**DOCUMENTATION DEVELOPPEUR**

La documentation décrira toutes les classes et fonctions actuellement utilisées qui compose l’application.

Liste des classes :

Classe **CustomAppBar** :

class CustomAppBar extends StatefulWidget implements PreferredSizeWidget

Fonction :

preferredSize : final Size preferredSize;

data : final Response<Map<String, dynamic>>? data;

createState : \_CustomAppBarState createState() => \_CustomAppBarState();

Value :

preferredSize : final Size preferredSize;

data : final Response<Map<String, dynamic>>? data;

Classe **CustomAppState** :

class \_CustomAppBarState extends State<CustomAppBar>

Fonction :

redirectSettingPage : void redirectSettingPage(data)

gestureLogo : Widget gestureLogo()

build : Widget build(BuildContext context)

Classe **MyApp** :

class MyApp extends StatelessWidget

Fonction :

build : Widget build(BuildContext context)

Classe **WebviewRedditech** :

class WebviewRedditech extends StatelessWidget

Fonction :

build : Widget build(BuildContext context)

Classe **ActionBar** :

class ActionBar extends StatefulWidget

Fonction :

createState : \_ActionBarState createState() => \_ActionBarState();

Value :

upvote : String upvote;

downvote : String downvote;

Subreddit : String subreddit

nbComment : String nbComment;

Classe **\_ActionBarState** :

class \_ActionBarState extends State<ActionBar>

Fonction :

build : Widget build(BuildContext context)

\_onPressedUpvote : void \_onPressedUpvote()

\_onPresseDownvote : void \_onPressedDownvote()

Classe **Subreddit** :

class Subreddit extends StatefulWidget

Fonction :

createState : \_SubredditState createState() => \_SubredditState();

Value :

subreddit : final String subreddit;

Classe **\_SubredditState** :

class \_SubredditState extends State<Subreddit>

Fonction :

build : Widget build(BuildContext context)

Classe **Settings** :

class Settings extends StatefulWidget

Fonction :

createState : \_SettingsState createState() => \_SettingsState();

Value :

data : final Map<String, dynamic>? data;

Classe **\_SettingsState** :

class \_SettingsState extends State<Settings>

Fonction :

build : Widget build(BuildContext context)

Classe **Profile** :

class Profile extends StatelessWidget

Fonction :

build : Widget build(BuildContext context)

Value :

data : final Response<Map<String, dynamic>>? data;

Classe **HomeFeed** :

class HomeFeed extends StatefulWidget

Fonction :

createstate : HomeFeedState createState() => HomeFeedState();

Value :

data : final dynamic data;

Classe **HomeFeedState** :

class HomeFeedState extends State<HomeFeed>

Fonction :

imagePost : Widget imagePost(i)

usernamePost : Widget usernamePost(i)

descriptionPost : Widget descriptionPost(i)

prefixNamePost : Widget prefixNamePost(i)

displayInfoProfil : Widget displayInfoProfil(i)

cardPost : Widget cardPost(i)

build : Widget build(BuildContext context)

Classe **Redditech** :

class Redditech extends StatefulWidget

Fonction :

createState : \_RedditechState createState() => \_RedditechState();

Value :

userData : final Response<Map<String, dynamic>>? userData;

postsData : final Response<Map<String, dynamic>>? postsData;

Classe **\_RedditechState** :

class \_RedditechState extends State<Redditech>

Fonction :

gestureLogo : Widget gestureLogo()

build : Widget build(BuildContext context)

Classe **AccountPage** :

class AccountPage extends StatefulWidget

Fonction :

createState : \_AccountPageState createState() => \_AccountPageState();

Value :

data : final dynamic data;

Classe **\_AccountPageState** :

class \_AccountPageState extends State<AccountPage>

Fonction :

build : Widget build(BuildContext context)

Classe **Data** :

class Data extends StatefulWidget

Fonction :

createState : DataState createState() => DataState();

Value :

data : final dynamic data;

i : final int i;

Classe **DataState** :

class DataState extends State<Data>

Fonction :

initState : void initState()

getUrlType : UrlType getUrlType(String url)

build : Widget build(BuildContext context)

dispose : void dispose()

Value :

Controller : late VideoPlayerController controller;

initializeVideoPlayerFuture : late Future<void> initializeVideoPlayerFuture;