

Documentation Technique

Authentification	2
Images	3
Récupération des Images	3
Affichage des Images	3
Fonctions Likes	4
Fonction Commentaires	5
Search	5
Barre de recherche	5
Filtres	6
Favoris	7
Récupération des Images	7
Mettre ou enlever une image des favoris	7
Upload	9
Récupération Images	9
Fonction upload de l'image	9

Authentification

login.dart :

Cette fonction permet d'utiliser l'api qui nous permet de nous authentifier sur imgur

```
52 void login() {  
53   Navigator.of(context).push(MaterialPageRoute(  
54     builder: (BuildContext context) => new WebViewScaffold(  
55       url: "https://api.imgur.com/oauth2/authorize?client_id=" +  
56         clientId +  
57         "&response_type=token&state=DEV",  
58       appBar: new AppBar( backgroundColor: Color(0xFFf1bb76e),title: new Text("Login to Imgur...")))); // WebViewScaffold // MaterialPageRoute  
59   }  
}
```

Cette partie de code dans login.dart permet de récupérer certaines informations dans l'url après s'être loguer telles que le token et le username de la personne

```
13 final flutterWebViewPlugin = new FlutterWebViewPlugin();  
14 StreamSubscription _onDestroy;  
15 StreamSubscription<WebViewStateChanged> _onStateChanged;  
16  
17 @override  
18 void dispose() {  
19   _onDestroy.cancel();  
20   _onStateChanged.cancel();  
21   flutterWebViewPlugin.dispose();  
22   super.dispose();  
23 }
```

```
25 @override  
26 void initState() {  
27   super.initState();  
28  
29   flutterWebViewPlugin.close();  
30  
31   _onStateChanged =  
32   flutterWebViewPlugin.onStateChanged.listen((WebViewStateChanged state) {  
33     if (state.url.startsWith("https://imgur.com/?state=DEV") ||  
34     state.url.startsWith("https://m.imgur.com/?state=DEV")) {  
35       RegExp regExpToken = new RegExp(r"#access_token=(.*)");  
36       RegExp regExpUsername = new RegExp(r"&account_username=(.*)");  
37       token = regExpToken.firstMatch(state.url)?.group(1);  
38       token = token.split("&")[0];  
39       username = regExpUsername.firstMatch(state.url)?.group(1);  
40       username = username.split("&")[0];  
41       Navigator.of(context).pop(true);  
42       Navigator.of(context).push(  
43         new MaterialPageRoute(builder: (context) => MyHomePage()),  
44       );  
45       ScaffoldMessenger.of(context).showSnackBar(SnackBar(  
46         content: Text("Connected as $username"),  
47       )); // SnackBar  
48     }  
49   });  
50 }  
51 }
```

Images

Récupération des Images

imports.dart :

Cette fonction permet de récupérer dans une liste tout les données des images récupérées, elle prend en paramètre l'url qui vas nous indiquer quel image vas être récupérée ainsi que les headers indispensables à la récupération des images.

```
45 // fonction fetch basique
46 Future<List<dynamic>> fetch(
47   String url, Map<String, String> headersParams) async {
48   //SharedPreferences prefs = await SharedPreferences.getInstance(); à implémenter plus tard
49   var result = await http.get(Uri.parse(url), headers: headersParams);
50   if (result.statusCode == 200) {
51     return json.decode(result.body)['data'];
52   } else {
53     return ["ErrorCode : ${result.statusCode}"];
54   }
55 }
```

home.dart :

Paramètre (url,headers) utilisé pour récupérer les images affichées sur la page home

```
23 : 'https://api.imgur.com/3/gallery/hot/${filter}/day/0?showViral=true&mature=true&album_previews=false',
24 {"Authorization": 'Client-ID ' + clientId},
```

Affichage Les Images

home.dart :

Cette partie de code permet l'affichage des images elle retourne le widget card qui permet son affichage avec le style et les autres éléments tels que les likes et commentaires.

```
17 return Scaffold(
18   body: Container(
19     child: FutureBuilder<List<dynamic>>(
20       future: fetch(
21         search.trim().isEmpty
22           ? 'https://api.imgur.com/3/gallery/search/${filter}/day/0?q=${search}'
23           : 'https://api.imgur.com/3/gallery/hot/${filter}/day/0?showViral=true&mature=true&album_previews=false',
24         {"Authorization": 'Client-ID ' + clientId},
25       builder: (BuildContext context, AsyncSnapshot snapshot) {
26         if (snapshot.hasData) {
27           snapshot.data.removeWhere((i) => ((i["Homes"] != null &&
28             i["Homes"].length != 0 &&
29             i["Homes"][0]["type"].contains('mp4')) ||
30             (i["cover"] == null)));
31           return ListView.builder(
32             padding: EdgeInsets.all(8),
33             itemCount: snapshot.data.length,
34             itemBuilder: (BuildContext context, int index) {
35               return card(snapshot, index, context);
36             }); // ListView.builder
37         } else
38           return Card(child: Text("Pas d'Homes retrouvées"));
```

imports.dart :

