

Usando axios antes de refatorarmos o código

```
entrar = () => {
  axios
    .post('http://localhost:8080/api/usuarios/autenticar', {
      email: this.state.email,
      senha: this.state.senha
    }).then(response => {
      localStorage.setItem('_usuarioLogado', JSON.stringify(response.data));
      this.props.history.push('/home');
    }).catch(erro => {
      this.setState({ mensagemErro: erro.response.data });
    });
}

axios.get(`http://localhost:8080/api/usuarios/${usuarioLogado.id}/saldo`)
  .then(response => {
    this.setState({ saldo: response.data });
  }).catch(error => {
    console.error(error.response);
  });
```

```
1 const url = "https://api.github.com/users/nodejs"
2 fetch(url)
3   .then(res => res.json())
4   .then(resp => console.log(resp))
```

Linha 3 - a fetch retorna uma promessa e então transformamos esse resposta em json
Linha 4 - com a promessa já resolvida, podemos olhar a resposta que recebemos da nossa requisição

```
//É o mesmo exemplo acima, mas aqui usamos async e await
async function exemplo() {
  const response = await fetch("https://api.github.com/users/nodejs");
  const data = await response.json();
  console.log(data.id);
}
exemplo();
```

```
const url = "https://api.github.com/users/nodejs"
fetch(url, {
  method: 'POST',
  body: JSON.stringify(body)
})
  .then(res => res.json())
  .then(response => console.log(response))
```