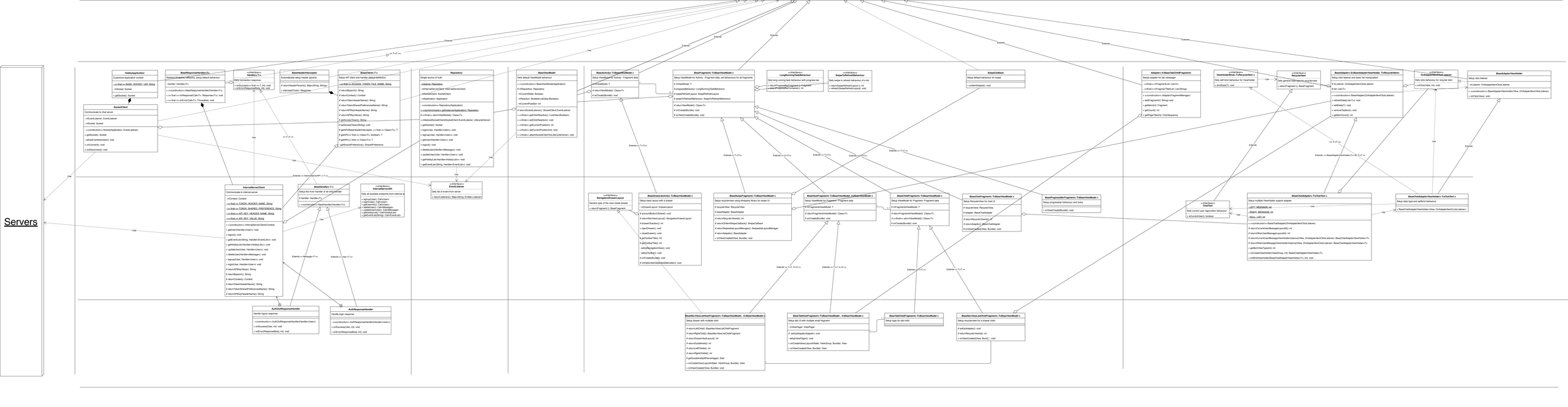
Third Party API + Android Classes


Servers
HobbyApplication
Customize application context
- cx:final cx:BASE_SOCKET_URI: String
- mSocket: Socket
+ getSocket(): Socket
SocketClient
Communicate to chat server
- mEventListener: EventListener
- mSocket: Socket
+ <constructor> SocketApplication: EventListener
+ getSocket(): Socket
- setUpEventsSockets(): void
+ onConnect(): void
+ onDisconnect(): void
BaseResponseHandler<T>
Retrieve Response HTTPBOOL setup default behaviour
- handler: Handler<T>
+ <constructor> BaseResponseHandler(Handler<T>)
+ << final >> onResponse(Call<T>, Response<T>): void
+ << final >> onError(Call<T>, Throwable): void
<<interface>>
Handler<T>
Daily Connection response
+ onSuccess(var first >> T, int): void
+ onError(ResponseBody, int): void
BaseHeaderInterceptor
Automatically setup header params
returnHeaderParams(): Map<String, String>
+ intercept(Chain): Response
BaseClients T >
Setup API client and header setup/setMSKEY
- cxs:final cx:ACCESS_TOKEN_FILE_NAME: String
returnBaseUrl(): String
returnContext(): Context
returnSharedHeaderName(): String
returnSharedPreferenceserviceName(): String
returnVPAKyHeaderName(): String
returnVPAKyValue(): String
getAccessToken(): String
setAccessToken(String): void
getAPIDisposeInterceptor(): << final >> Class<?>> T
getAPI(var first >> Class<?>, boolean): T
getAPI(var first >> Class<?>): T
- getSharedPreference(): SharedPreference
Repository
Single source of truth.
