IT-Academy.BG

Асинхронност и заявки в JavaScript

Димитър Митев

Асинхронност в JS

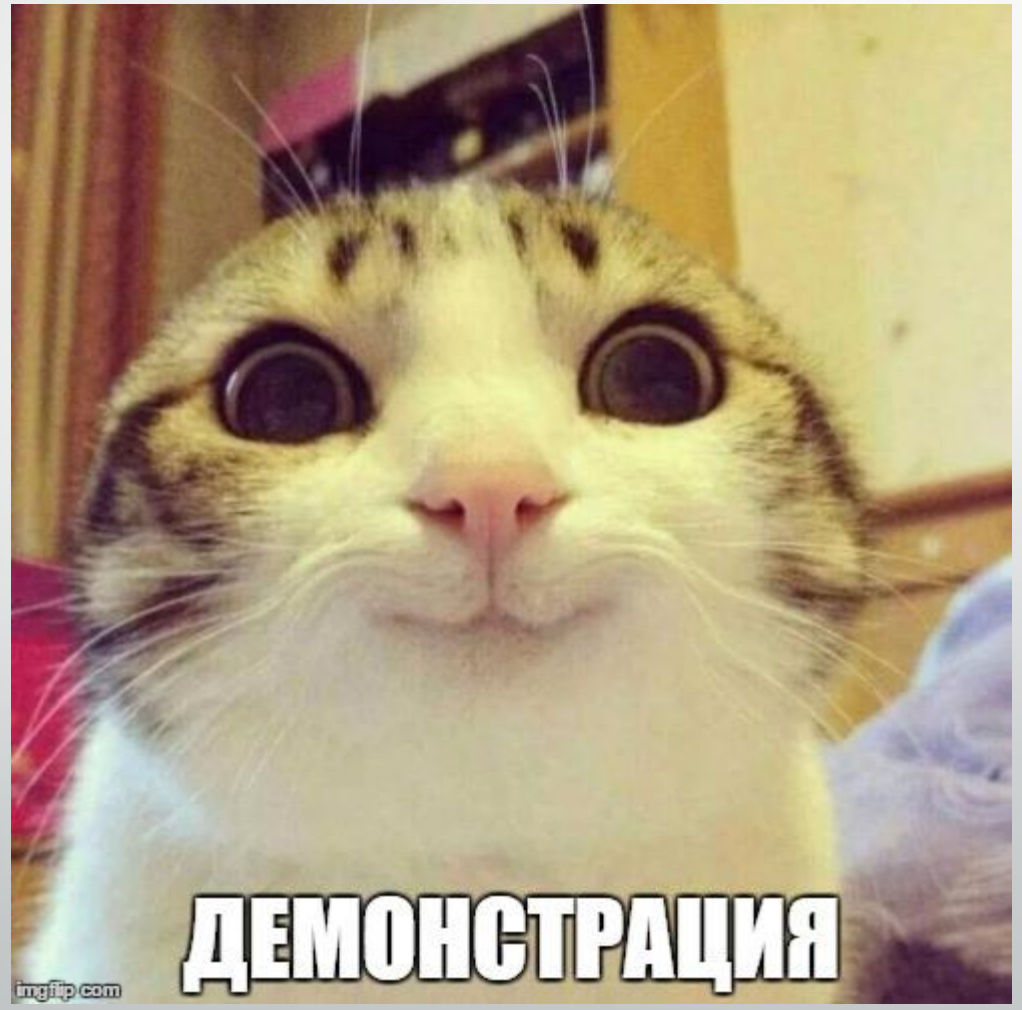
- JS се изпълнява само на 1 процесорна нишка
  - Операции, които отнемат повече време, могат да блокират процеса
- В JS може да се използва асинхронност
  - Дълги операции се разделят на по-къси
  - Отлагаме изпълненението на дадени парчета код
- Използват се т.нар callbacks
  - Функции, които да се изпълнят на даден етап
- Асинхронност в браузърите – AJAX, Geolocation, CSS3 Animations….

