Desafio 2 – Data list.

1 - **Using Async/Await Syntax**: I revised the data fetching method to use async/await inside the fetchData function. This change (lines 9-14) makes the asynchronous operations more readable and easier to follow compared to the use of chained .then() statements. It also brings the code up to date with contemporary JavaScript practices.

2- **Error Handling with try...catch**: The addition of a try...catch block (lines 10-14) enhances the error handling mechanism. This approach provides a clear structure for handling errors, ensuring that any issues encountered during data retrieval are logged appropriately and can be addressed systematically.

3- **Organizing with useEffect**: I placed the data fetching logic within a dedicated function, fetchData (lines 8-14), inside the useEffect hook. This encapsulation not only organizes the code better but also prepares it for the inclusion of more complex logic or additional effects in the future, if needed.

4- **Simplified Conditional Rendering for Loading**: The conditional rendering for the loading state (lines 16-18) has been kept straightforward and efficient, clearly communicating to users that data is being loaded. This ensures a good user experience by providing feedback during the data loading process.

References

<https://swizec.com/categories/react%20query/>

<https://medium.com/tag/react-query>

<https://tkdodo.eu/blog/practical-react-query>