



IT Education Academy

WWW.ITEA.UA



www.itea.ua

Егор Маркевич

- ◆ Инструктор **IT Education Academy**
- ◆ Frontend developer **Matrixian Group**



AJAX

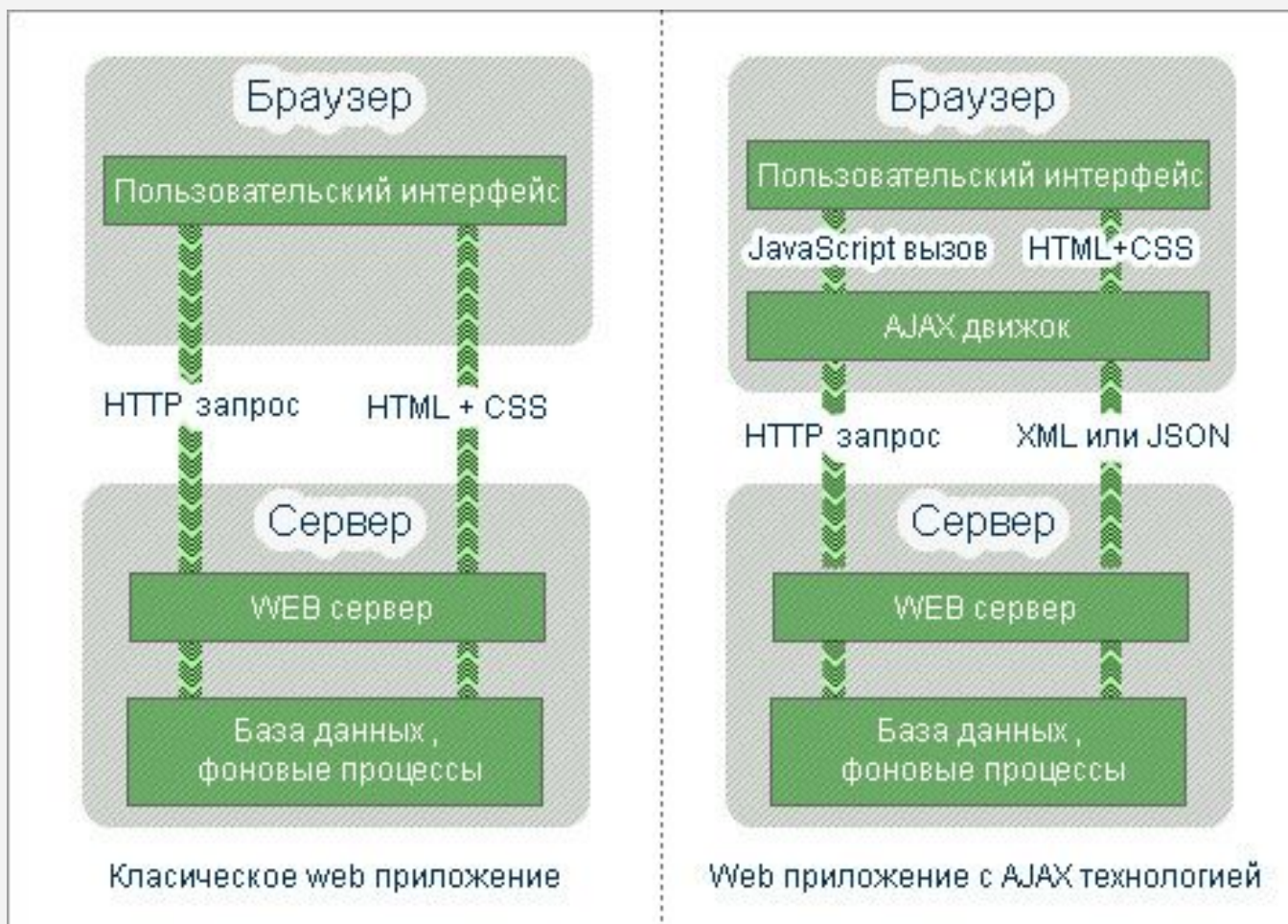
AJAX - Asynchronous JavaScript and XML



Заголовок слайда

AJAX, Ajax (от англ. Asynchronous Javascript and XML — «асинхронный JavaScript и XML») — подход к построению интерактивных пользовательских интерфейсов веб-приложений, заключающийся в «фоновом» обмене данными браузера с веб-сервером. В результате, при обновлении данных веб-страница не перезагружается полностью, и веб-приложения становятся быстрее и удобнее.