- instance: Repository
- mInternalServerClient: InternalServerClient
- mMsMSKClient: SocketClient
- mApplication: Application
+ <constructor> Repository(Application)
+ <static/synchronized> getInstance(Application): Repository
# << final >> returnSharedModel(): Class<?>
+ << final >> releaseSocketClient(SocketClient EventListener, LifecycleOwner): void
+ getSocket(): Socket
+ signUp(User, Handler<User>): void
+ signIn(User, Handler<User>): void
+ getUser(Handler<User>): void
+ logout(): void
+ deleteUserHandler(Message): void
+ updateUserUser, Handler<User>): void
+ getMsMsUsl-Handler<MsMsUsl>): void
+ getEventLatLng(String, Handler<EventLatln>): void
BaseViewModel
Only default ViewModel behaviour
+ << constructor >> BaseViewModel(Application)
# # Repository: Repository
- mCurrentState: Boolean
- mReaction: MutableLiveData<Boolean>
- mCurrentPosition: int
<< final >> socketListener(): SocketClient.EventListener
+ << final >> getOldReaction(): LiveData<Boolean>
+ << final >> setOldReaction(): void
+ << final >> getCurrentPosition(): int
+ << final >> setCurrentPosition(int): void
+ << final >> attachSocketClientToLifeCycleOwner(): void
BaseActivity< TvBaseFlowModel >
Setup ViewBinding/FC Activity - Fragment data
mFlowModel: T
<< final >> initFlowModel(): Class<?>
<< final >> onCreate(Bundle): void
BaseFragment< TvBaseViewModel >
Setup ViewModel for Activity - Fragment data, set behaviours for all fragments
<< final >> initViewModel(): T
progressBehaviour: LongRunningTaskBehaviour
swipeRefreshLayout: SwipeRefreshLayout
swipeToRefreshBehaviour: SwipeToRefreshBehaviour
<< final >> initViewModel(): Class<?>
<< final >> onCreate(Bundle): void
<< final >> onViewCreated(Bundle): void
<<interface>>
LongRunningTaskBehaviour
Daily long running task behaviour with progress bar
+ << final >> showProgressBar(Fragment or Fragment, int, int): void
+ << final >> dismissBar(Container): void
<<interface>>
SwipeToRefreshBehaviour
Daily swipe to refresh behaviour of a list
+ << final >> setupSwipeRefreshLayout(): void
+ << final >> setRefreshing(boolean, int): void
SwipeCallback
Setup default behaviour for swipe
+ << final >> onItemSwiped(): void
Adapter< VvBaseTabChildFragments >
Setup adapter for tab viewpager
+ << final >> mFragmentList: List<Fr>
+ << final >> mFragmentTitleList: List<String>
+ << constructor >> AdapterFragmentManager(FragmentManager)
+ addFragment(Y, String): void
+ getItem(int): Fragment
+ getCount(): int
+ getPageTitle(int): CharSequence
<<interface>>
ViewHolderBinds< TvRecyclerItem >
Daily self-bind behaviour for ViewHolder
+ bind(Item?): void
<<interface>>
RecyclerItem
Only general data type for recyclerview
+ << final >> mTypeID(): int
+ << final >> mTypeID(): BaseFragment
BaseAdapter< EvBaseAdapterViewHolder, TvRecyclerItem>
Setup click listener and basic list manipulation
mListener: OnAdapterItemClickListener
list: List<T>
+ << constructor >> BaseAdapter(OnAdapterItemClickListener, listener)
+ setNewDataList(List<T>): void
+ addData(T): void
+ removeTo(item): void
+ getItemCount(): int
<<interface>>
OnAdapterItemClickListener
Daily click behaviour for recycler item
+ onClick(View, int): void
BaseAdapterViewHolder
Setup click listener
+ << constructor >> BaseAdapterViewHolder(View, OnAdapterItemClickListener)
+ onClick(View): void
<<interface>>
InternalServerClient
Communicate to internal server
- mContext: Context
- << final >> TOKEN_HEADER_NAME: String
- << final >> TOKEN_SHARED_PREFERENCE: String
- << final >> API_KEY_HEADER_NAME: String
- << final >> API_KEY_VALUE: String
+ << constructor >> InternalServerClient(Context)
+ getUserHandler(User): void
+ logout(): void
+ getEventLatln(String, Handler<EventLatln>): void
+ getHbbyUsl-Handler<HbbyUsl>): void
+ updateUserUser, Handler<User>): void
+ deleteUserHandler(Message): void
+ signUpUser, Handler<User>): void
+ signIn(User, Handler<User>): void
+ returnVPAKyValue(): String
+ returnBaseUrl(): String
+ returnContext(): Context
+ returnSharedHeaderName(): String
+ returnSharedPreferenceserviceName(): String
+ returnVPAKyHeaderName(): String
BaseHandlers T >
Setup the inner handler of all chq/MsSer
handler: Handler<T>
+ << constructor >> BaseHandler(Handler<T>)
<<interface>>
InternalServerAPI
Only all possible endpoints from internal sa
+ signup(User): CallcUser>
