

Exercise (Instructions): Fetch Handling Errors

Objectives and Outcomes

In this exercise you will learn how to handle errors encountered while communicating with the server. At the end of this exercise you will be able to:

- Configure your app to appropriately handle errors encountered while communicating with the server

Handling Errors

- Open `ActionCreators.js` and update it as follows:

```
1  ...
2
3  export const fetchDishes = () => (dispatch) => {
4
5      dispatch(dishesLoading(true));
6
7      return fetch(baseUrl + 'dishes')
8        .then(response => {
9          if (response.ok) {
10             return response;
11          } else {
12             var error = new Error('Error ' + response.status + ': ' + response
13               .statusText);
14             error.response = response;
15             throw error;
16          }
17        },
18        error => {
19          var errmess = new Error(error.message);
20          throw errmess;
21        })
22        .then(response => response.json())
23        .then(dishes => dispatch(addDishes(dishes)))
24        .catch(error => dispatch(dishesFailed(error.message)));
25    }
26  ...
27
28  export const fetchComments = () => (dispatch) => {
29    return fetch(baseUrl + 'comments')
30      .then(response => {
31        if (response.ok) {
32          return response;
33        } else {
34          var error = new Error('Error ' + response.status + ': ' + response
35            .statusText);
36          error.response = response;
37          throw error;
38        }
39      },
40      error => {
41        var errmess = new Error(error.message);
42        throw errmess;
43      })
44      .then(response => response.json())
45      .then(comments => dispatch(addComments(comments)))
46      .catch(error => dispatch(commentsFailed(error.message)));
47    };
48  ...
49
50  export const fetchPromos = () => (dispatch) => {
51
52      dispatch(promosLoading());
53
54      return fetch(baseUrl + 'promotions')
55        .then(response => {
56          if (response.ok) {
57             return response;
58          } else {
59             var error = new Error('Error ' + response.status + ': ' + response
60               .statusText);
61             error.response = response;
62             throw error;
63          }
64        },
65        error => {
66          var errmess = new Error(error.message);
67          throw errmess;
68        })
69        .then(response => response.json())
70        .then(promos => dispatch(addPromos(promos)))
71        .catch(error => dispatch(promosFailed(error.message)));
72    }
73  ...
```

- Save all the changes and do a Git commit with the message "Fetch Handling Errors".

Conclusions

In this exercise you learnt how to configure your app to handle errors in communicating with the server.

