

# AJAX



Web.Dev.  
Courses

[sp.courses.dp.ua](http://sp.courses.dp.ua)

В чём проблема?

# Пример: «простой чат»

**Вариант** [http://courses.dp.ua/demo/ajax\\_1/](http://courses.dp.ua/demo/ajax_1/)

**№1  
Вариант** [http://courses.dp.ua/demo/ajax\\_2/](http://courses.dp.ua/demo/ajax_2/)

**№2  
Вариант** [http://courses.dp.ua/demo/ajax\\_3/](http://courses.dp.ua/demo/ajax_3/)

**№3**

*Есть необходимость обновлять только часть страницы. При этом данных которые необходимо разместить на момент загрузки не было. Возникает необходимость сделать дополнительные сетевые запросы, чтобы получить нужные данные. Причём всё это будет происходить когда пользователь уже взаимодействует со страницей в своём браузере.*

# Что делать?

The screenshot shows a web interface titled "AJAX DEMO (v.1.0)". It features a form with a text input field and a "Отправить" button. Below the form is a message log displaying "02.02.2017 13:17:16 : Hello world!". A copyright notice "Copyright (c) 2017" is visible at the bottom right. Two callout boxes on the right side of the interface provide instructions: an orange box labeled "Не обновлять" points to the form, and a purple box labeled "Обновлять" points to the message log.

*Хорошо бы обновлять часть страницы с сообщениями, а часть с формой не обновлять пока посетитель сам не отправит новое сообщение.*

AJAX

# AJAX

*(Asynchronous Javascript and XML — «асинхронный JavaScript и XML»)*



*За изменение страницы в браузере пользователя отвечает JavaScript, но до этого момента JavaScript изменял страницу только на основе данных полученные еще при загрузке страницы в браузер и/или в зависимости от действий пользователя. Получить какие-то новые (дополнительные) данные JavaScript не мог.*

# AJAX

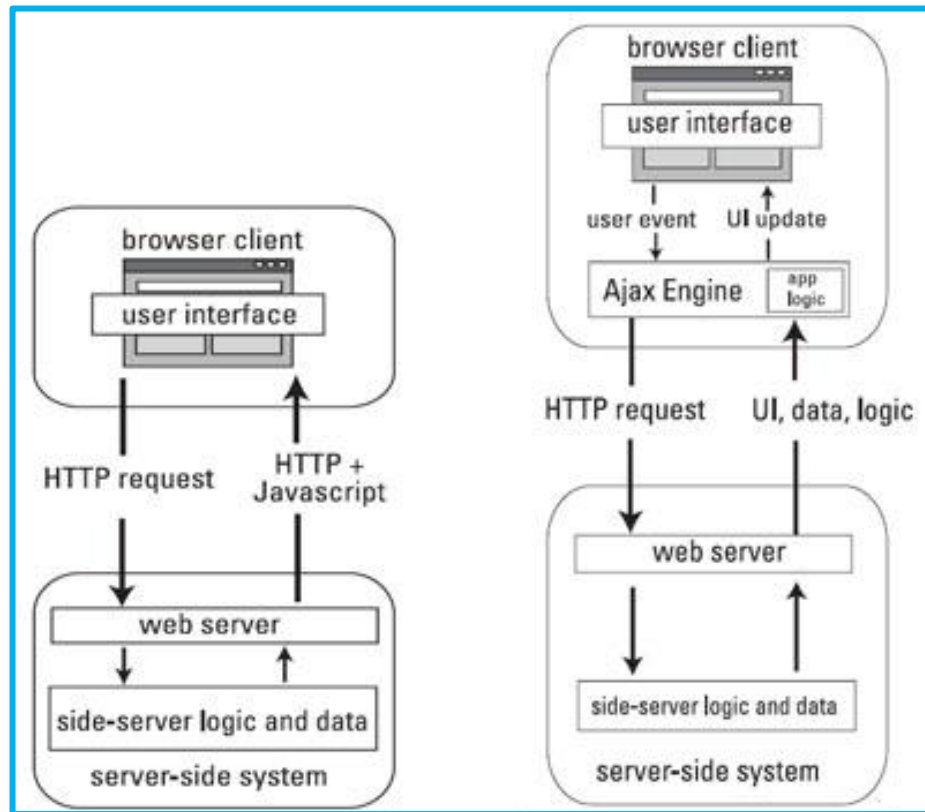
*(Asynchronous Javascript and XML — «асинхронный JavaScript и XML»)*



