

УЕБ-БАЗИРАНО ПРОГРАМИРАНЕ

Павел Кюркчиев

Ас. към ПУ „Паисий Хилендарски“

<https://github.com/pkyurkchiev>

@pkyurkchiev

SINGLE PAGE APPLICATIONS



Какво е Single Page Application(SPA)?

- SPA е веб сайт или веб приложение, което комуникира с потребителя като динамично презаписва съдържанието на страницата вместо да зарежда изцяло нова страница от сървъра.

Предимства на Single Page Application

- По – малък трафик по мрежата;
- Лесна поддръжка;
- Подобрява и потребителското изживяване(user experience).

...

Този подход заобикаля неприятното презареждане между отварянето на различни страници, и така наподобява работата на десктоп приложение.

Подходи на работа

- Подход 1 – зареждане на цялото приложение при първо повикване;
- Подход 2 – зареждане на layout (основата на проекта) при първо повикване и при потребителско действие зареждане само на необходимите ресурси.

Технологии използвани за комуникация при SPA

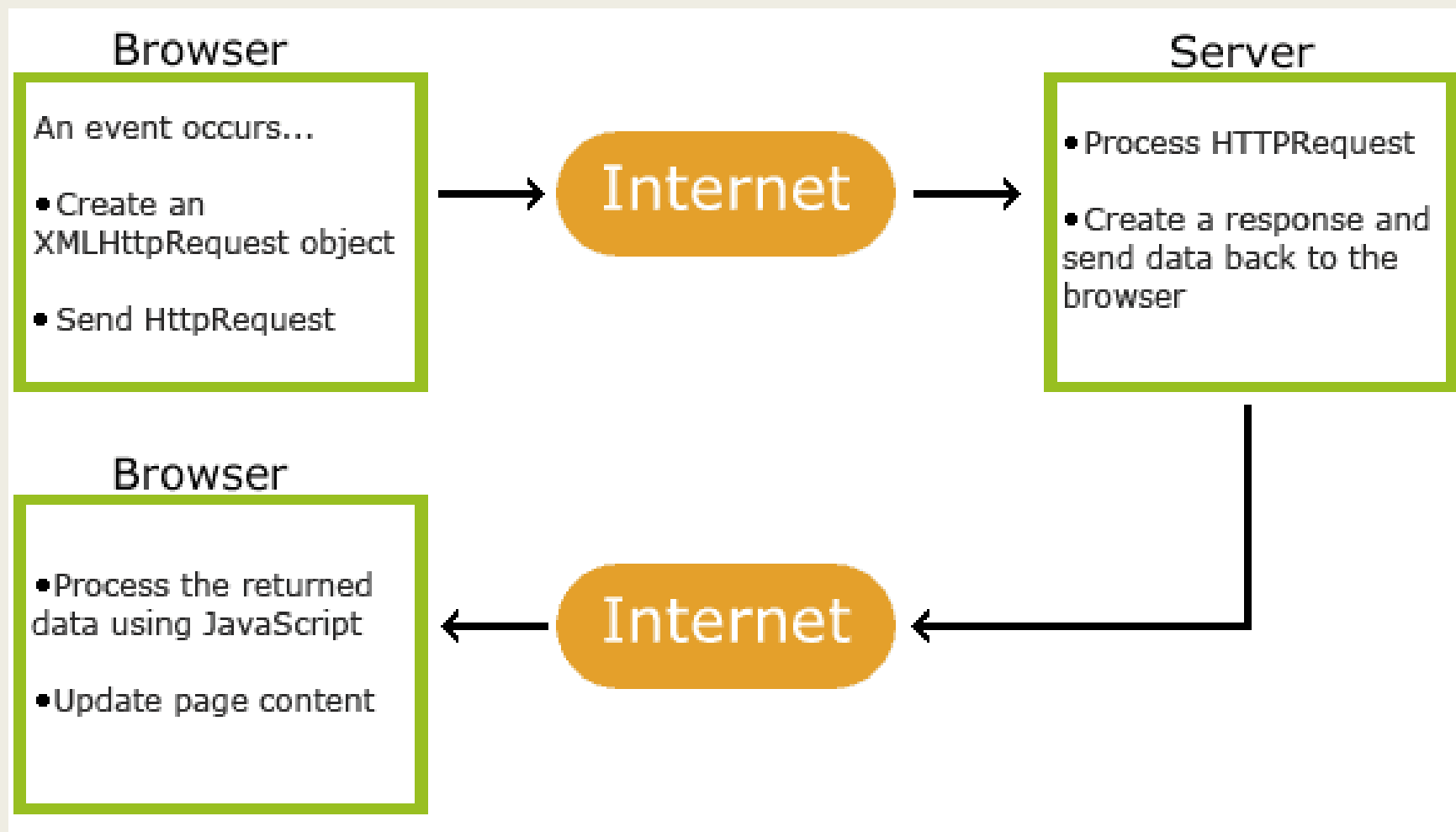
Какво представлява JavaScript frameworks?

- JavaScript Framework е рамка за разработка на приложения, написана на JavaScript. JavaScript Framework се различава от JavaScript библиотеката в своя контрол на потока: Библиотеката предлага функции, които да бъдат извиквани от нейния родителски код, докато рамката определя целия дизайн на приложението.

Какво представлява Ajax?

- AJAX = Asynchronous JavaScript And XML
- AJAX не е език за програмиране. AJAX представлява комбинация:
 - *Вграден в браузъра XMLHttpRequest обект (за заявяване на данни от уеб сървър)*
 - *JavaScript и HTML DOM (за показване или използване на данните)*
- AJAX може да бъде използван не само с XML но и с текст и с JSON.