Callback-ориентирано програмиране

- Callback функция
  - Функция, която се подава като параметър на др функция
    - Event handler-ите са вид callback
    - *setTimeout()* и *setInterval()* , приемат callback функция като параметър
  - Много характерно за nodejs
  - В уеб се счита са остарял подход и се използват Promise-и

```
setTimeout(function () {
    alert('BOOOM');
}, 3000);
```

Callback фунцкия



Promise – модерният начин в JS

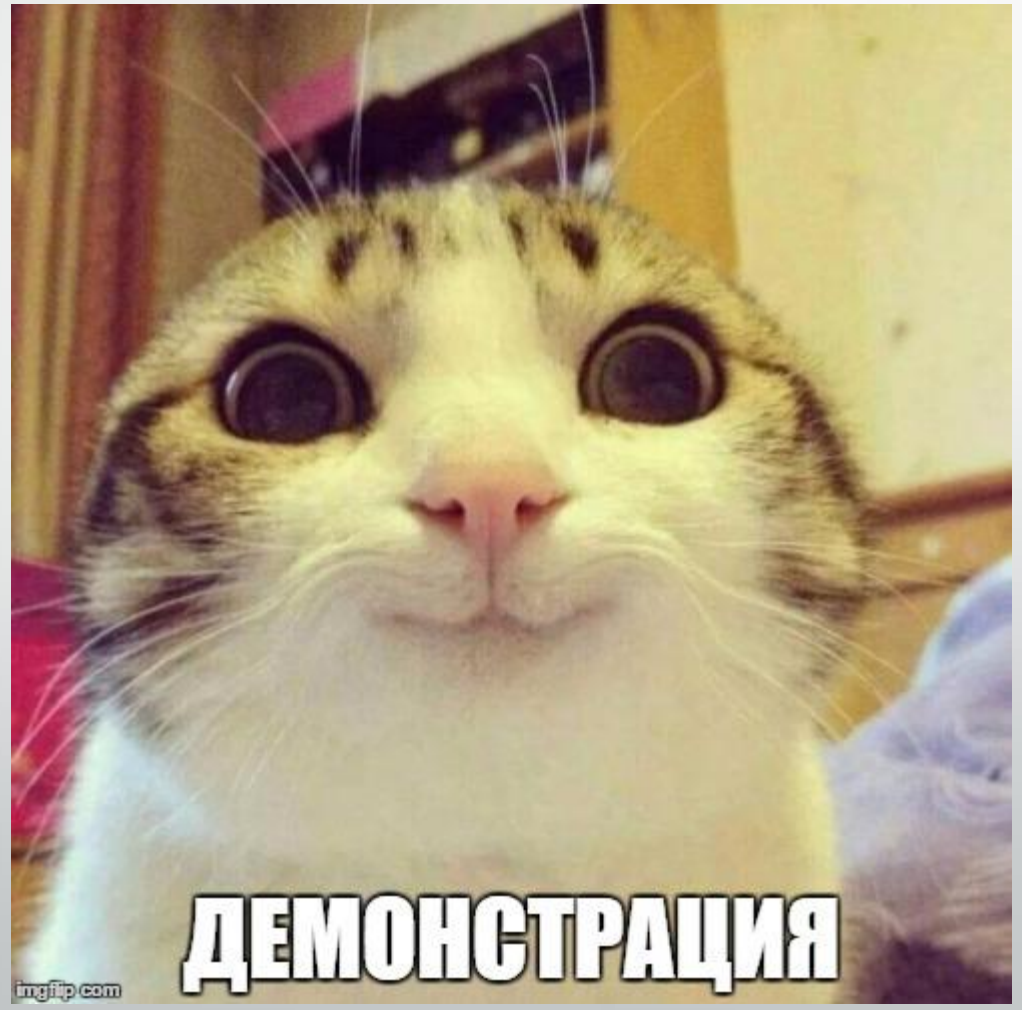
- *Promise* е такъв обект, който представя евентуално бъдещо състояние или стойност
  - Функцията, която връща *Promise*, „обещава“ да върне стойност
- *Promise*-а има състояние:
  - Изпълнен (fulfilled, resolved, succeeded)
  - Отказан (rejected)
  - Чакащ (pending)
- *Promise*-а се използва като обект, на който все пак му знаем стойността

