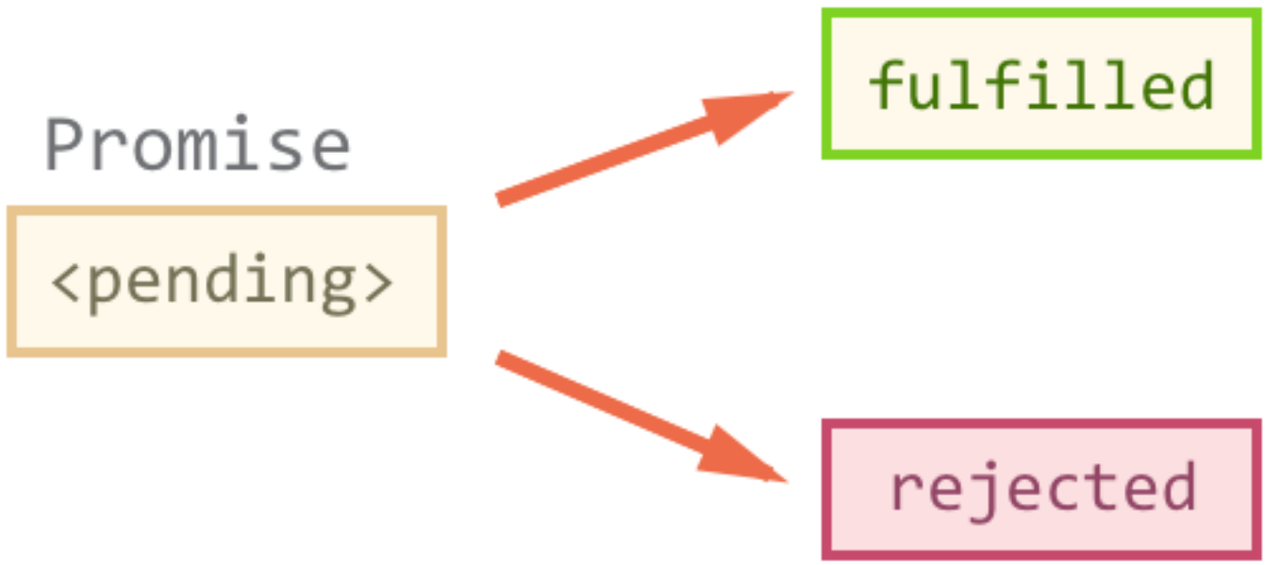
Promise


Promise
<pending>
fulfilled
rejected

```
const promise = new Promise((resolve, reject) => {
  // now in pending state
  if (10 > 2) {
    resolve(1234) // to fulfilled state
  } else {
    reject('Error') // to rejected state
  }
})
```

```
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

```
POST https://request.url.com/api/path/login/
Accept: application/json, text/plain, */*
Content-Type: application/json;charset=utf-8
Origin: https://request.url.com
Cookie: PHPSESSID=UarhdHU%2Fs2LRgF%2FYhht43XEad3SAC;
Content-Length: 60
Accept-Language: ru
Host: request.url.com
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_3)
AppleWebKit/605.1.15 (KHTML, like Gecko) Version/13.0.5 Safari/605.1.15
Referer: https://request.url.com/request/was/from/here/
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
--data-binary {"login":"this_is_login","password":"my_secret_password"}
```

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

Redux-thunk

Это middleware для redux, чтобы была возможность совершать асинхронные действия в action creators