Atelier de Professionnalisation 4
Maison des Ligues

Contexte :

Le contexte proposé est celui de la Maison de Ligues de Lorraine (M2L) qui a pour mission de fournir des espaces et des services aux différentes ligues sportives régionales et à d'autres structures hébergées.

Maison des Ligues

SOMMAIRE :

PRODUIT

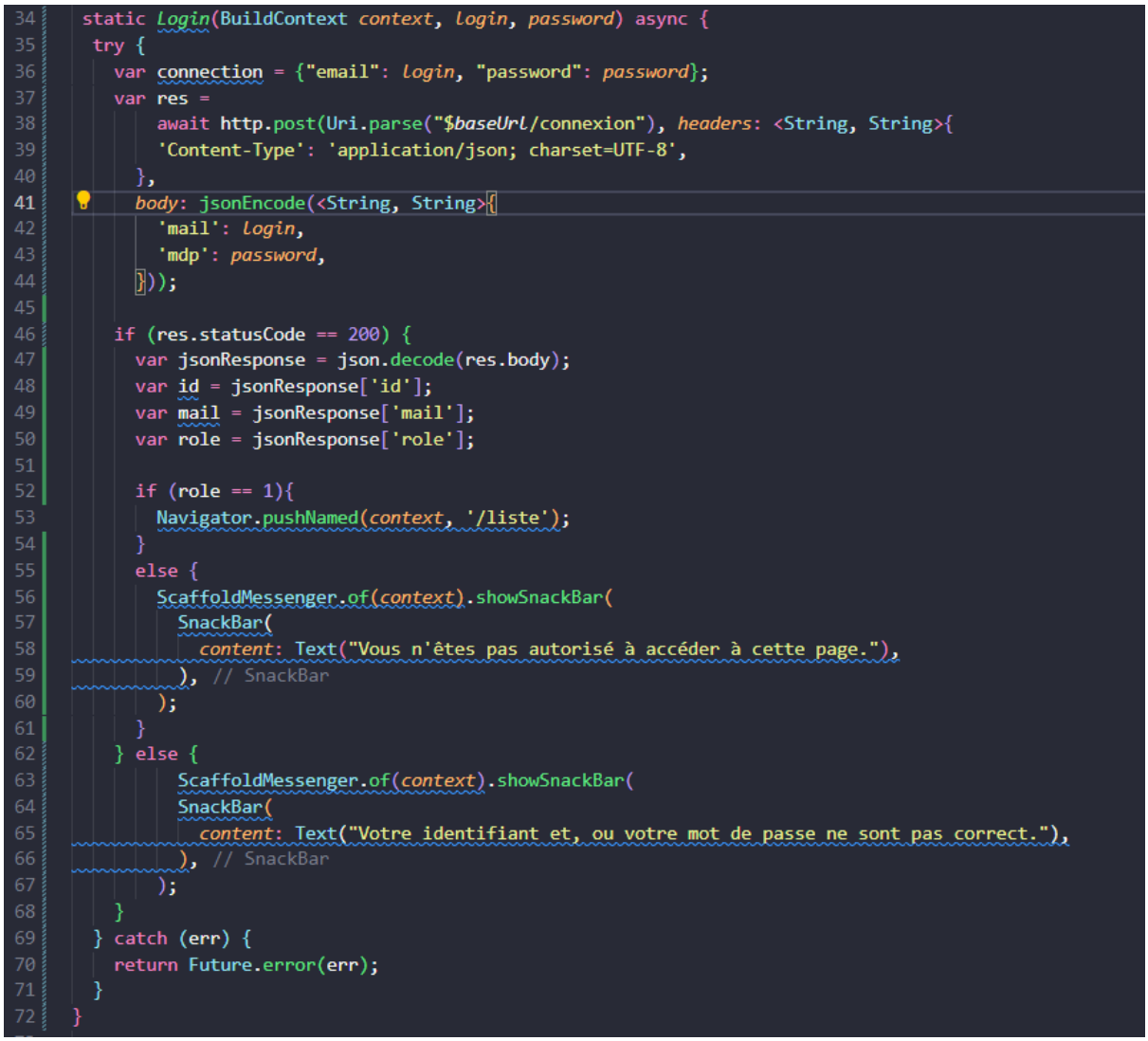
Cette méthode envoie une requête HTTP GET à l'API pour récupérer tous les produits. Elle retourne une future contenant une liste de produits, si la réponse est valide (code HTTP 200), ou une erreur future si la réponse est invalide.

```dart
class Produit {
  static String baseUrl = "http://localhost:8000";

  static Future<List> getAllProduit() async {
    try {
      var res = await http.get(Uri.parse("$baseUrl/produit"));
      if (res.statusCode == 200) {
        return jsonDecode(res.body);
      } else {
        return Future.error("erreur serveur");
      }
    } catch (err) {
      return Future.error(err);
    }
  }
```

