

11 Async/await a ošetření chyb

Ošetření chybových stavů API a použití async/await místo then()

◀ Předchozí
Opakování: LeviExpress, druhá část

Nemá následující
lekci ▶

Chybové stavy API

Dosud jsme při komunikaci s backendem přes API spoléhali na to, že vždy všechno dopadne dobře a server nám odpoví tak, jak očekáváme. Realita je ale jiná – občas server nemá svůj den a místo očekávané odpovědi vrátí chybu. Někdy nemáte úplně stabilní připojení k internetu nebo naopak nemáte světlý den vy a chyba bude v tom, co na server posíláte. Server tak místo kýžené odpovědi pošle něco ve smyslu „tomu nerozumím“.

Pokud nastane chyba v komunikaci se serverem, váš kód někde skončí chybou, do konzole prohlížeče se vypíše chyba, ale uživatel se na stránce nedozví, že došlo k nějakému problému. Pojdme to nyní spravit a upravit aplikaci tak, aby přežila (alespoň některé) chyby při komunikaci s backendem.

Chyby v komunikaci můžeme podle závažnosti rozdělit na dvě kategorie:

1. Server odpoví, ale vrátí chybu.
2. Vůbec se nepodaří se serverem spojit – například proto, že jste zadali špatnou adresu serveru, a nebo proto, že vůbec nefunguje připojení k internetu.

Začneme první variantou. Vytvoříme si jednoduchou komponentu s jedním tlačítkem, na které když uživatel klikne, zavolá se endpoint, který vrací náhodná čísla od 1 do 6 (simuluje hod kostkou). Používat budeme API na adrese `https://random.kodim.app`.

Hlavní komponenta aplikace může vypadat třeba takto – komunikace se serverem je schválně vytažená do funkce `fetchData`, protože teď až do konce lekce budeme měnit už jen tělo této funkce:

```
const App = () => {
  const [roll, setRoll] = useState();
  const [errorMessage, setErrorMessage] = useState(null);

  const fetchData = () => {
    fetch('https://random.kodim.app/api/diceroll')
      .then((resp) => resp.json())
      .then((data) => setRoll(data.result.number));
  };

  const handleClick = () => {
    fetchData();
  };

  return (
    <div className="container">
      <div>
        <button onClick={handleClick}>Budiž číslo: </button>
      </div>
      {errorMessage === null ? (
        <div>Náhodné číslo: {roll}</div>
      ) : (
        <div>{errorMessage}</div>
      )}
    </div>
  );
};
```

Následující

Stavové kódy HTTP odpovědi



Verze 2.0.0-beta.7