academeez

Promise

2. Creating our first Promise

Shouter - Listeners

academeez

- We can divide the promise to 2 parts
  - Shouter - will wrap the async code and emit a single shout when the async code executes
  - Listeners - Will attach a callback that will be called when the shout is emitted

Shout

academeez

- The shout will wrap our async code, and send a shout when the async code is executes
- For example a promise that shout “hello world” after 1 sec
- It is also common that we will not create ourselves the promise, rather use a method which will return a promise.

```
const helloPromise = new Promise((resolve) => {
    setTimeout(() => {
        resolve('Hello Listeners');
    }, 1000);
});
```

resolve / reject

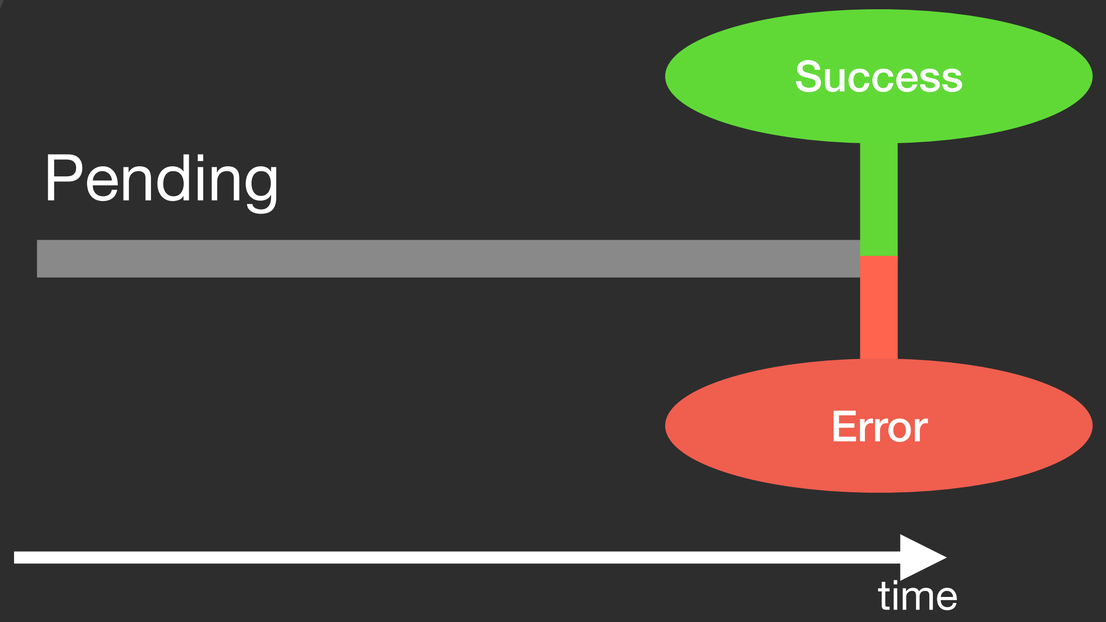
academeez

- We need to pass the promise constructor a function, that function will be called with two callbacks: resolve, reject
- The resolve is used to say: our async code ran successfully and pass data to the listeners
- The reject is used to say: our async code fail, here is the error
- You can call either resolve or reject and only once

```
const helloPromise = new Promise((resolve, reject) => {
    setTimeout(() => {
        // reject(new Error('something happened'));
        resolve('Hello Listeners');
    }, 1000);
});
```

Promise state

- If we can call the **resolve / reject** once this means our promise can be in one of the following states:



Listener

- After we have the promise, we can now attach listeners that will be called when the shout is emitted
- To attach a listener we use the **then** method on the promise and pass a callback as the first argument that will be called when the promise emits a shout
- You can attach multiple listeners the same way

```
helloPromise.then(
    (message) => console.log(message)
);
```

academeez

Thank You

Next Lesson: Dealing with errors on Promises