*С появлением в браузерах специального объекта **XMLHttpRequest** у **JavaScript** появилась возможность делать HTTP-запросы к сайтам, и изменять страницу уже на основе данных которых не было при загрузке странице. Т.е. дозагружать HTML или другие данные и вставлять их на страницу.*

# AJAX

Asynchronous Javascript And XML



*Идея заложенная в **AJAX** - не перезагружая страницу полностью запросить у сервера данные и вставить их в дерево документа.*



JSON

# JSON (JavaScript Object Notation)

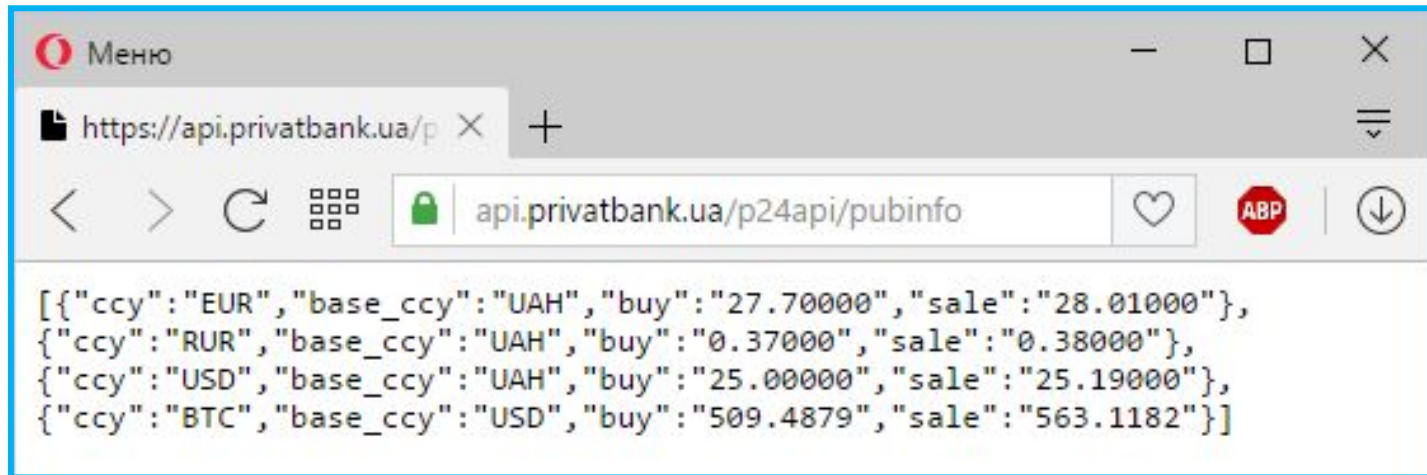
*JSON - текстовый формат обмена данными, удобный для чтения и написания как человеком, так и компьютером. Основан на синтаксисе (правилах записи) массивов в JavaScript (ECMAScript 3). Формат поддерживается практически во всех современных языках программирования.*

```
'{ "name": "Вася", "age": 35, "isAdmin": false, "friends": [0,1,78,99] }';
```

<http://www.json.org/json-ru.html>

*Сохранение объекта/массива в строковом виде и последующее восстановление объекта еще называют **сериализацией**.*

# JSON в интернете



<https://api.privatbank.ua/p24api/pubinfo?json&exchange&coursid=5>

***JSON** является популярным форматом для обмена информацией в Интернете. Большое количество сайтов позволяют получить информацию в формате **JSON** для более удобной её обработки.*

