

Асинхронный JavaScript | Promise

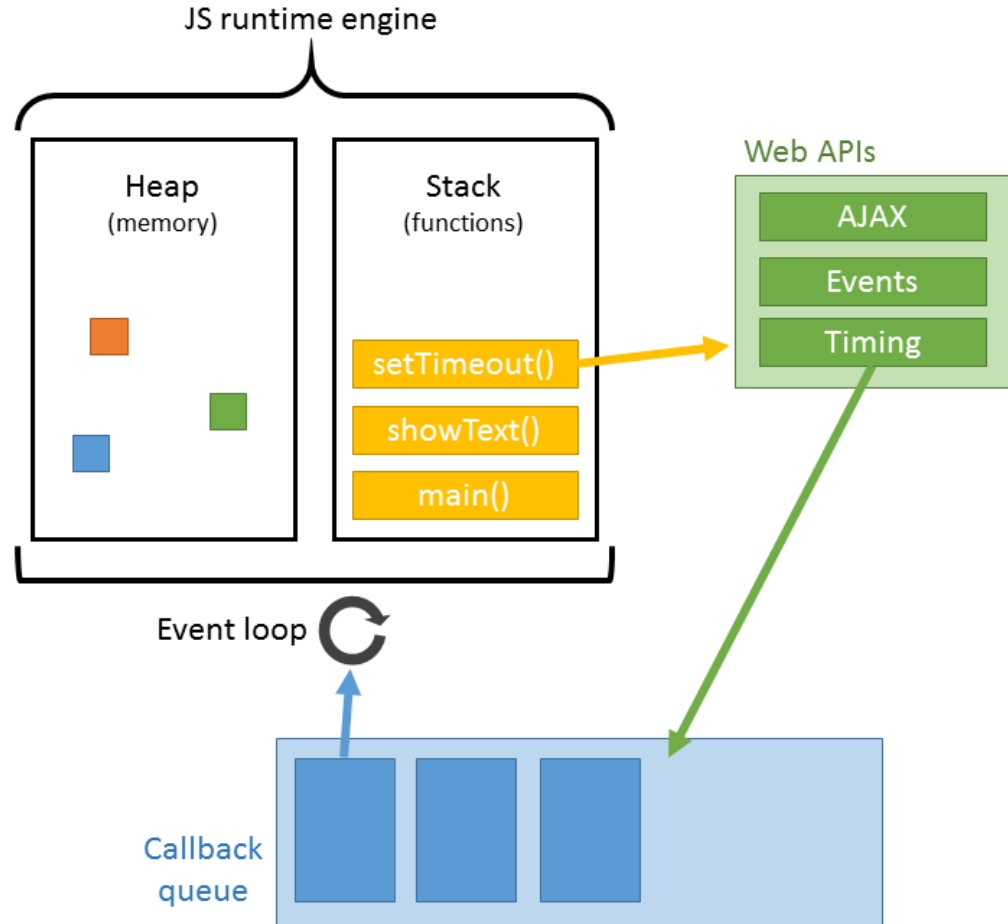


JavaScript
Courses

www.courses.dp.ua

Event Loop / JS

Вспомним об Event Loop в JavaScript



JavaScript – однопоточный язык программирования, в любой момент времени возможно выполнение только одного фрагмента кода. Но есть механизм **callback**’ов...

Подробнее: <https://developer.mozilla.org/ru/docs/Web/JavaScript/EventLoop>

Promise

Объект Promise

```
2
3 ✓ let promise = new Promise((resolve, reject) => {
4
5 ✓   setTimeout( () => {
6     let result = 2 ** 10;
7     resolve(result);
8     //reject('Operation imposible');
9   }, 5000);
10
11 });
12
13 promise.then( result => console.log('Successful, result is', result) );
14
15 promise.catch( error => console.log('Failed, error is', error) );
16
17 promise.finally( () => console.log('Promise Finished'));
18
19 console.log("After Promise");
20
```

Promise – механизм позволяющий писать асинхронный код последовательно (насколько это возможно), избегая вложенности **callback**’ов. **Promise** – объект который принимает функцию, в которой запускается асинхронная операция, при помощи параметров функции есть возможность из асинхронного кода сообщить об успешном или не успешном завершении операции (параметры **resolve**, **reject** – функции вызов которых приведёт к завершению работы **Promise**’а).

