```
import { createApi, fetchBaseQuery } from '@reduxjs/toolkit/query/react'

// Define a service using a base URL and expected endpoints
export const userAuthApi = createApi({
  reducerPath: 'userAuthApi',
  baseQuery: fetchBaseQuery({ baseUrl: 'http://127.0.0.1:8000/api/user/' }),
  endpoints: (builder) => ({
    registerUser: builder.mutation({
      query: (user) => {
        return {
          url: 'register/',
          method: 'POST',
          body: user,
          headers: {
            'Content-type': 'application/json',
          }
        }
      }
    }),
    loginUser: builder.mutation({
      query: (user) => {
        return {
          url: 'login/',
          method: 'POST',
          body: user,
          headers: {
            'Content-type': 'application/json',
          }
        }
      }
    }),
    getLoggedUser: builder.query({
      query: (access_token) => {
        return {
          url: 'profile/',
          method: 'GET',
          headers: {
            'authorization': `Bearer ${access_token}`,
          }
        }
      }
    }),
    changeUserPassword: builder.mutation({
      query: ({ actualData, access_token }) => {
        return {
          url: 'changepassword/',
          method: 'POST',
          body: actualData,
          headers: {
            'authorization': `Bearer ${access_token}`,
          }
        }
      }
    }),
    sendPasswordResetEmail: builder.mutation({
      query: (user) => {
        return {
          url: 'send-reset-password-email/',
          method: 'POST',
          body: user,
          headers: {
            'Content-type': 'application/json',
          }
        }
      }
    }),
    resetPassword: builder.mutation({
      query: ({ actualData, id, token }) => {
        return {
          url: `/reset-password/${id}/${token}/`,
          method: 'POST',
          body: actualData,
          headers: {
            'Content-type': 'application/json',
          }
        }
      }
    }),
  }),
})

export const { useRegisterUserMutation, useLoginUserMutation,
useGetLoggedUserQuery, useChangeUserPasswordMutation,
useSendPasswordResetEmailMutation, useResetPasswordMutation } = userAuthApi
```