# JSON в реальности

```
1 <script>
2
3   var usd_sum = prompt("Введите цену в долларах: ");
4
5   var xhr = new XMLHttpRequest();
6
7   xhr.open("GET", "https://api.privatbank.ua/p24api/pubinfo?json&exchange&coursid=5", true);
8
9   xhr.onload = function() {
10      var result = xhr.responseText;
11      result = JSON.parse(result);
12      var uah_sum = parseFloat(result[2]["sale"]) * usd_sum;
13
14      alert(usd_sum + "$ => " + uah_sum + " грн.");
15   }
16
17   xhr.onerror = function() {
18      alert("Ошибка загрузки данных.");
19   }
20
21   xhr.send();
22
23 </script>
```

*Скрипт для своей работы использует актуальные курсы валют Приватбанка.*

Документация:

<https://api.privatbank.ua/#p24/exchange>

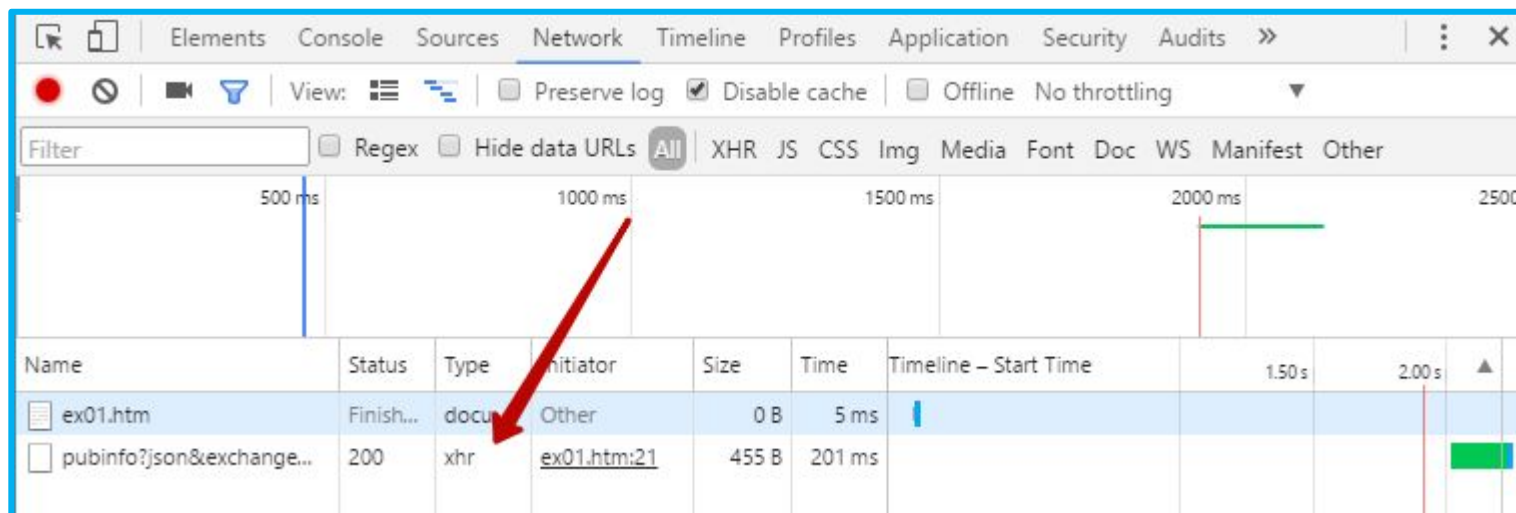


## Объект XMLHttpRequest

*XMLHttpRequest* – поддерживает событийную модель, и в зависимости от развития ситуации генерирует те или иные события.

**Синхронный запрос** – при котором браузер ждёт ответа, скрипт при этом «замирает» до прихода ответа. **Асинхронный** – скрипт продолжает выполняться, при поступлении ответа будет вызвана функция зарегистрированная как обработчик события **onload**.

# Объект XMLHttpRequest



*В консоли разработчика хорошо заметны запросы которые делались через **XMLHttpRequest** – по характерной метке **type** равной **xhr**.*

Что-то много кода...