По-английски AJAX произносится как «эйджэкс», по-русски довольно распространено «аякс».



HTTP Методы

OPTIONS используется для определения возможностей веб-сервера или параметров соединения для конкретного ресурса. В ответ серверу следует включить заголовок Allow со списком поддерживаемых методов. Также в заголовке ответа может включаться информация о поддерживаемых расширениях

GET&POST

HEAD аналогичен методу **GET**, за исключением того, что в ответе сервера отсутствует тело

PUT применяется для загрузки содержимого запроса на указанный в запросе URI. Если по заданному URI не существовало ресурса, то сервер создаёт его и возвращает статус 201 (Created).

PATCH аналогично PUT, но применяется только к фрагменту ресурса.

DELETE удаляет указанный ресурс.

TRACE возвращает полученный запрос так, что клиент может увидеть, какую информацию промежуточные серверы добавляют или изменяют в запросе.

HTTP статусы

1xx Informational («Информационный»)

В этот класс выделены коды, информирующие о процессе передачи. В HTTP/1.0 сообщения с такими кодами должны игнорироваться. В HTTP/1.1 клиент должен быть готов принять этот класс сообщений как обычный ответ, но ничего отправлять серверу не нужно. Сами сообщения от сервера содержат только стартовую строку ответа и, если требуется, несколько специфичных для ответа полей заголовка. Прокси-серверы подобные сообщения должны отправлять дальше от сервера к клиенту.

HTTP статусы

2xx Success («Успех»)

Сообщения данного класса информируют о случаях успешного принятия и обработки запроса клиента. В зависимости от статуса сервер может ещё передать заголовки и тело сообщения.

3xx Redirection («Перенаправление»)

Коды класса 3xx сообщают клиенту что для успешного выполнения операции необходимо сделать другой запрос (как правило по другому URI). Из данного класса пять кодов 301, 302, 303, 305 и 307 относятся непосредственно к перенаправлениям (редирект). Адрес, по которому клиенту следует произвести запрос, сервер указывает в заголовке Location. При этом допускается использование фрагментов в целевом URI.

HTTP статусы

4xx Client Error («Ошибка клиента»)

Класс кодов 4xx предназначен для указания ошибок со стороны клиента. При использовании всех методов, кроме HEAD, сервер должен вернуть в теле сообщения гипертекстовое пояснение для пользователя.

5xx Server Error («Ошибка сервера»)

Коды 5xx выделены под случаи неудачного выполнения операции по вине сервера. Для всех ситуаций, кроме использования метода HEAD, сервер должен включать в тело сообщения объяснение, которое клиент отобразит пользователю.

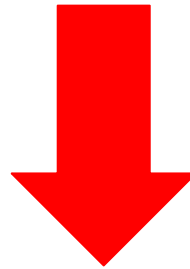
JSONP

JSONP или «JSON with padding» (JSON с набивкой) — это дополнение к базовому формату JSON. Он предоставляет способ запросить данные с сервера, находящегося в другом домене — операцию, запрещённую в типичных веб-браузерах из-за [политики ограничения домена](#).

Запросы для JSONP получают не JSON, а произвольный JavaScript-код. Они обрабатываются интерпретатором JavaScript, а не парсером JSON. Существуют серьёзные риски, связанные с безопасностью при использовании JSONP, в большинстве ситуаций использование CORS является лучшим выбором.