Подробнее: <https://learn.javascript.ru/promise/>

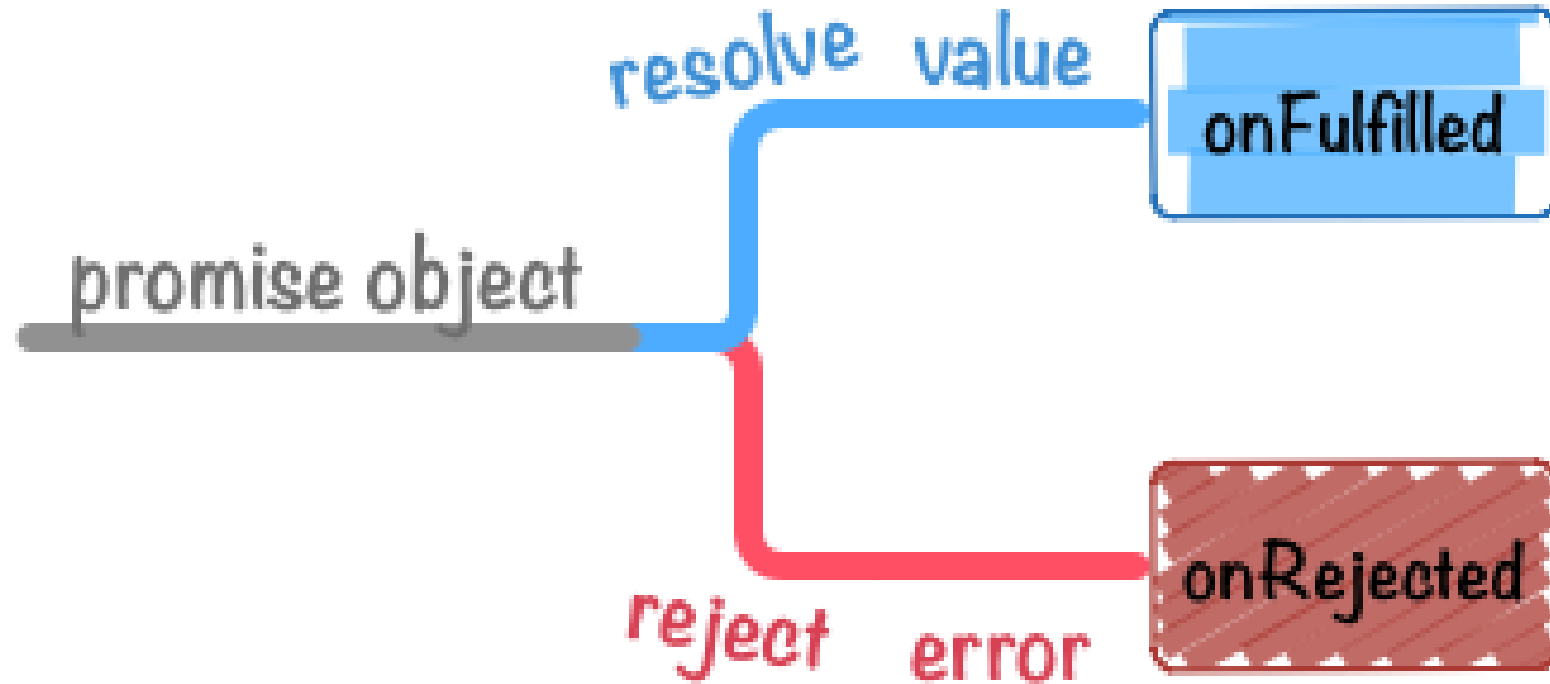
Объект Promise

```
2
3 ✓ let promise = new Promise((resolve, reject) => {
4
5 ✓   setTimeout( () => {
6     let result = 2 ** 10;
7     resolve(result);
8     //reject('Operation impossible');
9   }, 5000);
10
11 });
12
13 promise
14   .then( result => console.log('Successful, result is', result) )
15   .catch( error => console.log('Failed, error is', error) )
16   .finally( () => console.log('Promise Finished') );
17
18 console.log("After Promise");
19
```

У объекта **Promise** есть 3 полезных метода для возможности зарегистрировать функции на случай успешного завершения **Promise**'а, для случая завершения с ошибкой, и для ситуации когда код нужно выполнить как бы **Promise** не завершился (успешно или нет). Эти методы соответственно **.then()**, **.catch()** и **finally()**. Эти методы могут быть вызваны цепочкой т.к. эти методы возвращают ссылку на сами **Promise**. Функция переданная **.then()** может вернуть результат (в т.ч. другой Promise) и цепочка может опять включать **.then()** для его обработки.

Подробнее: <https://learn.javascript.ru/promise/>

Жизненный путь Promise



Жизненный путь **Promise** всегда завершается одним из двух состояний:
Fulfilled – успешное завершение, либо **Rejected** – неудачное завершение.

