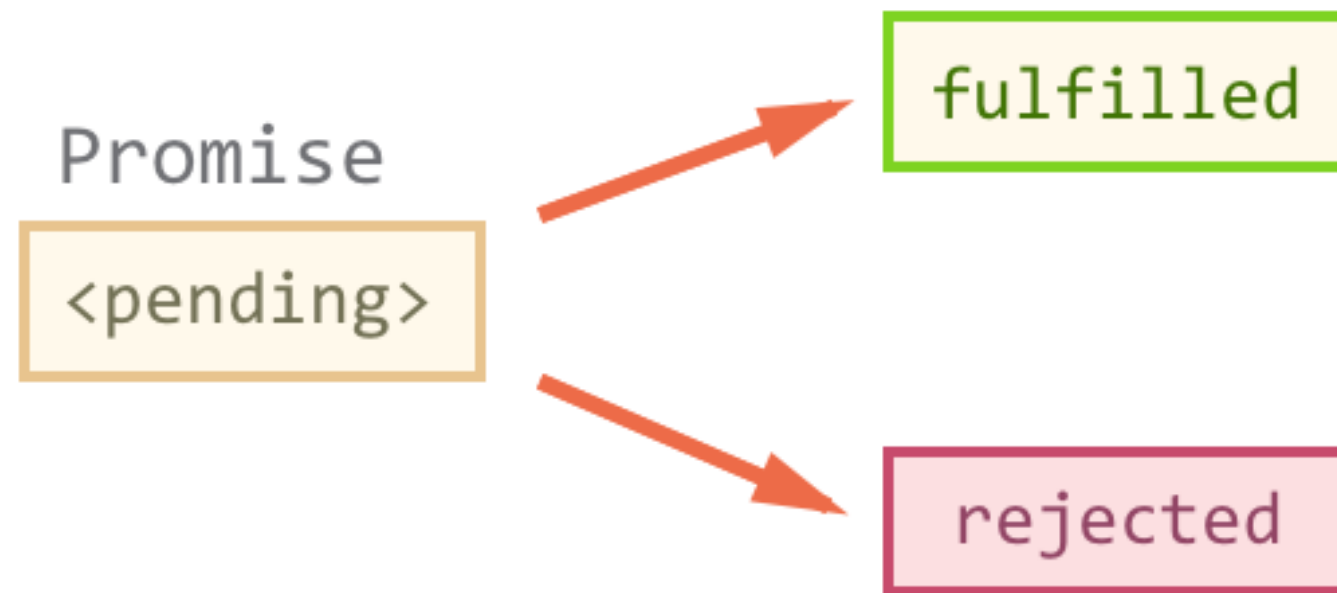


Promise



```
const promise = new Promise((resolve, reject) => {  
  // some async staff  
  setTimeout(() => {  
    const result = 'Anything';  
    if (10 > 2) {  
      resolve(result);  
    }  
    else {  
      const error = 404;  
      reject(error);  
    }  
  }, 5000);  
});
```

```
const handleFirstResult = result => {  
  console.log('first handle');  
  return result;  
}
```

```
const handleSecondResult = result => {  
  console.log('second handle');  
  return result;  
}
```

```
promise  
  .then(handleFirstResult)  
  .then(handleSecondResult)  
  .catch(error => {  
    // do smth  
  })
```



```
const promise = new Promise((resolve, reject) => {  
  // some async staff  
  setTimeout(() => {  
    resolve(10);  
  }, 5000);  
});  
  
let a;  
  
promise.then(res => { a = res; });  
  
console.log(a); // ???
```

Fetch



```
let promise = fetch(url, [options])
```

Без options - это простой GET запрос,
скачивающий данные по заданному url

Options

- method - http метод - 'GET' | 'POST' | 'PATCH' | ...
- headers: { key: value, ... }
- body
 - A. строка
 - B. объект FormData
 - C. Blob
 - D. JSON

Процесс получения ответа обычно происходит в два этапа

- Promise выполняется с объектом класса `Response` в качестве результата, как только придет ответ с сервера
- Получение тела ответа с помощью дополнительного метода класса `Response`

Response

Properties

- ok
- headers
- redirected
- status
- statusText
- ...

Methods

- blob()
- text()
- json()
- ...
-

Redux-thunk

`npm install —save redux-thunk`