JSONP

```
<script type="text/javascript" src="http://server2.example.com/Users/1234?jsonp=parseResponse">  
</script>
```



```
parseResponse({"Name": "Foo", "Id": 1234, "Rank": 7});
```

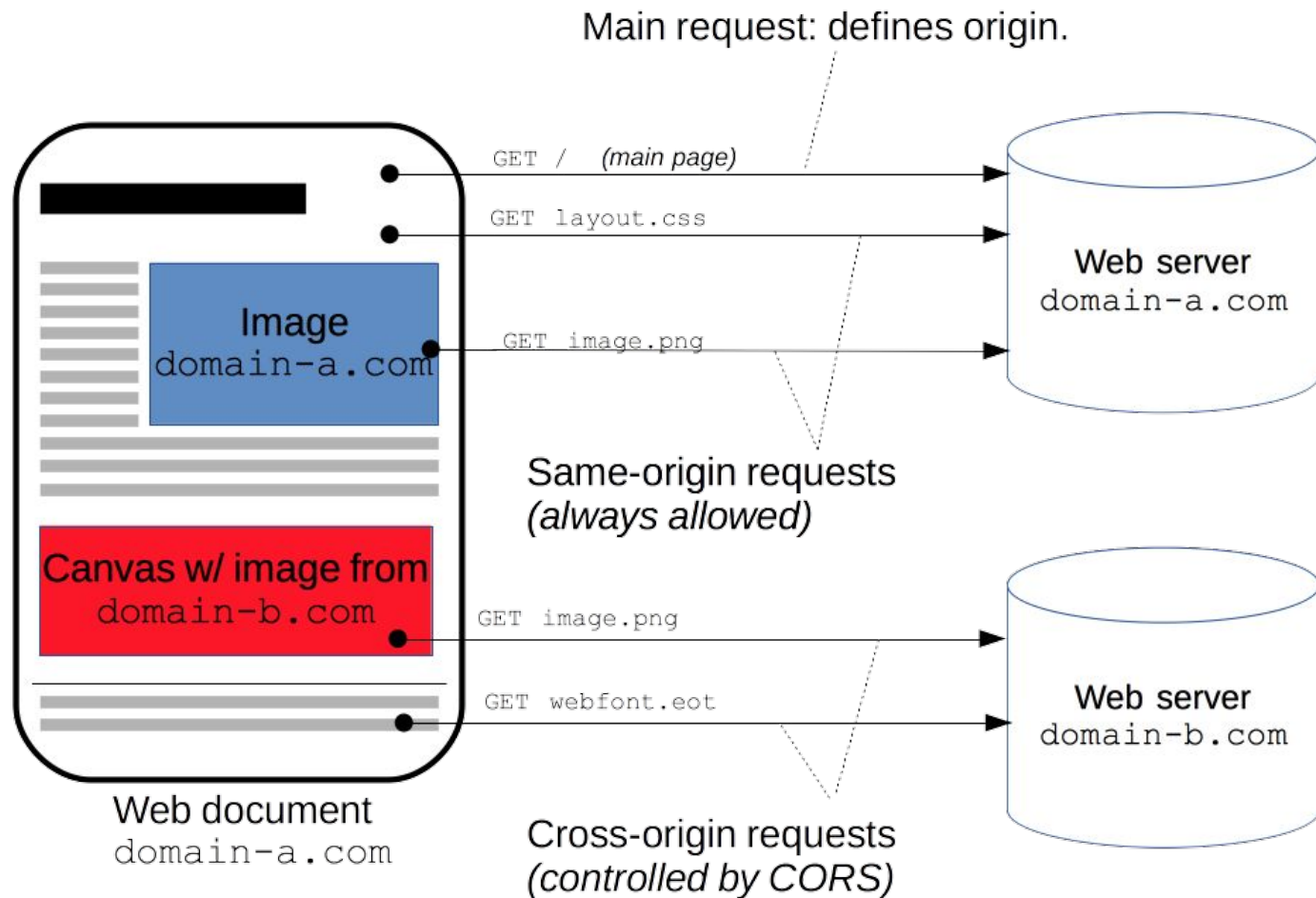
CORS

Cross-Origin Resource Sharing (CORS) — механизм, использующий дополнительные HTTP-заголовки, чтобы дать возможность агенту пользователя получать разрешения на доступ к выбранным ресурсам с сервера на источнике (домене), отличном от того, что сайт использует в данный момент. Говорят, что агент пользователя делает запрос с другого источника (**cross-origin HTTP request**), если источник текущего документа отличается от запрашиваемого ресурса доменом, протоколом или портом.

