

Promise

4. Promise Chaining

Promise Chaining

- Promise Chaining allows you to manipulate the content of the promise



Promise.prototype.then

- then accepts 2 functions, one that will be called on resolve (**onfulfilled**), and the second will be called on reject (**onrejected**)
- then - returns a Promise
- The content of the Promise the then returns depends on the value returned from **onfulfilled**, **onrejected**

```
then<TResult1 = T, TResult2 = never>(
  onfulfilled?: ((value: T) => TResult1 | PromiseLike<TResult1>) | undefined | null,
  onrejected?: ((reason: any) => TResult2 | PromiseLike<TResult2>) | undefined | null
): Promise<TResult1 | TResult2>;
```

Promise chaining - example

- With then manipulating the promise, we can turn a promise containing a certain value to a promise containing another value.
- This manipulation allows us to do **Promise Chaining**

```
helloPromise
  .then(
    url => fetch(url)
  )
  .then(
    res => res.json()
  )
  .catch(() => [])
  .then((dataFromServer) => {
    ...
  })
```

Thank You

Next Lesson: Error handling best practices