



React/React Native Http Interceptor.

Date: 2023-11-13

By: csr13

Download as PDF

When developing user based mobile application or web applications and using JWTokens for authentication, there is the need to always return the user when the backend returns a not authenticated response -- wether is to do a token refresh or a session expiration action e.g: session token expires user needs to log in again.

This is a simple http interceptor that can be used when developing Mobile Applications and Web Applications that handle user actions e.g: social media.

AuthService has some helper methods, self explanatory.

```
import axios from "axios";
import AuthService from "../auth.service";

const redirect = (url, clear = false) => {
  if (clear) {
    window.localStorage.clear();
  }
  window.location.href = url;
};

const token_expired = () => {
  var user = AuthService.decode_user();
  if (!user.exp) {
    redirect("/login", false);
  }
  if (new Date(user.exp * 1000) < new Date()) {
    return true;
  }
  return false;
};

const Interceptor = axios.interceptors.response.use(
  (response) => {
    if (token_expired()) {
      redirect("/login", true);
    }

    if (response.status === 401) {
      redirect("/login", true);
    } else if (response.status === 403) {
      redirect("/login", true);
    }
    return response;
  },
  (error) => {
    if (error.response && error.response.status === 401) {
      redirect("/login", true);
    } else if (error.response && error.response.status === 403) {
      redirect("/login", true);
    }
    return error;
  }
);

export default Interceptor;
```

Related Notes

[1\) Deploying production React app.](#)

Date published: 2023-11-10

[devops react docker](#)

[2\) React/React Native Http Interceptor.](#)

Date published: 2023-11-13

[react http best-practice](#)

© csr13 2023