# Функции асинхронной загрузка данных jQuery

.load()

.get()

.post()

.getJSON()

**.ajax()**

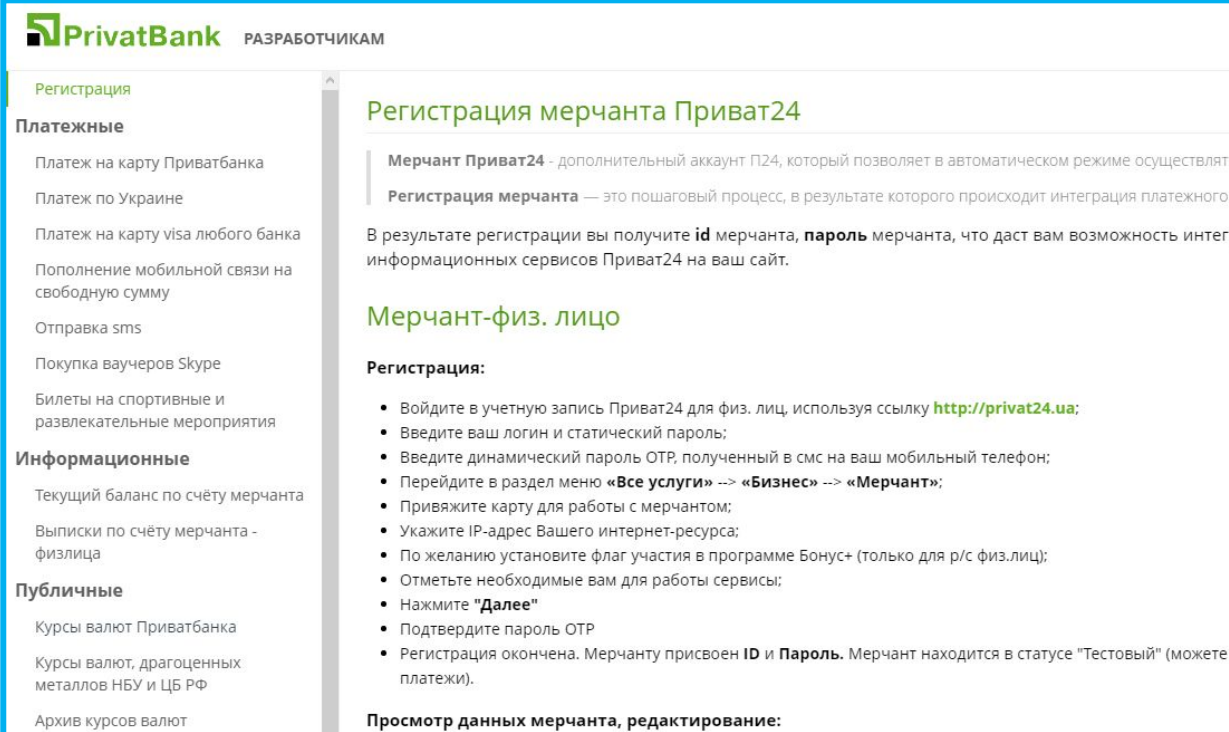
*jQuery предоставляет несколько функций для совершения AJAX запросов. Но можно обойтись одной **.ajax()**, все остальные лишь её обёртки.*

<http://api.jquery.com/category/ajax/>



# API

API (интерфейс программирования приложений, интерфейс прикладного программирования) ([англ. application programming interface<sup>1</sup>](#)) — Набор правил, которые определяют как правильно общаться с другим сайтом/программой/системой.



The screenshot shows the PrivatBank developer portal. The main heading is 'Регистрация мерчанта Приват24'. Below the heading, there is a brief description of the merchant account and the registration process. A list of steps for registration is provided, including logging in, entering a static password, a dynamic OTP, navigating to the 'Business' section, linking a card, specifying an IP address, and confirming the password. The page also includes a sidebar with navigation options like 'Платежные' and 'Информационные' services.

**Регистрация мерчанта Приват24**

Мерчанти Приват24 - дополнительный аккаунт П24, который позволяет в автоматическом режиме осуществлять платежи.

