Lab 05

Daten einer API abfragen

- BITTE EIGENEN API-KEY NUTZEN -

1. API Key erstellen auf https://api.nasa.gov
2. NasaFetch.jsx:

```
import React, { useState, useEffect } from
'react';

export function NasaFetch() {
    const [curiosityData, setCuriosityData] =
useState();
    useEffect(() => {
      const fetchData = async () => {
        const response = await fetch(

"https://api.nasa.gov/planetary/apod?api_key=UC
HWX8OxC9eVQyp3nT4J4OvfVSOl6aI5o9ifzsg2",
          {
            method: "GET",
            headers: {
              "Content-Type":
"application/json",
            },
  } );
        setCuriosityData(await
response.json());
      };
      fetchData();
    }, []);
    return (
      <div>
        <h1>Curiosity Fetcher</h1>
        <div>{JSON.stringify(curiosityData,
null, 2)}</div>
        {console.log(curiosityData)}
        <img src={curiosityData.url}></img>
      </div>
  ); }
```

3. Wir können auch einen eigenen useFetch-Hook bauen:

```
import React, { useState, useEffect } from
'react';

export const useFetch = (url, config) => {
    const [response, setResponse] = useState();
    useEffect(() => {
      const fetchData = async () => {
        const data = await fetch(url, {
          method: "GET",
          headers: {
            "Content-Type": "application/json",
          },
  });
        setResponse(await data.json());
      };
      fetchData();
    }, [url]);
    return response;
  };

  export function NasaUseFetch() {
    const curiosityData = useFetch(
      "https://api.nasa.gov/mars-
photos/api/v1/rovers/curiosity?api_key=pfbmfMce
G1yKkZqq9KouM871p78HssyISz7ef84W"
  );
    return (
      <div>
        <h1>Curiosity Fetcher</h1>
        <div>{JSON.stringify(curiosityData,
null, 2)}</div>
      </div>
  );
    };
```

4. Als nächstes nutzen wir Axios:
`npm install axios`

```
import React, { useState, useEffect } from
'react';
import axios from "axios";

const useFetch = (url, config) => {
  const [response, setResponse] = useState();
  useEffect(() => {
    const fetchData = async () => {
      const { data } = await axios.get(url, {
        headers: {
          "Content-Type": "application/json",
        },
});
      setResponse(await data);
    };
    fetchData();
  }, [url]);
  return response;
};

export function NasaUseAxios() {
  const curiosityData = useFetch(
    "https://api.nasa.gov/mars-
photos/api/v1/rovers/curiosity?api_key=pfbmfMce
G1yKkZqq9KouM871p78HssyISz7ef84W"
);
  return (
    <div>
      <h1>Curiosity Fetcher</h1>
      <div>{JSON.stringify(curiosityData, null,
2)}</div>
    </div>
);
  };
```

5. Als letztes die gleiche Abfrage mit React Query:
`npm install react-query`

```
import { QueryClient, QueryClientProvider, useQuery }
from 'react-query'

 const queryClient = new QueryClient()

 export default function ReactQueryNasa() {

   return (
     <QueryClientProvider client={queryClient}>
       <Example />
     </QueryClientProvider>
   )
 }

 export function Example() {

   const { isLoading, error, data } =
useQuery('repoData', () =>
     fetch('https://api.nasa.gov/mars-
photos/api/v1/rovers/curiosity?api_key=pfbmfMceG1yKkZ
qq9KouM871p78HssyISz7ef84W').then(res =>
       res.json()
     )
   )

   if (isLoading) return 'Loading...'
   if (error) return 'An error has occurred: ' +
error.message

   return (

     <div>
      {console.log(data)}
       <h1>{data.rover.name}</h1>
       {JSON.stringify(data)}
     </div>
   )
 }
```