Promise'фикация

```
2
3 let url_nbu = 'https://bank.gov.ua/NBUStatService/v1/statdirectory/ovdp?json';
4
5 function loadJSON(url){
6
7     return new Promise((resolve, reject) => {
8
9         let xhr = new XMLHttpRequest();
10        xhr.open("GET", url);
11
12        xhr.onload = () => resolve(JSON.parse(xhr.response));
13        xhr.onerror = (error) => reject(error);
14        xhr.send()
15    });
16
17 }
18
19 loadJSON(url_nbu).then(data => {
20
21     console.log(data);
22
23 });
24
```

Процесс перевода асинхронного кода на базе **callback**'ов к применению **Promise**'ов.

Подробнее: <https://learn.javascript.ru/promisify>

Promise API

Благодаря методу **Promise.all()** есть возможность «подождать» успешного завершения всех *Promise*'ов, для последующей обработки результатов. Метод **Promise.allSettled()** –позволяет дождаться всех *Promise*'ов, независимо от результата, а **Promise.race()** позволяет дождаться только первого завершенного *Promise*'а.

Подробнее: <https://learn.javascript-api>

fetch()

AJAX на Promise'ax

fetch() – Promise-обёртка над XMLHttpRequest

```
2  
3 let url_nbu = 'https://bank.gov.ua/NBUStatService/v1/statdirectory/ovdp?json';  
4  
5 fetch(url_nbu)  
6   .then(result => result.json())  
7   .then(result => console.log(result))  
8   .catch(error => console.log("Error:", error));  
9  
10
```

Функция **fetch()** – выполняет AJAX-запросы, возвращая **Promise**, который завершится с поступлением ответа на запрос или завершится с ошибкой, если запрос будет неудачный.

Подробнее: <https://learn.javascript.ru/fetch/>

Немного практики #1

Геолокация в Теории



Широта == Latitude

Долгота == Longitude

```
{ ..., latitude: 48.4767, longitude: 35.0543, ... };
```

Геолокация на практике

```
2
3 // 'Classic' version
4 navigator.geolocation.getCurrentPosition( position => {
5     console.log('Your position: ', position.coords);
6 }, error => {
7     console.log('Geolocation error:', error);
8 })
9
10 // With Promise
11 const getPosition = () => {
12     return new Promise( (resolve, reject) => {
13         navigator.geolocation.getCurrentPosition( position => {
14             resolve(position.coords);
15         }, error => {
16             reject(error);
17         })
18     });
19 };
20
21 getPosition().then( position => console.log('Your position', position));
22
```

У браузера есть возможность узнать координаты пользователя на местности. Для этого мы можем воспользоваться методом **navigator.geolocation.getCurrentPosition()**. Но важно **проверять поддерживает ли браузер геолокацию** проверяя наличие свойства **geolocation** объекта **navigator**.

Подробнее: <https://developer.mozilla.org/ru/docs/Web/API/Geolocation/getCurrentPosition>

Немного о статических карта на примере Here Map

```
2  
3 let map_url = 'https://image.maps.api.here.com/mia/1.6/mapview?  
app_id=oZmMWRV4tAjQmgkxBvF0&app_code=x5pKHqifhw1mnS_zBTIFsA&z=11&w=800&  
h=600&c=48.2758,35.0131';  
4  
5 document.write(`<img src='${map_url}'>`);  
6
```

https://image.maps.api.here.com/mia/1.6/mapview?app_id=oZmMWRV4tAjQmgkxBvF0&app_code=x5pKHqifhw1mnS_zBTIFsA&z=11&w=800&h=600&c=48.2758,35.0131

Сервис **Here Map** предоставляет возможность размещать на наших страницах картографические материалы, управляя позицией и масштабом отображения.

Подробнее: https://developer.here.com/documentation/map-image/dev_guide/topics/examples-map-scheme-1.html

Немного практики #2

Weather forecast

[Home](#) / [Weather forecast](#)[↗ Current location](#)

Current weather and forecasts in your city

[Main](#)[Daily](#)[Hourly](#)[Chart](#)[Map](#)