Dans le widget card c'est la ligne image (87) qui permet l'affichage de l'image

```

81  return Card(
82    child: Container(
83      decoration: BoxDecoration(
84        color: Color(0xFF2c2f34),
85      ), // BoxDecoration
86    child: Column(children: <Widget>[
87      Image.network(links(snapshot.data[index])),

```

et pour afficher l'image il faut également utiliser la fonction links qui récupère l'url de l'image qui est une des donnée récupérée par la fonction fetch

```

32  // récupérer le lien d'une image venant de l'api
33  String links(Map<dynamic, dynamic> gallery) {
34    if (gallery["images"] == null) {
35      if (gallery['cover'] == null) {
36        return gallery["link"].toString();
37      } else {
38        return "https://i.imgur.com/${gallery['cover']}.${gallery['type'].split("/")[1]}";
39      }
40    } else {
41      return gallery["images"][0]["link"].toString();
42    }
43  }

```

Fonctions Likes

imports.dart :

Cette fonction permet de récupérer le nombre de likes sur une image et retourne une string avec ce nombre

```

10  // récupérer le nombre de likes
11  String likes(Map<dynamic, dynamic> gallery) {
12    if (gallery['ups'] == null) {
13      return "0";
14    } else if (gallery['ups'] < 1000) {
15      return gallery["ups"].toString();
16    }
17    var likes = (gallery["ups"] / 1000).floor();
18    return likes.toString() + "k";
19  }

```

Cette fonction est présente dans le widget card qui permet d'afficher le nombre de likes

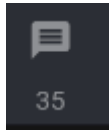
```

149  ), // IconButton
150  Text(
151    likes(snapshot.data[index]),
152    style: TextStyle(color: Color(0xFF8e9094)),
153  ) // Text

```

Fonction Commentaires

Affichage des commentaires



imports.dart :

Cette fonction permet de récupérer le nombre de commentaires sur une image et retourne une string avec ce nombre

```
21 // récupérer le nombre de commentaires
22 String comment(Map<dynamic, dynamic> gallery) {
23   if (gallery['comment_count'] == null) {
24     return "0";
25   } else if (gallery['comment_count'] < 1000) {
26     return gallery["comment_count"].toString();
27   }
28   var likes = (gallery["comment_count"] / 1000).floor();
29   return likes.toString() + "k";
30 }
```

Cette fonction est présente dans le widget card qui permet d'afficher le nombre de commentaires.

```
107 ), // IconButton
108 Text(
109   comment(snapshot.data[index]),
110   style: TextStyle(color: Color(0xFF8e9094)),
111 ) // Text
```

Search

Barre de recherche

main.dart :

c'est cette partie qui vas permettre d'écrire et récupérer des mots clé pour faire une recherche.

```

124     Padding(
125       padding: EdgeInsets.only(right: 20.0),
126       child: GestureDetector(
127         onTap: () {
128           setState(() {
129             if (this._searchIcon.icon == Icons.search) {
130               this._searchIcon = new Icon(Icons.close);
131               this._appBarTitle = new TextField(
132                 controller: _filterListener,
133                 decoration:
134                   new InputDecoration(hintText: 'Search by $filter'),
135               ); // TextField
136             } else {
137               this._searchIcon = new Icon(Icons.search);
138               this._appBarTitle = new Text('Imgur');
139               _filterListener.clear();
140             }
141           });
142         },
143         child: _searchIcon,
144       )), // GestureDetector // Padding
145     myPopupMenu(),

```

home.dart:

Cette partie du code permet de changer l'url sur laquelle on récupère les images. On place dans cette nouvelle url les mots clé écrits dans la barre de recherche, l'url change en fonction de l'état de la barre de recherche si elle est vide ou non.

```

21     search.trim().isEmpty
22       ? 'https://api.imgur.com/3/gallery/search/$filter/day/0?q=$search'
23       : 'https://api.imgur.com/3/gallery/hot/$filter/day/0?showViral=true&mature=true&album_previews=false',
24     {"Authorization": 'Client-ID ' + clientId}),

```

Filter

main.dart :

Cette partie crée une popup qui permet la sélection d'un filtre

home.dart:

Cette partie du code permet de changer l'url sur laquelle on récupère les image et ou on place dans cette nouvelle url le filtre sélectionné.

```

21     search.trim().isEmpty
22       ? 'https://api.imgur.com/3/gallery/search/$filter/day/0?q=$search'
23       : 'https://api.imgur.com/3/gallery/hot/$filter/day/0?showViral=true&mature=true&album_previews=false',
24     {"Authorization": 'Client-ID ' + clientId}),