Promise – модерният начин в JS

- *Promise* е шаблон в програмирането
  - От ES2015 съществува вграден Promise обект в браузърите
  - Може да се използва външна библиотека – q, rsvp, jQuery
- Promise-а обикновено има .then() метод, който може да приеме 3 параметъра – success, error и progress

```
promiseMeSomething()
        .then(function (value) {
            //handle success here
        }, function (reason) {
            //handle error here
});
```

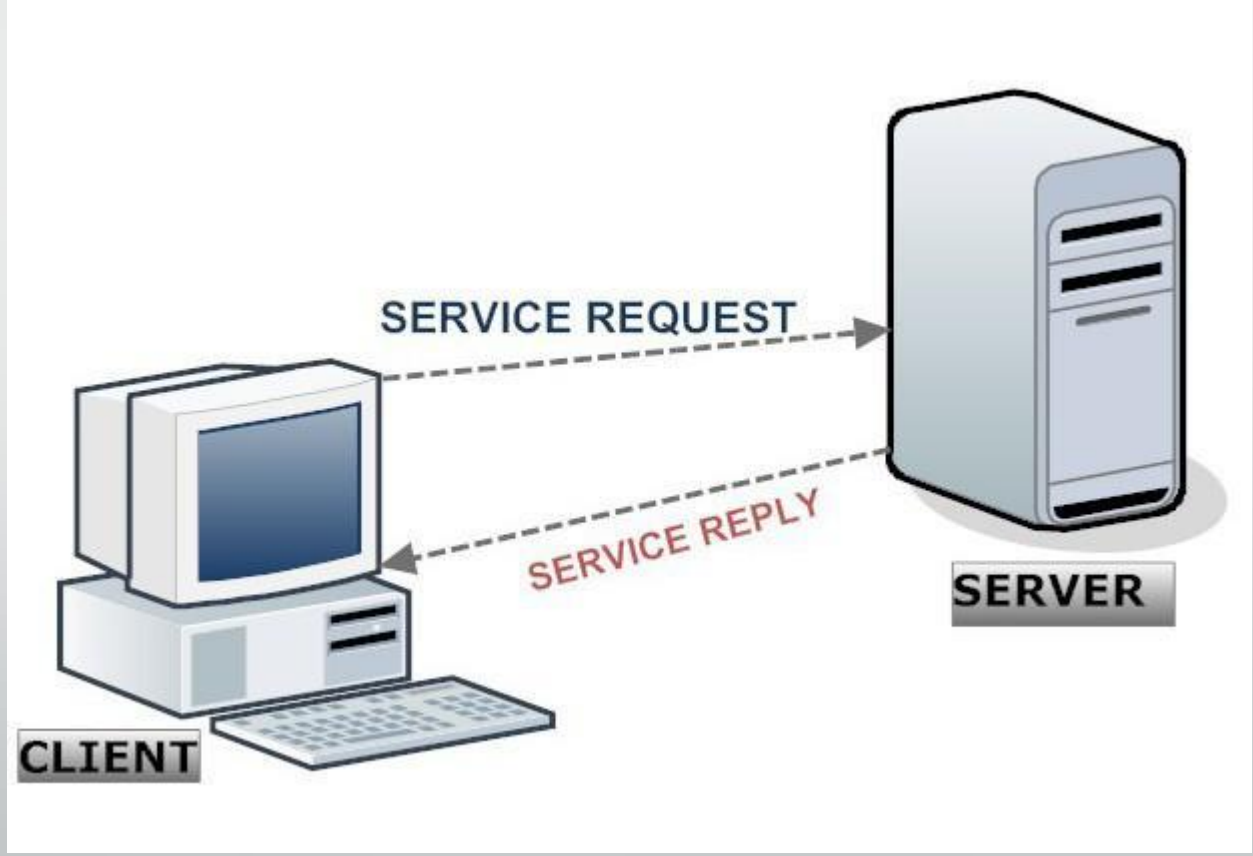
Promise обекта



Какво е HTTP

- Hyper Text Transfer Protocol
  - Клиент-сървър протокол за пренасяне на уеб ресурси (HTML, картинки…)
- Характеристики на HTTP
  - Request-Response модел
  - Текстово базиран формат
  - Разчита на уникални URL-и
  - Позволява метаданни за дадени ресурси (напр. Encoding)
  - **Няма състояние!!!**

Какво е HTTP



HTTP статуси

- 1xx -> информационни
- 2xx -> успех
- 3xx -> пренасочване
- 4xx -> грешки на клиента
- 5xx -> грешки на сървъра
- Повече информация за статусите може да намерите тук

HTTP методи / HTTP verbs/

- GET -> използва се, когато просто искаме да вземем данни от сървъра
- POST -> използва се, когато искаме да изпратим данни към сървъра
- PUT -> използва се, когато искаме да променим данни на сървъра
- DELETE -> използва се, когато изкаме да изтрием данни
- Други методи: HEAD, TRACE, OPTIONS, PATCH, CONNECT
- Повече информация

Заглавия /headers/

- HTTP е в текстово базиран формат
  - Чрез т.нар. заглавия /headers/ може да внесем допълнителна информация
- Често използвани
