

## src\features\auth\service.ts

```
1 import {
2   sendEmailDto,
3   sendSMSDto,
4 } from './dto';
5 import { serviceResponseType } from '../../utilities/response';
6
7 import mailService from '../../services/mailService';
8 import axios from 'axios';
9 export default class AuthService {
10
11   static async sendSMS(phone: string, otpCode: string) {
12     try {
13       console.log(
14         process.env.TEXTNG_SMS_API_KEY,
15         process.env.TEXTNG_SENDER_ID,
16         process.env.TEXTNG_route,
17       );
18       const message = `Your OTP is ${otpCode}`.replace(/ /g, '_');
19       const apiUrl = `https://api.textng.xyz/sendsms/?key=${process.env.TEXTNG_SMS_API_KEY}&
sender=${process.env.TEXTNG_SENDER_ID}&phone=${phone}&message=${message}&route=
${process.env.TEXTNG_route}`;
20       const response = await axios.get(apiUrl, {
21         headers: {
22           'User-Agent': 'TEXTNG',
23         },
24       });
25
26       console.log(response.data, 'response.data');
27       // check if response.data has success in the string, if not throw error
28       if (!response.data?.toLowerCase().includes('success')) {
29         throw new Error(response.data);
30       }
31       //console status
32       console.log(response.status, 'response.status');
33       return response.data;
34     } catch (error) {
35       console.error(error, 'error');
36       throw error;
37     }
38   }
39
40   static async sendGeneralSMS(payload: sendSMSDto) {
41     try {
42       const message = payload.message.replace(/ /g, '_');
43       const apiUrl = `https://api.textng.xyz/sendsms/?key=${
44         process.env.TEXTNG_SMS_API_KEY
45       }&sender=${"FDR"}&phone=${payload.phone}&message=${
46         message
47       }&route=${process.env.TEXTNG_route}`;
48       const response = await axios.get(apiUrl, {
49         headers: {
50           'User-Agent': 'TEXTNG',
51         },
```

```

52     });
53
54     console.log(response.data, 'response.data');
55     // check if response.data has success in the string, if not throw error
56     if (!response.data?.toLowerCase().includes('success')) {
57         throw new Error(response.data);
58     }
59     //console status
60     console.log(response.status, 'response.status');
61     return response.data;
62 } catch (error) {
63     console.error(error, 'error');
64     throw error;
65 }
66 }
67
68 static async sendGeneralEmail(payload: sendEmailDto): Promise<serviceResponseType> {
69     try {
70         const mail = await mailService(
71             payload.subject,
72             payload.email,
73             // welcome message with email and password
74             payload.message,
75         );
76         return {
77             success: true,
78             message: 'Email sent successfully',
79             data: mail,
80         };
81     } catch (error) {
82         return {
83             success: false,
84             message: error.message,
85             data: error,
86         };
87     }
88 }
89 }
90
91 }
92 }
93

```