Допустимые методы для запроса:

- GET
- HEAD
- POST

CORS



REST

(REpresentational State Transfer) — это архитектура, т.е. принципы построения распределенных гипермедиа систем, того что другими словами называется World Wide Web, включая универсальные способы обработки и передачи состояний ресурсов по HTTP

Преимущества:

- Масштабируемости взаимодействия компонентов системы
(приложения)
- Общность интерфейсов
- Независимое внедрение компонентов
- Промежуточные компоненты, снижающие задержку,
усиливающие безопасность

REST

(REpresentational State Transfer) — это архитектура, т.е. принципы построения распределенных гипермедиа систем, того что другими словами называется World Wide Web, включая универсальные способы обработки и передачи состояний ресурсов по HTTP

Что нам дает REST ПОДХОД

- Масштабируемости взаимодействия компонентов системы
(приложения)
- Общность интерфейсов
- Независимое внедрение компонентов
- Промежуточные компоненты, снижающие задержку,
усиливающие безопасность

REST

Когда использовать REST

- Когда есть ограничение пропускной способности соединения
- Если необходимо кэшировать запросы
- Если система предполагает значительное масштабирование
- В сервисах, использующих AJAX

RESTful

Чтобы распределенная система считалась сконструированной по REST архитектуре (Restful), необходимо, чтобы она удовлетворяла следующим критериям:

Client-Server. Система должна быть разделена на клиентов и на серверов. Разделение интерфейсов означает, что, например, клиенты не связаны с хранением данных, которое остается внутри каждого сервера, так что мобильность кода клиента улучшается. Серверы не связаны с интерфейсом пользователя или состоянием, так что серверы могут быть проще и масштабируемы. Серверы и клиенты могут быть заменяемы и разрабатываться независимо, пока интерфейс не изменяется.

RESTful

Stateless. Сервер не должен хранить какой-либо информации о клиентах. В запросе должна храниться вся необходимая информация для обработки запроса и если необходимо, идентификации клиента.

Cache. Каждый ответ должен быть отмечен является ли он кэшируемым или нет, для предотвращения повторного использования клиентами устаревших или некорректных данных в ответ на дальнейшие запросы.

Uniform Interface. Единый интерфейс определяет интерфейс между клиентами и серверами. Это упрощает и отделяет архитектуру, которая позволяет каждой части развиваться самостоятельно.

RESTful

Stateless. Сервер не должен хранить какой-либо информации о клиентах. В запросе должна храниться вся необходимая информация для обработки запроса и если необходимо, идентификации клиента.

Cache. Каждый ответ должен быть отмечен является ли он кэшируемым или нет, для предотвращения повторного использования клиентами устаревших или некорректных данных в ответ на дальнейшие запросы.

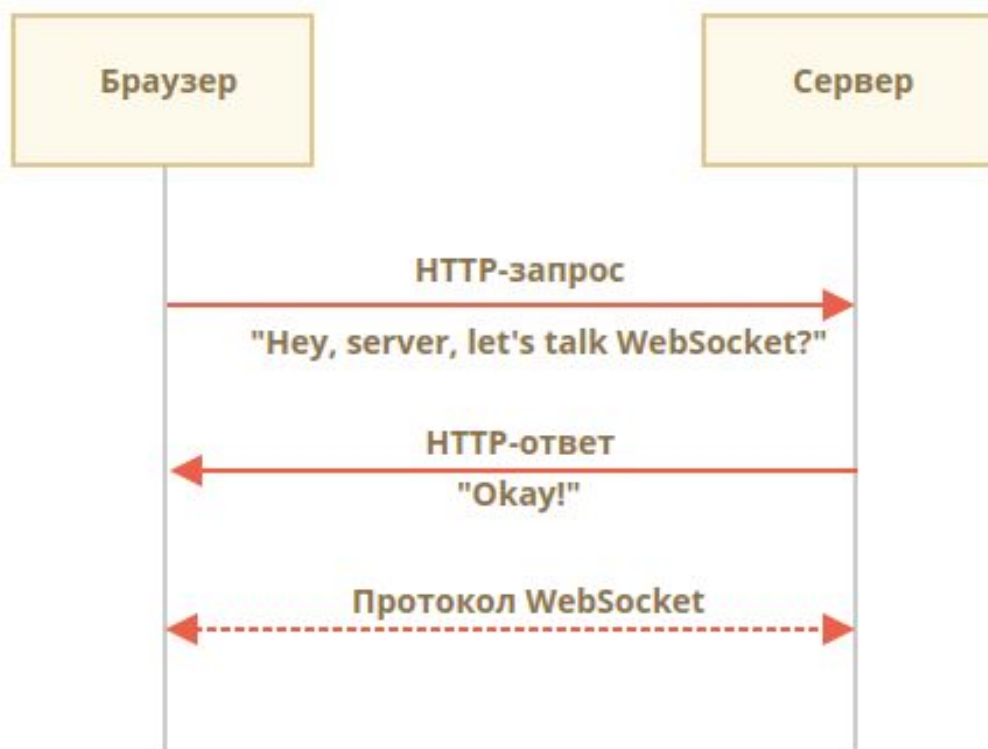
Uniform Interface. Единый интерфейс определяет интерфейс между клиентами и серверами. Это упрощает и отделяет архитектуру, которая позволяет каждой части развиваться самостоятельно.

