<https://github.com/axios/axios>

* ComponentDidMount http requestler yapmak icin en iyi yerdir.

**AXIOS KULLANIMI**

  componentDidMount() {

    axios.get("https://jsonplaceholder.typicode.com/posts").then(response => {

      this.setState({ posts: response.data });

    });

  }

  render() {

    const posts = this.state.posts.map((post, index) => (

      <Post key={index} title={post.title} />

    ));

    return (

      <div>

        <section className="Posts">{posts}</section>

        …

      </div> )}}

**SADECE 4 TANE POST GOSTERMEK**

  componentDidMount() {

    axios.get("https://jsonplaceholder.typicode.com/posts").then(response => {

//4 POSTU SECIP CEKMEK

      const posts = response.data.slice(0, 4);

//YENI OBJELERDEN OLUSAN DIZIYI OLUSTURMAK

      const updatedPosts = posts.map((post, index) => ({

        ...post,

        key: index,

        author: "Max"

      }));

      this.setState({ posts: updatedPosts });

    });

  }

**BIRDEN FAZLA ISTEGIN ONUNE GECMEK -** Fetching Data on Update  
(without Creating Infinite Loops)

  state = {

    loadedPost: null

  };

  componentDidUpdate() {

    if (this.props.id) {

      if (

        !this.state.loadedPost ||

        (this.state.loadedPost && this.state.loadedPost.id !== this.props.id)

      ) {

        axios

          .get("https://jsonplaceholder.typicode.com/posts/" + this.props.id)

          .then(response => {

            this.setState({ loadedPost: response.data });

          });

      }

    }

  }

**AXIOS POST ISLEMI**

  state = {

    title: "",

    content: "",

    author: "Max"

  };

  postDataHandler = () => {

      const data = {

          title: this.state.title,

          body: this.state.content,

          author: this.state.author

      }

      axios.post('https://jsonplaceholder.typicode.com/posts', data)

        .then(response => {

            console.log(response.data);

        })

        .catch(err => {

            console.error(err);

        });

  };

**AXIOS DELETE ISLEMI**

  deletePostHandler = () => {

    axios

      .delete("https://jsonplaceholder.typicode.com/posts/" + this.props.id)

      .then(response => {

        console.log(response);

      })

      .catch(err => {

        console.error(err);

      });

  };

BASE URL AYARLAMAK

axios.defaults.baseURL = "https://jsonplaceholder.typicode.com";

Yukaridaki gibi baseURL ayarladiktan sonra isteklerimizi asagidaki gibi kisa sekillerde atabiliriz.

  deletePostHandler = () => {

    axios

      .delete("/posts/" + this.props.id)

      …

  };

DEFAULT Ayarlari Ayarlamak

axios.defaults.headers.common["Authorization"] = "AUTH TOKEN";

axios.defaults.headers.post["Contect-Type"] = "application/json";

yada kendi instancelerimizi olusturarak bunlari import ederek de kullanabiliriz.

**AXIOS INSTANCE OLUSTURMAK**

İndex.js dizininde axios.js dosyasini olusturup icini ayarliyoruz. Ornek asagidaki gibi

import axios from 'axios';

const instance = axios.create({

    baseURL = "https://jsonplaceholder.typicode.com";

});

instance.defaults.headers.common["Authorization"] = "AUTH TOKEN";

instance.defaults.headers.post["Contect-Type"] = "application/json";

export default instance;

daha sonra asil axios yerine bunu import ederek de kullanabiliyoruz.

**INTERCEPTORS EKLEME**

axios.interceptors.request.use(

  request => {

    console.log(request);

    return request;

  },

  error => {

    console.log(error);

    return Promise.reject(error);

  }

);

axios.interceptors.response.use(

  response => {

    console.log(response);

    return response;

  },

  error => {

    console.log(error);

    return Promise.reject(error);

  }

);