CONNEXION

Cette méthode envoie une requête HTTP POST à l'API pour connecter l'utilisateur en utilisant son adresse email et son mot de passe. Si les identifiant et mot de passe sont valide et que l'utilisateur a un rôle égal à 1, alors l'utilisateur est redirigé vers une autre page contenant la liste de nos produits. Sinon, elle affiche un message d'erreur "Vous n'êtes pas autorisé à accéder à cette page."



```
static Login(BuildContext context, login, password) async {
 try {
   var connection = {"email": login, "password": password};
   var res =
       await http.post(Uri.parse("$baseUrl/connexion"), headers: <String, String>{
       'Content-Type': 'application/json; charset=UTF-8',
     },
     body: jsonEncode(<String, String>{
       'mail': login,
       'mdp': password,
     }));

   if (res.statusCode == 200) {
     var jsonResponse = json.decode(res.body);
     var id = jsonResponse['id'];
     var mail = jsonResponse['mail'];
     var role = jsonResponse['role'];

     if (role == 1){
       Navigator.pushNamed(context, '/liste');
     }
     else {
       ScaffoldMessenger.of(context).showSnackBar(
         SnackBar(
           content: Text("Vous n'êtes pas autorisé à accéder à cette page."),
         ), // SnackBar
       );
     }
   } else {
         ScaffoldMessenger.of(context).showSnackBar(
         SnackBar(
           content: Text("Votre identifiant et, ou votre mot de passe ne sont pas correct."),
         ), // SnackBar
       );
   }
 } catch (err) {
   return Future.error(err);
 }
}
```

AJOUTER

Cette méthode envoie une requête HTTP POST à l'API pour ajouter un nouveau produit en utilisant les paramètres d'entrée. Une fois l'ajout réussi, l'utilisateur est redirigé vers notre page d'accueil.

```dart
static ajout(BuildContext context, String Articles, String Image,
    int Prix, int Quantite) async {
  try {
    var res = await http.post(
      Uri.parse("$baseUrl/Ajt"),
      headers: <String, String>{
        'Content-Type': 'application/json; charset=UTF-8',
      },
      body: jsonEncode(<String, String>{
        'Articles': Articles,
        'Image': Image,
        'Prix': Prix.toString(),
        'Quantite': Quantite.toString()
      }),
    );
    if (res.statusCode == 200) {
      Navigator.pushNamed(context, '/liste');
    } else {
      Navigator.pushNamed(context, '/');
    }
  } catch (error) {
    return Future.error(error);
  }
}
```

MODIFIER

Cette méthode envoie une requête HTTP PUT à l'API pour mettre à jour un produit existant identifié par son ID en utilisant les paramètres d'entrée. Si la modification est validée, l'utilisateur est renvoyé dans notre page contenant nos produits.

```dart
static Update(BuildContext context, int id,String Articles, String Image,
    int Prix, int Quantite) async {
  try {
    var res = await http.put(
      Uri.parse("$baseUrl/question/$id"),
      headers: <String, String>{···
      body: jsonEncode(<String, String>{
        'Articles': Articles,
        'Image': Image,
        'Prix': Prix.toString(),
        'Quantite': Quantite.toString(),
        'id': id.toString()
      }),
    );
    if (res.statusCode == 200) {
      Navigator.pushNamed(context, '/liste');
    } else {
      Navigator.pushNamed(context, '/');
    }
  } catch (error) {
    return Future.error(error);
  }
}
```

SUPPRIMER

Cette méthode envoie une requête HTTP DELETE à l'API pour supprimer un produit existant identifié par son ID.

```
126  static Delete(BuildContext context, int id) async {
127    var res = await http.delete(Uri.parse('$baseUrl/Del/$id'),
128        body: id.toString());
129    if (res.statusCode == 200) {
130      Navigator.pushNamed(context, '/liste');
131    } else {
132      Navigator.pushNamed(context, '/');
133    }
134  }
```