+ signIn(User): CallcUser>
+ getUser(): CallcUser>
+ getEvents(): CallcMessage>
+ updateUser(User): CallcMessage>
+ deleteUser(): CallcMessage>
+ getEventLatln(String): CallcEventLatln>
<<interface>>
EventListener
Daily bit of event from server
+ << return.interface >> Map<String, Emitter.Listener>
AuthOtpResponseHandler
Handle logout response
+ << constructor >> AuthOtpResponseHandler(Handler<User>)
+ onSuccess(User, int): void
+ onError(ResponseBody, int): void
AuthPsResponseHandler
Handle login response
+ << constructor >> AuthPsResponseHandler(Handler<User>)
+ onSuccess(User, int): void
+ onError(ResponseBody, int): void
<<interface>>
NavigationDrawerLayout
Generic type of the view inside drawer
+ << return.Fragment >>: BaseFragment
BaseDrawerActivity< TvBaseViewModel >
Setup basic layout with a drawer
- mDrawerLayout: DrawerLayout
<< return.BottomSheet >>: void
<< return.View.layout >>: NavigationDrawerLayout
drawerOpen(): int
+ << open.Drawer >>: void
+ << closeDrawer >>: void
+ # getToolbarTitle(): int
+ # getToolbarTitle(): int
- setUpNavigationView(): void
- setUpToolbar(): void
<< onCreate.bundle >>: void
<< onOptionItemSelected.MenuItem >>: void
BaseSwipeFragment< TvBaseViewModel >
Setup recyclerview using thirdparty library for swipe UI
recyclerView: RecyclerView
baseAdapter: BaseAdapter
<< return.RecyclerView >>: int
<< return.OnItemSwipeCallback >>: SwipeCallback
<< return.SwipeLayoutManager >>: SwipeLayoutManager
<< return.Adapter >>: BaseAdapter
+ << onViewCreated.View, Bundle >>: void
BaseHostFragments< TvBaseViewModel, VpBaseFlowModel >
Setup ViewModel for Fragment / Fragment data
mFragmentViewModel: T
<< return.FragmentViewModel >>: Class<?>
+ << onCreate.Bundle >>: void
BaseChildFragments< TvBaseViewModel >
Setup ViewModel for Fragment / Fragment data
mFragmentViewModel: T
<< return.FragmentViewModel >>: Class<?>
<< final >> << return.ViewModel >>: Class<?>
+ << onCreate.Bundle >>: void
BaseChatFragments< TvBaseViewModel >
Setup RecyclerView for chat UI
recyclerView: RecyclerView
adapter: BaseChatAdapter
<< return.RecyclerView >>: int
<< return.Adapter >>: BaseChatAdapter
<< onViewCreated.View, Bundle >>: void
BaseProgressBarFragment< TvBaseViewModel >
Setup progressbar behaviour and locks
+ << onViewCreated.Bundle >>: void
<<interface>>
ChatTest
Daily current user registration behaviour
+ << current.bool >>: boolean
BaseChatAdapter< TvChatTest >
Setup multiple ViewHolder support adapter
- LEFT_MESSAGE: int
- RIGHT_MESSAGE: int
- NULL_LIST: int
+ << constructor >> BaseChatAdapter(OnAdapterItemClickListener)
# # return.CurrentUserMessage.layoutId(): int
# # return.OtherUserMessage.layoutId(): int
# # return.CurrentUserMessageViewHolderInstanceView, OnAdapterItemClickListener): BaseChatAdapterViewHolder<T>
# # return.OtherUserMessageViewHolderInstanceView, OnAdapterItemClickListener): BaseChatAdapterViewHolder<T>
+ getItemViewType(int): int
+ << onCreateViewHolder.ViewGroup, int >>: BaseChatAdapterViewHolder<T>
+ << onBindViewHolder.BaseChatAdapterViewHolder<T>, int >>: void
BaseChatAdapterViewHolder< TvChatTest >
Setup data type and setBind behaviour
+ << BaseChatAdapterViewHolder(View, OnAdapterItemClickListener) >>
BaseNoteViewListHostFragments< TvBaseViewModel , VvBaseViewModel >
Setup drawer with multiple child
return.LeftChild(): BaseNoteViewListChildFragment
return.RightChild(): BaseNoteViewListChildFragment
return.CreateNoteLayout(): int
return.Guidelines(): int
return.LeftChild(): int
return.RightChild(): int
getGuidelineSplitPercentage(): float
+ << CreateViewLayout.inflater, ViewGroup, Bundle >>: View
+ << onViewCreated.View, Bundle >>: void
BaseTabHostFragments< TvBaseViewModel, VvBaseViewModel >
Setup tab UI with multiple small fragment
- << initialPager: ViewPager
setUpAdapterAdapter(): void
- << setupViewPager(): void
+ << onCreateView.LayoutInflater, ViewGroup, Bundle>: View
+ << onViewCreated.View, Bundle>: View
BaseTabChildFragments< TvBaseViewModel >
Setup logic for tab child
BaseNoteViewListChildFragments< TvBaseViewModel >
Setup recyclerview for a drawer child
setUpAdapter(): void
recyclerRecyclerView(): int
+ << onViewCreated.View, Bundle >>: void
Extends
Extends << TvVP >>
Extends << TvFlow >>
Extends << TvFragment >>
Extends << TvChildFragment >>
Extends << TvChildFragment >>
Extends << TvChild