AJAX диаграма



Какво представлява WebSocket?

- WebSocket е двупосочна комуникационна технология клиент-сървър в реално време, част от спецификацията на HTML5, по-добра от Ajax по отношение на производителност и простота.

SPA с Ajax и Websockets

- WebSockets обикновено се използват в приложения, където се изискват актуализации в реално време. WebSockets ни дава възможност да мултиплексираме данни на една и съща връзка, като по този начин човек лесно може да внедри „AJAX библиотека върху WebSockets“ и все пак да използва същата WebSocket връзка за двупосочно прехвърляне на данни в реално време.

Какво представлява Server-sent events?

- Server-sent events (SSE) е техника, при която сървърите могат да инициират предаване на данни към клиентите на брауъра. След като е установена първоначална връзка, потока от събития остава отворен, докато не бъде затворен от клиента. SSE се изпращат през HTTP и имат разнообразни функции, които в WebSockets липсват по дизайн, като автоматично повторно свързване, идентификатори на събития и възможност за изпращане на произволни събития.

Какво представлява браузър приставки(browser plugins)?

- Въпреки че този метод е остарял, асинхронните запитвания към сървъра могат да бъдат постигнати с помощта на приставки за браузър като Silverlight, Flash или Java applets.

Архитектури в контекста на SPA

Thin server architecture

- SPA премества логиката от сървъра към клиента, като ролята на уеб сървъра се изпълнява от Data API (уеб услуга/ги). Тази архитектурна промяна в някои кръгове се нарича „Thin Server Architecture“, за да подчертае, че сложността е преместена от сървъра към клиента, с аргумента, че това в крайна сметка намалява общата сложност на системата.

Thick stateful server architecture

- Сървърът запазва необходимото състояние в паметта на клиентската страница. По този начин, когато всяка заявка извика сървъра (действия на потребителя), сървърът изпраща съответния HTML и / или JavaScript с конкретните промени, за да приведе клиента в новото желано състояние (обикновено добавяне / изтриване / актуализиране на част от клиентския DOM).

В същото време състоянието в сървъра се актуализира. По-голямата част от логиката се изпълнява на сървъра, а HTML се рендва и на сървъра. Сървърът симулира уеб браузър, приемайки събития и извършва промени в състоянието, тези промени автоматично се разпространяват към клиента.

- Този подход се нуждае от повече сървърна памет и обработки на сървър, но предимството е опростения модел за разработка:
 - *приложението е напълно кодирано в сървъра*
 - *данните и състоянието на потребителския интерфейс в сървъра се споделят в едно и също пространство на паметта.*

Thick stateless server architecture

- Това е подобно на stateful сървърния подход. Клиентската страница изпраща данни, представящи текущото ѝ състояние до сървъра, чрез заявки с Ajax. Използвайки тези данни, сървърът може да реконструира състоянието на частта от страницата (намираща се на клиента), която трябва да бъде променена и генерира необходимите данни или код (JSON или JavaScript),

които се връщат към клиента, за да доведе до ново състояние. Обикновено се променя DOM-ма на страницата според действието на клиента, който инициира заявката.

- Този подход се нуждае от много повече ресурси за да може да възстанови всички данни. В същото време този подход по-лесно се скалира, тъй като в сървъра не се съхраняват данни за страницата на клиента.

Предизвикателства пред SPA модела

- Search engine optimization
 - *Заради слабата поддръжка на JavaScript от някои Web search engines, SEO оптимизацията представлява голям проблем за страници използващи SPA модела.*

- Клиент/Сървър код споделяне(Client/Server code partitioning)
 - За да се увеличи количеството код споделян между сървъра и клиента се използват *log-less* темплейт езици като *Mustache* или *Handlebars*. Възпроизвеждането от шаблони може да има отрицателни ефекти върху ефективността, когато актуализирате само малка част от страницата - като стойността на въвеждане на текст в голям шаблон.
- История на брауъра (Browser history)
 - По дефиниция SPA (единична страница) прави бутоните за навигация *Forward/Back* на брауъра напълно неизползваеми.

■ Анализи (Analytics)

- *Инструменти като Google Analytics разчитат в голяма степен на цели нови страници, заредени в браузъра, да предизвикат зареждане на нова страница. SPA не работят по този начин.*

■ Добавяне на страница за зареждане към SPA (Adding page loads to an SPA)

- *Възможно е добавяне на страница за зареждане към SPA посредством HTML5 API, това ще помогне за добавянето на Анализи. Но няма да реши изцяло проблема, трудността произтича от правилното разпределяне на събития.*

- Скорост на първоначалното зареждане (Speed of initial load)
 - *Първоначалното зареждане на SPA е значително по бавно от, сървър базираните приложения. Това е така защото при първоначалното зареждане на SPA освен информацията на страниците се зареждат както рамката за разработка така и допълнително кодове за обработка.*
- Скорост зареждането на страници (Speeding up the page load)
 - *Скоростта на зареждане на страниците е значително по – бавна заради допълнителното обръщение към Web API за извличане на информацията за страниците.*

Жизнен цикъл на страницата (Page lifecycle)

- SPA се зарежда напълно при първоначалното зареждане на страниците и след това регионите на страниците се заменят или актуализират с нови фрагменти на страници, заредени от сървъра при поискване. За да се избегне прекомерното изтегляне на неизползвани функции, SPA често ще изтегля постепенно повече функции, когато станат необходими, или малки фрагменти от страницата, или завършени екранни модули. SPA наподобява Single Document Interface (SDI).

ВЪПРОСИ ?

