academeez

Promise

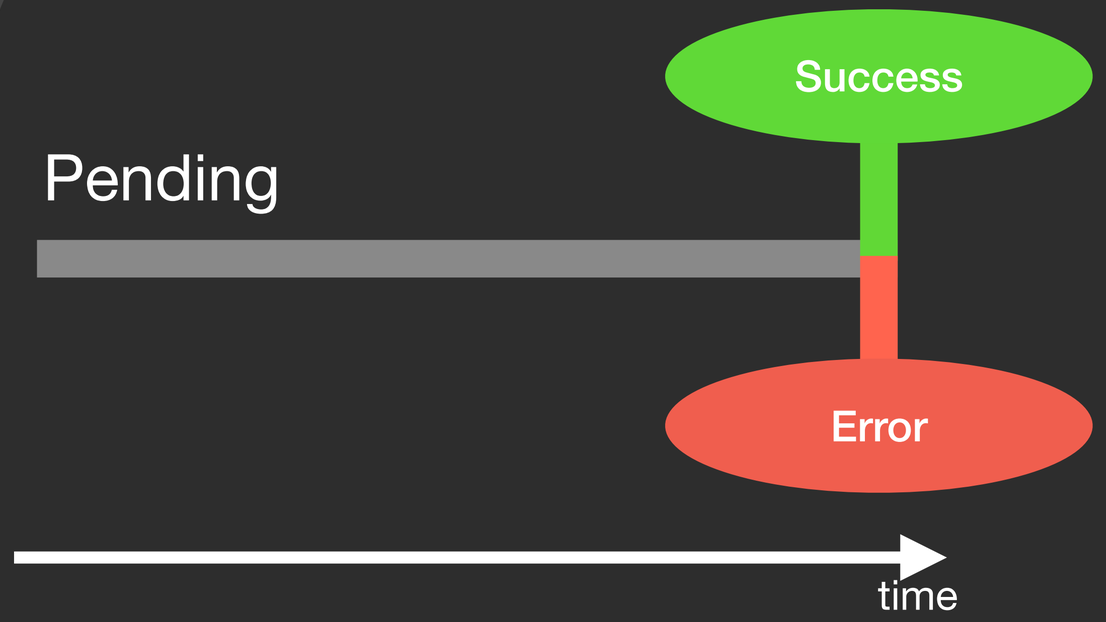
3. Rejecting Promise

Promise Reject

- An async action, at times, can fail
  - For example query a server when you don’t have an internet connection
- Promise wraps an async code which can fail
- A Promise can reject and send the listeners an Error

Promise state

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



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 an Error

- The shout will wrap our async code, and send a shout when the async code is executes
- For example a promise that shout an Error after 1 sec
- The reject is used to say: our async code fail, here is the error

```
const timerPromise = new Promise((resolve, reject) => {
  setTimeout(() => {
    reject(new Error("Something went wrong"));
  }, 1000);
});
```

Listen for error

- A listener can also listen for an error, by passing a second callback to the **then**
- A listener can also use the **catch** method on the promise to pass an error callback without the need to add a success callback as well

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

helloPromise.catch(
    (err) => console.log(err.message)
)
```

academeez

Thank You

Next Lesson: Promise Chaining