

Acessando a api do GitHub.

- No terminal instale o axios, digitando **yarn add axios**
- Crie a pasta services dentro da src.
- Crie o arquivo src/services/api.js conforme print a seguir.

```
1  import axios from 'axios'
2
3  const api = axios.create({
4    |  baseUrl: 'https://api.github.com',
5  })
6
7  export default api;
```

- Na linha 1 importa o axios
- Na linha 3 cria uma constante chamada api para armazenar o endereço da api do github definido na linha 4.
- Na linha 7 exporta a constante api.

Edite o arquivo src/pages/Main/index.js conforme print a seguir:

- Na linha 3 importa a propriedade Keyboard do react-native.
- Na linha 4 importa a api que contém o endereço do github.
- Nas linhas 8 até 11 cria o estado com os atributos newUser (será utilizado para digitado o login a ser pesquisado github) e Users como um array para armazenar todas as contas do github.
- Na linha 13 cria a função handAddUser como assíncrona.
- Na linha 14 desestrutura os atributos newUser e Users do estado (this.state)
- Na linha 16 realiza uma pesquisa na api do github em /users conforme o login digitado e o resultado armazena na variável response.
- Na linha 18 imprime todos os atributos da api pública do github (response.data)
- Nas linhas 20 até 25 cria um objeto data com os atributos (nome, login do usuário, biografia e imagem (avatar)) que serão exibidos na tela do smartphone

```
1  import React, { Component } from 'react'
2  import Icon from 'react-native-vector-icons/MaterialIcons'
3  import { Keyboard } from 'react-native'
4  import api from '../services/api'
5  import { Container, Form, Input, SubmitButton } from './styles'
6
7  export default class Main extends Component {
8    state = {
9      newUser: '',
10     users: [],
11   }
12
13   handleAddUser = async () => {
14     const { users, newUser } = this.state;
15
16     const response = await api.get(`/users/${newUser}`)
17
18     console.log(response.data)
19
20     const data = {
21       name: response.data.name,
22       login: response.data.login,
23       bio: response.data.bio,
24       avata: response.data.avatar_url,
25     }
```

- Na linha 29 altera o estado dos atributos users e newUser.
- Na linha 30 copia todos os usuários que estão no array e adiciona um novo usuário (objeto data).
- Na linha 31 apaga o conteúdo do atributo newUser.
- Na linha 34 oculta o teclado depois de clicar no botão adicionar (+)

```
26
27   console.log(data)
28
29   this.setState({
30     users: [...users, data],
31     newUser: '',
32   })
33
34   Keyboard.dismiss(); // desabilita o teclado depois que clicar no
35   | | | | | | | | | | // botão de adicionar +
36 }
37
38 render() {
39
40   const { users, newUser } = this.state;
41
```

- Na linha 46 a propriedade `autoCorrect=false` desabilita a correção automática do texto.
- Na linha 47 a propriedade `autoCapitalize=none` desabilita a inicial de cada palavra em maiúscula.
- Na linha 48 define o texto do input como Adicionar usuário.
- Na linha 49 define a propriedade `value` com o conteúdo do usuário digitado (`newUser`) quando o estado for alterado.
- Na linha 50 define a propriedade `onChangeText` para receber o texto digitado no input e automaticamente alterar o estado da propriedade `newUser`
- Na linha 51 substitui a tecla enter pelo texto `send` no ios e uma setinha para o android.
- Na linha 52 define que se pressionar a tecla `send` ou setinha no teclado automaticamente será executado o método `handleAddUser` para pesquisar o login do usuário na api do github.
- Na linha 54 define que se o botão adicionar (+) for clicado a função `handleAddUser` será executado para pesquisar o login na api do github.

```
42     return (  
43       <Container>  
44         <Form>  
45           <Input  
46             autoCorrect={false}  
47             autoCapitalize='none'  
48             placeholder='Adicionar usuário'  
49             value={newUser}  
50             onChangeText={text => this.setState({ newUser: text })}  
51             returnKeyType='send'  
52             onSubmitEditing={this.handleAddUser}  
53           />  
54           <SubmitButton onPress={this.handleAddUser}>  
55             <Icon name='add' size={20} color='#FFF' />  
56           </SubmitButton>  
57         </Form>  
58       </Container>  
59     )  
60   }  
61 }  
62  
63 Main.navigationOptions = {  
64   title: 'Usuários'  
65 }
```