WebSocket

Протокол **WebSocket** («веб-сокет»), обеспечивает возможность обмена данными между браузером и сервером через постоянное соединение. Данные передаются по нему в обоих направлениях в виде «пакетов», без разрыва соединения и дополнительных HTTP-запросов.

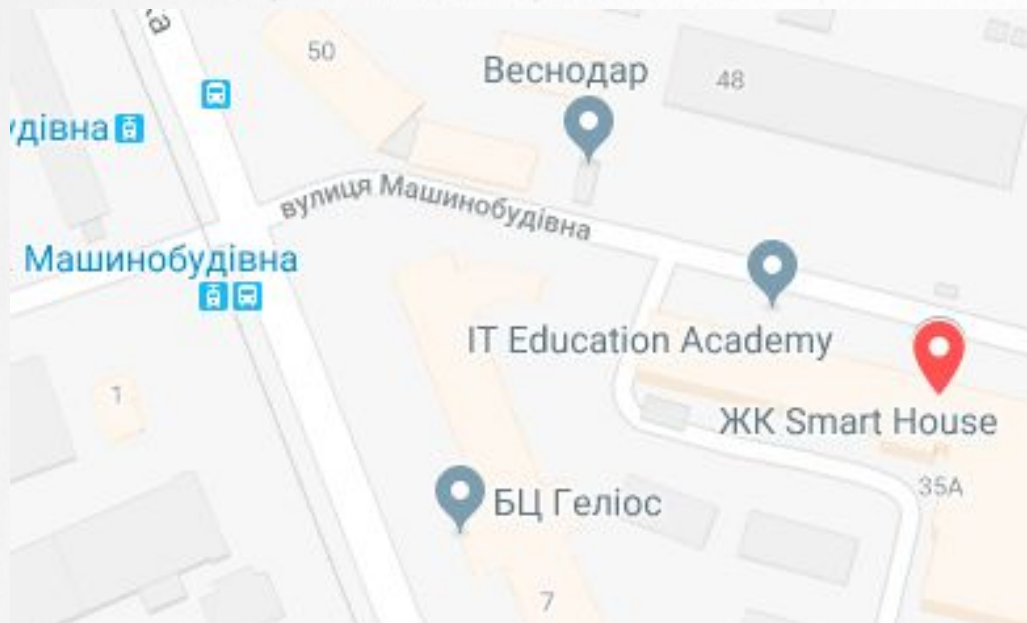
WebSocket особенно хорош для сервисов, которые нуждаются в постоянном обмене данными, например онлайн игры, торговые площадки, работающие в реальном времени, и т.д.

WebSocket



События WebSocket

- `open` – соединение установлено,
- `message` – получены данные,
- `error` – ошибка,
- `close` – соединение закрыто.



КОНТАКТНЫЕ ДАННЫЕ

ITEA

ЖК "Smart House", ул.
Машиностроительная, 41 (м.
Берестейская)

ЖК «Корона» улица
Срибнокильская,1
м. Позняки

+38 (044) 599-01-79

facebook.com/Itea

info@itea.ua

itea.ua



IT EDUCATION
ACADEMY

Вул. Срібнокільська, 1 (офіс 269)

ЖК «Корона» 2 під'їзд, 2 поверх

Від ст.м. Позняки — 7 хвилин пішки