Регистрация мерчанта — это пошаговый процесс, в результате которого происходит интеграция платежного сервиса Приват24 на ваш сайт.

В результате регистрации вы получите **id** мерчанта, **пароль** мерчанта, что даст вам возможность интеграции информационных сервисов Приват24 на ваш сайт.

**Мерчанти-физ. лица**

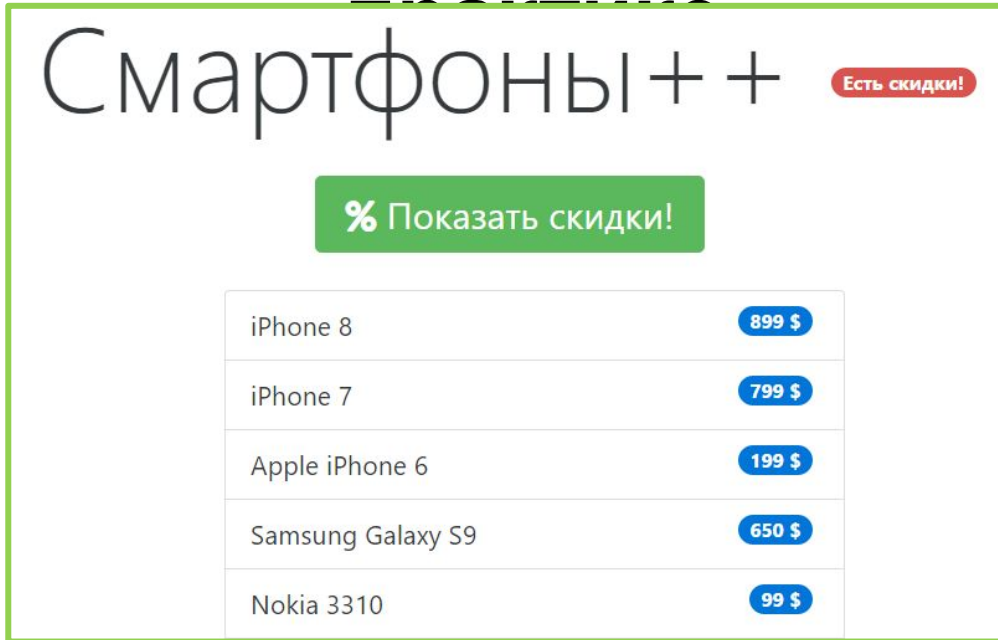
**Регистрация:**

- Войдите в учетную запись Приват24 для физ. лиц, используя ссылку <http://privat24.ua>;
- Введите ваш логин и статический пароль;
- Введите динамический пароль OTP, полученный в смс на ваш мобильный телефон;
- Перейдите в раздел меню «Все услуги» --> «Бизнес» --> «Мерчанти»;
- Привяжите карту для работы с мерчантом;
- Укажите IP-адрес Вашего интернет-ресурса;
- По желанию установите флаг участия в программе Бонус+ (только для р/с физ.лиц);
- Отметьте необходимые вам для работы сервисы;
- Нажмите "Далее"
- Подтвердите пароль OTP
- Регистрация окончена. Мерчанту присвоен ID и Пароль. Мерчант находится в статусе "Тестовый" (может осуществлять платежи).

Просмотр данных мерчанта, редактирование:

<https://api.privatbank.ua/>

# jQuery + AJAX на



Смартфоны + + Есть скидки!

% Показать скидки!

iPhone 8	899 \$
iPhone 7	799 \$
Apple iPhone 6	199 \$
Samsung Galaxy S9	650 \$
Nokia 3310	99 \$

Скачайте

заготовку:

<http://files.courses.dp.ua/web/23/ex03.html>

*Пересчитаем цены в гривневые, за основу курса возьмём данные Приватбанка.*

# jQuery + AJAX на

```
31 <script
32 src="https://code.jquery.com/jquery-1.12.4.min.js"
33 integrity="sha256-ZosEbRLbNQzLpnKIKEdrPv71Oy9C27hHQ+Xp8a4MxAQ="
34 crossorigin="anonymous"></script>
35 <script>
36   var url = "https://api.privatbank.ua/p24api/pubinfo?json&exchange&coursid=5";
37
38   $(function() {
39     $.getJSON(url, function(data) {
40       var kurs = data[2]["sale"];
41       $("li > span").each(function(index, value) {
42         var uah_price = Math.floor(parseInt(value.innerHTML) * kurs);
43         value.innerHTML = uah_price + " грн.";
44       });
45     });
46   });
47
48 </script>
```

iPhone 8	24902 грн.
iPhone 7	22132 грн.
Apple iPhone 6	5512 грн.
Samsung Galaxy S9	18005 грн.
Nokia 3310	2742 грн.



*Пересчитаем цены в гривневые, за основу курса возьмём данные Приватбанка.*

# Кросс-доменные запросы

# Кросс-доменные

## запросы

API с курсами валют есть и у НБУ:

[https://bank.gov.ua/control/uk/publish/article?art\\_id=38441973](https://bank.gov.ua/control/uk/publish/article?art_id=38441973)

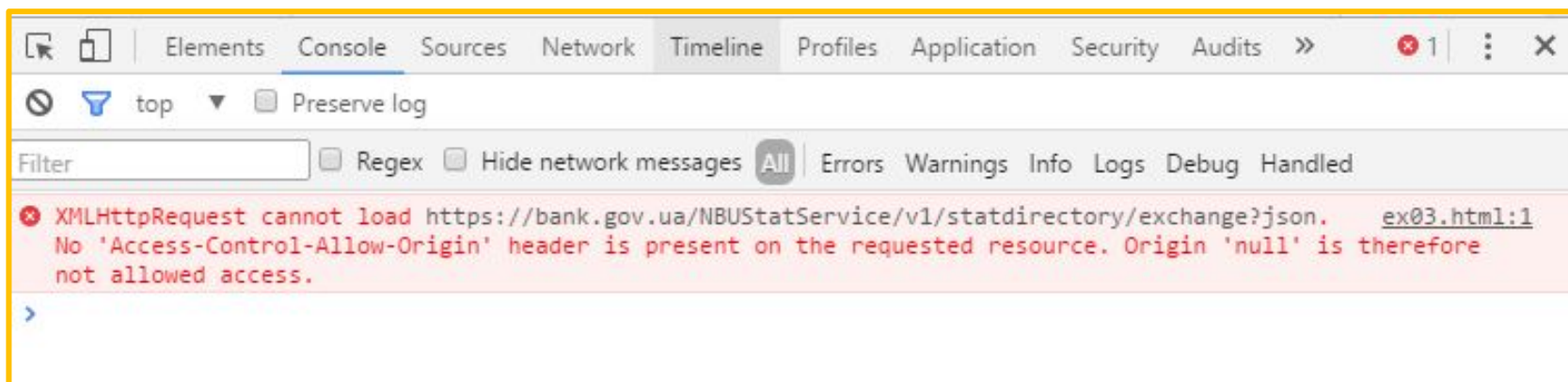
Почему мы не можем загрузить JSON с курсами по указанному там адресу:

<https://bank.gov.ua/NBUStatService/v1/statdirectory/exchange?json>

```
35 <script>
36   var url = "https://bank.gov.ua/NBUStatService/v1/statdirectory/exchange?json";
37
38   $(function(){
39     $.getJSON(url, function(data){
40       var kurs = data[2]["sale"];
41       $("li > span").each(function(index, value){
42         var uah_price = Math.floor(parseInt(value.innerHTML) * kurs);
```



# Кросс-доменные запросы



*Попытаемся загрузить данные от НБУ...*

# Кросс-доменные запросы

*Кросс-доменные запросы (т.е. запросы к другому домену, не к тому с которого загружен скрипт) проходят контроль безопасности (который осуществляет браузер).*

*Чтобы страница могла быть доступна через кросс-доменные запросы (читай **AJAX** запросы к страницам других сайтов), страница должна сама сказать об этом, а именно установить в **HTTP** ответе заголовок **Access-Control-Allow-Origin**.*