```

```

59  return PopupMenuButton(
60      icon: Icon(Icons.filter_alt),
61      onSelect: (value) {
62          setState(() {
63              switch (value) {
64                  case 1:
65                      filter = "viral";
66                      break;
67                  case 2:
68                      filter = "top";
69                      break;
70                  case 3:
71                      filter = "time";
72                      break;
73                  default:
74                      }
75              });
76      },

```

Favoris

Récupération des Images

L'affichage des images des favoris est identique à celle affichée sur la page home expliquée plus haut dans la documentation le seul qui change sont les paramètres envoyés à la fonction fetch.

favoris.dart :

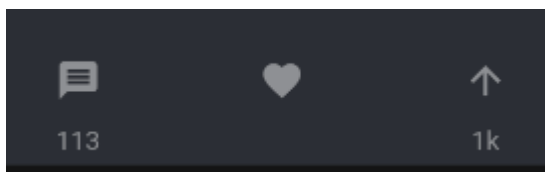
```

23      "https://api.imgur.com/3/account/$username/favorites/0/newest",
24      {"Authorization": "Bearer $token"}),

```

Mettre ou Enlever une images des favoris

L'icône cœur placé sous les images permet de mettre une image en favoris



imports.dart :

C'est cette fonction qui vas permettre la mise de l'image en favoris ainsi que sa suppression elle appelle la fonction favoris.

```

118 ~      onPressed: () async {
119 ~          if (token.isEmpty) {
120 ~              ScaffoldMessenger.of(context).showSnackBar(SnackBar(
121 ~                  content: Text(
122 ~                      "Essayez de vous connecter pour l'ajouter à vos favoris"), // Text
123 ~                  )); // SnackBar
124 ~          } else {
125 ~              var etatfavoris = await favoris(
126 ~                  getId(snapshot.data[index]),
127 ~                  {"Authorization": "Bearer $token"});
128 ~              if (etatfavoris == "favorited") {
129 ~                  ScaffoldMessenger.of(context).showSnackBar(SnackBar(
130 ~                      content: Text("Ajout de l'image aux favoris"),
131 ~                      )); // SnackBar
132 ~              } else {
133 ~                  ScaffoldMessenger.of(context).showSnackBar(SnackBar(
134 ~                      content: Text("Suppression du favori"),
135 ~                      )); // SnackBar
136 ~              }
137 ~          }
138 ~      },

```

C'est cette fonction qui va faire une requête à l'api, et qui, soit met l'image en favoris soit la supprime.

```

70 // récupérer les favoris du compte courant
71 Future<String> favoris(
72 ~     String imageHash, Map<String, String> headersParams) async {
73 ~     var result = await http.post(
74 ~         Uri.parse('https://api.imgur.com/3/image/$imageHash/favorite'),
75 ~         headers: headersParams);
76 ~     return (json.decode(result.body)['data']);
77 ~ }

```

Cette fonction appelle également une autre fonction quand elle est utilisée car pour utiliser la fonction favoris il faut lui donner l'id de l'image en favoris.

La fonction getId permet de récupérer l'id de la photo que l'on place en favoris ou enlève des favoris

```

57 // récupérer l'id d'une image venant de l'api
58 String getId(Map<dynamic, dynamic> gallery) {
59 ~     if (gallery["images"] == null) {
60 ~         if (gallery['cover'] == null) {
61 ~             return gallery["id"].toString();
62 ~         } else {
63 ~             return gallery['cover'];
64 ~         }
65 ~     } else {
66 ~         return gallery["images"][0]["id"].toString();
67 ~     }
68 ~ }

```


Upload

Recuperation Images

upload.dart

Cette fonction permet de récupérer une image qui est présente dans la galerie du téléphone

```
28  ✓  _openGallery(BuildContext context) async {
29      var picture = await ImagePicker().getImage(source: ImageSource.gallery);
30  ✓  this.setState(() {
31      |   imageFile = picture;
32      |   });
33      Navigator.of(context).pop();
34  }
```

Fonction upload de l'image

upload.dart :

Cette fonction permet de convertir l'image en base 64

```
118  bytesImage =
119  |   base64.encode(File(imageFile.path).readAsBytesSync());
```

Partie du code qui appelle la fonction upload et lui passe en paramètre l'image et les headers permettant l'envoi de l'image

```
121  ✓  |   upload(
122  |   |   bytesImage,
123  |   |   token.isEmpty
124  |   |   ? {"Authorization": "Client-ID $clientId"}
125  |   |   : {"Authorization": "Bearer $token"});
126  |   },
```

Cette fonction appelle l'api permettant l'upload de l'image

```
19  Future<dynamic> upload(
20  ✓  |   String imageUrl, Map<String, String> headersParams) async {
21  ✓  |   await http.post(Uri.parse('https://api.imgur.com/3/image'),
22  |   |   headers: headersParams, body: {'image': imageUrl});
23  |   }
```