  - Content-Type
  - Content-Disposition
    - Определят как да бъде върнато съдържанието от сървъра

```
Content-Type: text/html; charset=utf-8
```

```
Content-Type: application/pdf
Content-Disposition: attachment; filename="Report2010.pdf"
```

AJAX

- **A**synchronous **J**avaScript **A**nd **X**ML
  - Техника за асинхронно зареждане на динамично съдържание от сървъра
  - Позволява динамични промени на UI/UX
  - Не се презарежда цялата страница
- Има 2 основни типа AJAX
  - Частично зареждане на HTML фрагменти
  - Получаване и обработка на JSON обекти посредством JavaScripit/jQuery
- AJAX изпозла HTTP
  - Има request/response модел
  - Има заглавия /headers/
  - Трябва да таргетира конкретен URL

Уеб услуги /web services/

- Уеб услугата /web service/ е начин на комуникация м/у 2 устройства в WWW /world wide web/
  - Сървърът показва опередeлени сървиси на конкретен URL
  - Клиента изпраща заявки /requests/ към тези URL-ли
- Уеб услугите са част от т.нар SOA архитектура
  - Сървърът отговаря за бизнес логиката и бизнес обектите
  - Клиентската част отговаря за презентирането на бизнес логиката

XMLHttpRequest обекта

- *XMLHttpRequest* обекта е част от JS, която предоставя възможност да достъпваме ресурси по даден URL
  - Създаден е от Microsoft и се използва от Mozilla, Apple и Google
  - Стандартизиран
  - Може да се използва както синхронно, така и асинхронно
  - Данните могат да бъдат в различни формати – XML, JSON, HTML, plain text
- Използва се само, ако има нужда от него
- Обикновено ползваме jQuery
- DEMOS

