to fix in our table UI

Source: calendarlibrary

Jar: com.android.support:support-v4:19.+

Android support library

.

Jar: com.android.support:support-v13:19.+

Android support library

Jar: com.squareup:otto:1.3.4

Square Event bus, wildely used in the project

Jar: com.squareup.okhttp:okhttp:2.0.0

Square httpClient

Jar: com.squareup.okhttp:okhttp-urlconnection:2.0.0

Square httpClient

Jar: com.squareup.retrofit:retrofit:1.6.1

Square Restful library , used to rest client library and communicate with backed server .

Jar: com.google.code.gson:gson:2.2.4

Google json parser

Jar: com.j256.ormlite:ormlite-android:4.48

lightweighted orm for db access

Jar: org.slf4j:slf4j-api:1.7.6

logging

Jar: com.github.tony19:logback-android-classic:1.1.1-2

logging library

Jar: org.java-websocket:Java-WebSocket:1.3.0

websocket , will be using

Jar : com.squareup.dagger:dagger-compiler:1.2.1

Code Injection library

Jar: com.squareup.dagger:dagger:1.2.1

Jar: com.squareup.picasso:picasso:2.2.0

**2) Major components**

**a) HopeLocalServices**

This is a long running service so that we can hold all the necessary information such as userSession cross different activities. The reason to have an Android service is that there are requirement that we need to jump between different Uis ( main page, topic assignment etc) certain information (current course , topci , assignement ) need to be shared and keep in memory . Use a service to allow different activities access the info easily .

Service also responsible to communicates with backend service , in case there's updated from backend , service would update the userSession and UI components would be able to access to synced instance . Other wise , we have to always pull update and pass around same object by using Intent and serialize/deseriize necessary objects back and forth which is cumbersome.

**b) UserSessionContect**

This is an object that used to keep track status once the user logged in It holds following info which are critical for the application

PersonalInfo : basic user profile , it is populated after user successfully loggin .

cousreList: the course list that associated with the login user. For student there might be course for differnet subject . For teacher it could be course for different classes

todoItemList : all todoitem on the main page

planList : plan is always associated to the course .From backend side topic is tied to plan In our code for consistency I tie all the backend info to course since for student it doesn't make sense to expose plan concept to them

assignmentQuestion , this is a map to hold all the question associated with current assignment in memory . The reason of using this is once we retrieved questions from backend . we would like to save the round trip to backend again for the same question. But we are based on assumption that there's only one active backend web session . otherwise . if two clients access the same session could cause everythign out of sync.

**c) Restfull services**

We are using Square Retrofit to deal with all restful call to backend server .

* **Service Endpoint**

Service End point is just regular url , we support both http and https

* **Session Management**

We need to maintain session cross different rest call and download request

CookieManager public void put(URI uri, Map<String, List<String>> stringListMap)

is used to capture session

RequestInterceptor public void intercept(RequestFacade request) is overridden to inject session .

* **SSL Enable**

Tablet client is Two-way SSL authentication. But not enabled right now .

* **Sync vs Async call**

Retrofi support both Async and sync call . We used both . For async we need to provide a callback object . And thread is managed by Retrofit thread pool

* API

@GET("/rest/grade")

List<Grade> listGrades ();

/\*

@GET("/users/{user}/repos")

List<Repo> listRepos(@Path("user") String user);

\*/

@POST("/rest/logon/{user}/{password}")

LogonStatus logon(@Path("user") String user,@Path("password") String password);

@GET("/rest/account")

AccountStatus getMyAccount();

@GET("/rest/course")

List<RestCourseItem> getMyCourse();

@GET("/rest/to\_do")

List<RestTodoItem> getTodoList();

@GET("/rest/to\_do")

void getTodoList(TodoRequestCallback<List<RestTodoItem>> callback);

@POST("/rest/to\_do")

void addTodoItem(@Body String message,TodoRequestCallback<String> callback);

@DELETE("/rest/to\_do/{id}")

void deleteTodoItem(@Path("id") String id,TodoRequestCallback<String> callback);

@PUT("/rest/to\_do/{id}/close")

void closeTodoItem(@Path("id") String id,TodoRequestCallback<String> callback);

@PUT("/rest/to\_do/{id}/reopen")

void reopenTodoItem(@Path("id") String id,TodoRequestCallback<String> callback);

@PUT ("/rest/to\_do/{id}")

void updateTodoItem(@Path("id") String id,@Body String message,TodoRequestCallback<String> callback);

@GET("/rest/assignment\_alert")

List<RestAssignmentItem> getAssignmentAlert();

// http://72.93.93.62/rest/global/plan

@GET("/rest/global/plan")

List<RestMyPlan> getPublisedPlan();

@GET("/rest/plan")

List<RestMyPlan> getMyPlan();

// /rest/global/topic/1

@GET("/rest/global/topic/{topicId}")

RestTopic getTopic(@Path("topicId") String topicId);

// http://72.93.93.62/rest/student\_question/4524

@GET("/rest/student\_question/{id}")

void getStudentQuestion(@Path("id") String qId, GetQuestionCallback<RestStudentQuestionAndAnswer> callback);

// @Headers({

//

// "Content-Type: charset=UTF-8; application/json"

// })

@PUT("/rest/student\_question/{asnId}")

void submitStudentQuestion(@Path("asnId") int asnId, @Body String[] answers, SubmitQuestionCallback<String> callback);

@GET ("/rest/student\_question\_with\_official\_answers/{sqId}")

void getQuestinWithOfficalAnswer(@Path("sqId")String sqId, GetQuestionCallback<RestStudentQuestionAndAnswer> callback);

@PUT ("/rest/student\_assignment/{asnId}")

void submitStudentAssignment(@Path("asnId") String asnId,SubmitAssignmentCallback<String> callback);

@GET("/rest/global/q\_set/{qSetId}")

void getQSet(@Path("qSetId") String qSetId, GetQSetCallback<RestQset> callback);

@GET("/rest/question/{id}")

void getQSetQuestion(@Path("id") String id,GetQuestionCallback<RestStudentQuestionAndAnswer> callback );

@GET ("/rest/global/document/topic/{topicId}")

void getDocumentByTopicId(@Path("topicId") String topicId ,GetDocumentCallback<List<RestDocument>> callback);

**d)** **HopeLeftPanelActivity**

HopeLeftPanelActivity extend FragmentActivity . It is used to display common navigation panel cross all the activities.

It also contains a expand menu which is used to display all the course that associated with a new user . All the Activities ( Main page , topic and HomeWork) extends from HopeLeftPanelActivity.

e) **package com.lanking.mainpage**

This package contains all the source code for the UI component on main page

Calendar Widget , Time table . Todo List and , Latest update



**f) home.android.topic**

Source code for following topic page .

1. Topic content
2. horizontal list view for homework
3. right side sliding menu

You can swipe to display different topic .

Also you can switch topic by clicking menu item on right menu

By click homework icon it will open homework page



