MobileApp/lib/services/account_status.dart

```dart
 1  import 'package:flutter/material.dart';
 2  import 'package:project_9/bio/bio_reg_welcome.dart';
 3  import 'package:project_9/bio/bio_reg_login.dart';
 4  import 'package:project_9/services/api.dart';
 5
 6  class AccountStatus extends StatefulWidget {
 7    final String email;
 8    const AccountStatus({required this.email});
 9
10    @override
11    _AccountStatusState createState() => _AccountStatusState();
12  }
13
14  class _AccountStatusState extends State<AccountStatus> {
15    late Future<bool> _userBioFuture;
16
17    @override
18    void initState() {
19      super.initState();
20      _userBioFuture = checkUserBio(widget.email);
21    }
22
23    Future<bool> checkUserBio(String email) async {
24      try {
25        Object userBio = await API.fetchUserBio(email);
26        if (userBio == true) {
27          return true;
28        } else {
29          return false;
30        }
31      } catch (e) {
32        return false;
33      }
34    }
35
36    @override
37    Widget build(BuildContext context) {
38      return FutureBuilder<bool>(
39        future: _userBioFuture,
40        builder: (context, snapshot) {
41          if (snapshot.connectionState == ConnectionState.waiting) {
42            return Center(child: CircularProgressIndicator());
43          } else if (snapshot.hasError) {
44            return Center(child: Text('Erro ao verificar estado da conta'));
45          } else if (snapshot.hasData) {
46            bool registo = snapshot.data ?? false;
47            WidgetsBinding.instance.addPostFrameCallback((_) {
48              if (registo) {
49                Navigator.pushReplacement(
50                  context,
51                  MaterialPageRoute(builder: (context) => BioLoginCreatePage(email:
    widget.email)),
```

```
52                  );
53              } else {
54                Navigator.pushReplacement(
55                  context,
56                  MaterialPageRoute(builder: (context) => BioRegWelcomePage(email:
   widget.email)),
57                );
58              }
59          });
60          return Container(); // Retorna um container vazio enquanto espera o
   redirecionamento
61        } else {
62          return Center(child: Text('Erro inesperado'));
63        }
64      },
65    );
66  }
67 }
```