jQuery AJAX

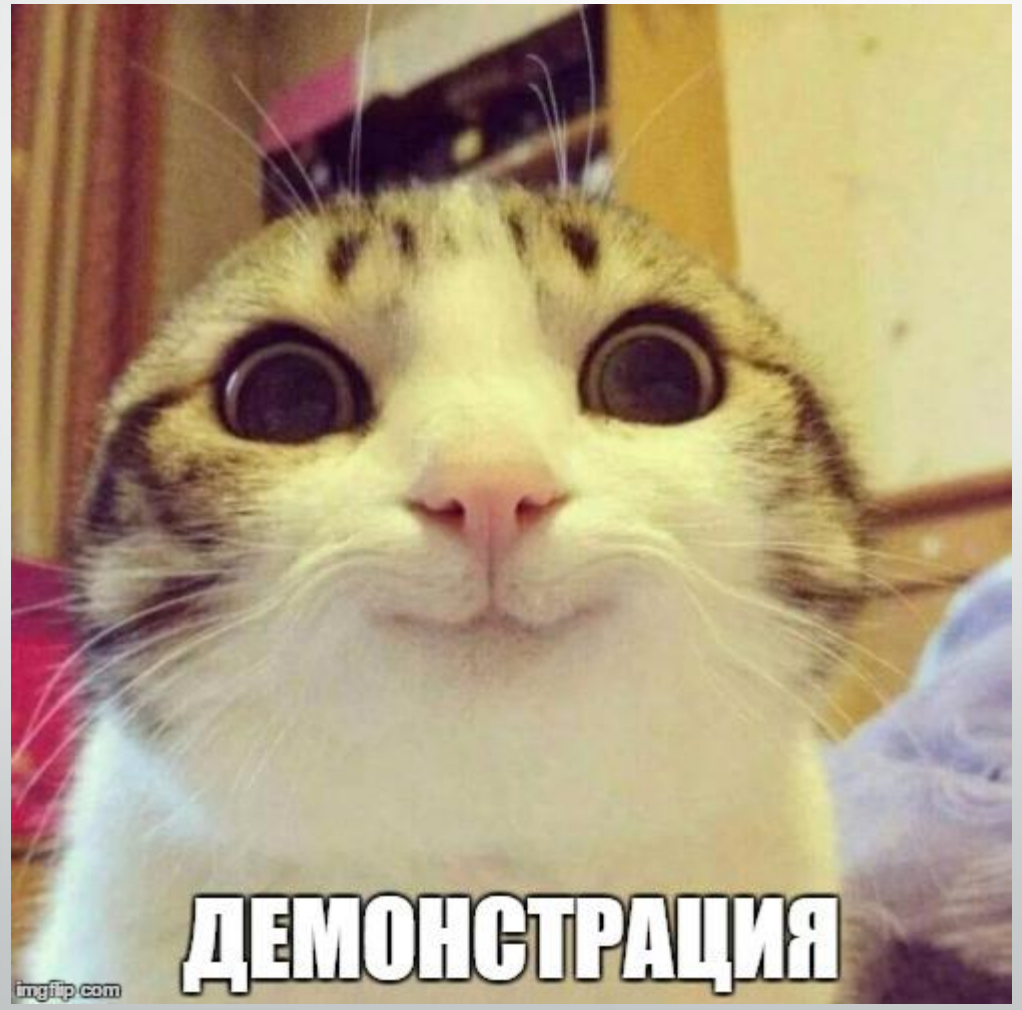
- jQuery има възможност да създава HTTP заявки
- AJAX методи от jQuery:
  - *$.ajax(options)*
  - *$.getJSON(url, success).error()*
  - *$.post(url, data, success, 'json').error()*
  - *$(selector).load(urlToPartialHTML)*

jQuery AJAX

- Основният метод са създаване на HTTP заявки е $.ajax(url [, options])
  - Options параметъра съдържа вс данни, които са необходими, за да създадем HTTP заявка
  - DEMO

```
$.ajax(url, {
    type: 'GET',
    timeout: 5000,
    contentType: 'application/json',
    success: function (response) { /* handle the success */ },
    error: function (err) { /* handle the error */}
});
```

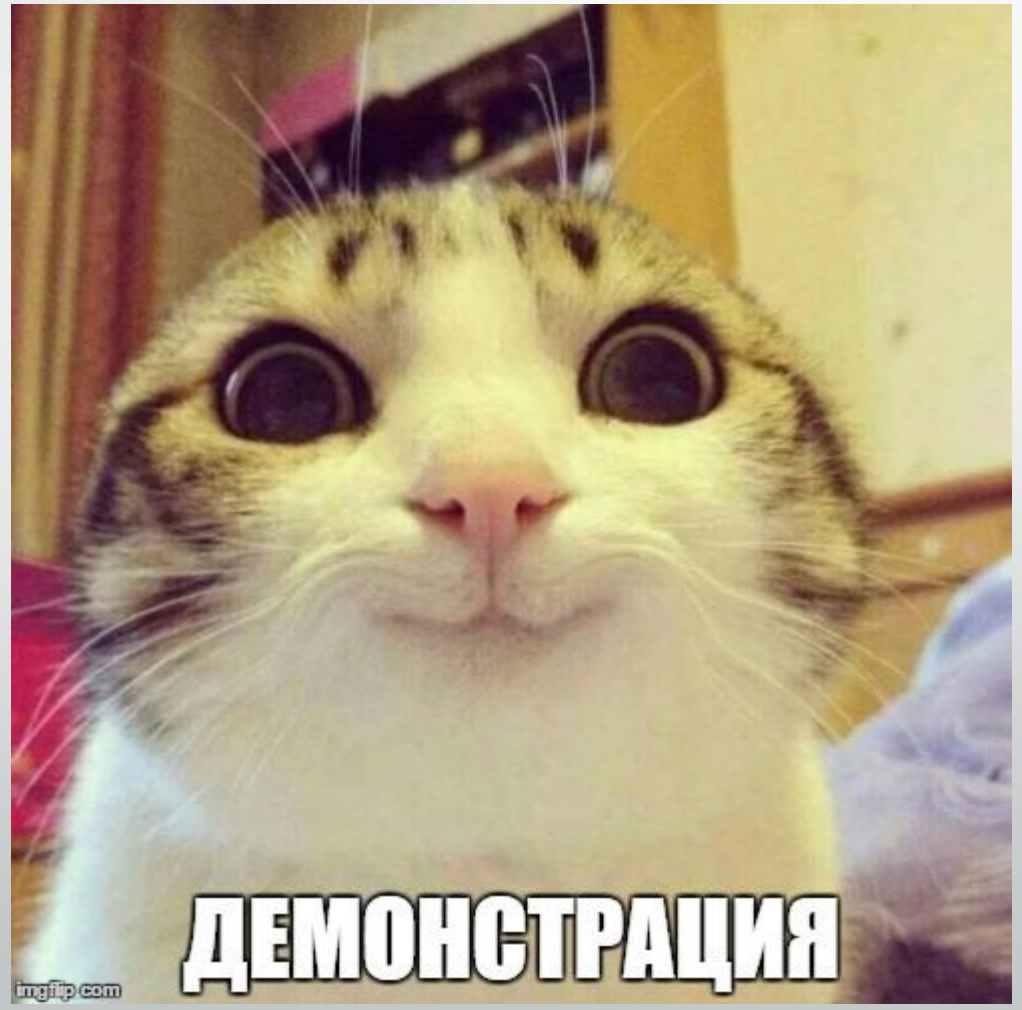
$.ajax()



$.load(url [, data] [, complete])

- Използва се за асинхронно зареждане на HTML-и
- Помага за динамичен UI без да презареждаме цялата страница
- Параметри
  - url -> мястото, където се намира въпросния HTML
  - Data -> обект, който може да бъде изпратен до сървъра
  - Complete -> callback функция, която се изпълнява, след като вече е зареден html

$.load()



IT-Academy.BG

IT-Academy.BG

Домашна работа

1. Напишете JS код, който по даден URL: https://api.themoviedb.org/3/movie/550?api_key=ee4147bedd685cdebb23042532c92117
   - Да направи GET заявка
   - Да визуализира на база получените данни
     - Заглавието на филма
     - Описанието /overview/ на филма
     - Дата на издаване /release date/
     - Линк към страницата на филма /homepage/