Weather and forecasts in Dnipropetrovsk, UA

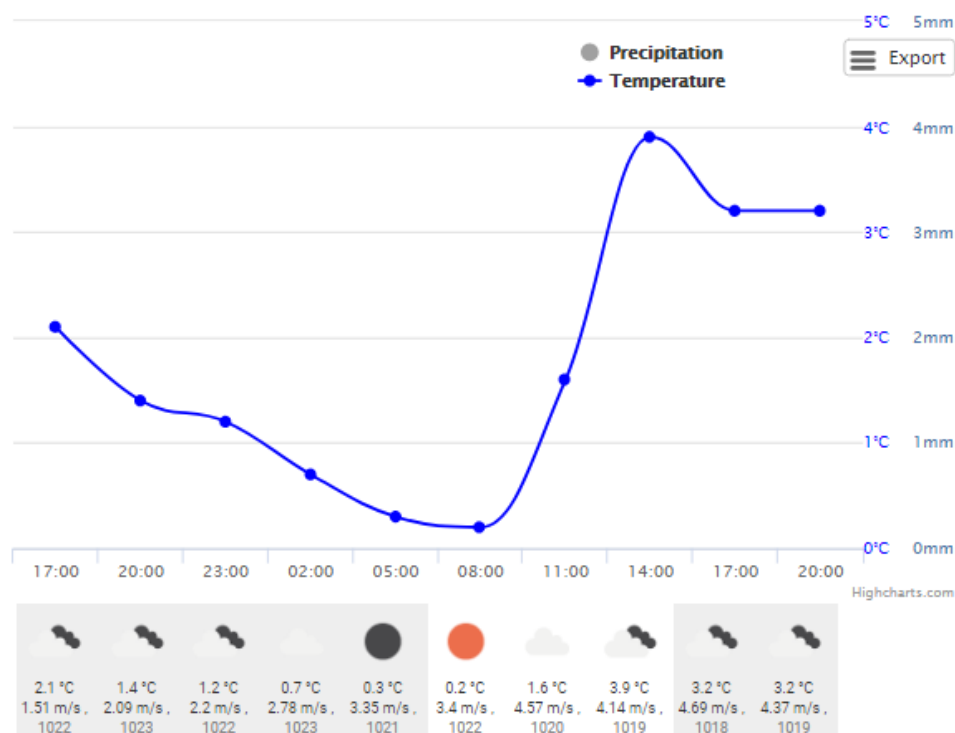
Weather in
Dnipropetrovsk, UA 4 °C

Fog

16:03 Dec 11 [Wrong data?](#)

Wind	Calm, 1.5 m/s, East (88)
Cloudiness	Overcast clouds
Pressure	1021 hpa
Humidity	100 %
Sunrise	07:21
Sunset	15:45
Geo coords	[48.45, 34.98]

The weather forecast is displayed in accordance with your local time. Please pay attention to it when you will watch the weather in another time zone.



Погодный API сервиса OpenWeather

Сервис требует
регистрации и
использование ключа
при выполнении
запросов

<https://openweathermap.org/>

Current & Forecast weather data collection

Current weather data

[API doc](#) [Subscribe](#)

- Access current weather data for any location including over 200,000 cities
- Current weather is frequently updated based on global models and data from more than 40,000 weather stations
- Data is available in JSON, XML, or HTML format
- Available for Free and all other paid accounts

Hourly forecast ^{NEW}

[API doc](#) [Subscribe](#)

- Hourly forecast is available for 4 days
- Forecast weather data for 96 timestamps
- Higher geographic accuracy
- Forecast is available in JSON and XML
- Available for Developer, Professional and Enterprise accounts

16 day / daily forecast

[API doc](#) [Subscribe](#)

- 16 day forecast is available at any location or city
- 16 day forecast includes daily weather
- Forecast is available in JSON and XML
- Available for all paid accounts

We have combined **Weather services** and **Satellite imagery** in a simple and fast **Agro API**. We have also launched a **Dashboard** for it - it is a **visual service** where you can easily work with satellite imagery and weather data for your fields: satellite, weather and historical data, soil temperature and moisture, accumulated temperature and precipitation, etc.
Learn more on agromonitoring.com

Climate forecast for 30 days ^{NEW}

[API doc](#) [Subscribe](#)

- Forecast weather data for 30 days
- Based on a statistical approach to our **Historical weather data**
- Forecast is available only in JSON format
- The frequency of weather data update is 1 hour
- Available for Developer, Professional and Enterprise accounts

Bulk downloading

[API doc](#) [Subscribe](#)

- We provide number of bulk files with current weather and forecasts
- Current weather bulk is available for 209,000+ cities
- Variety of hourly and daily forecast bulks depends on frequency of data updating
- Available for Professional and Enterprise accounts

5 day / 3 hour forecast

[API doc](#) [Subscribe](#)

- 5 day forecast is available at any location or city
- 5 day forecast includes weather data every 3 hours
- Forecast is available in JSON and XML
- Available for Free and all other paid accounts

Погодный API сервиса OpenWeather

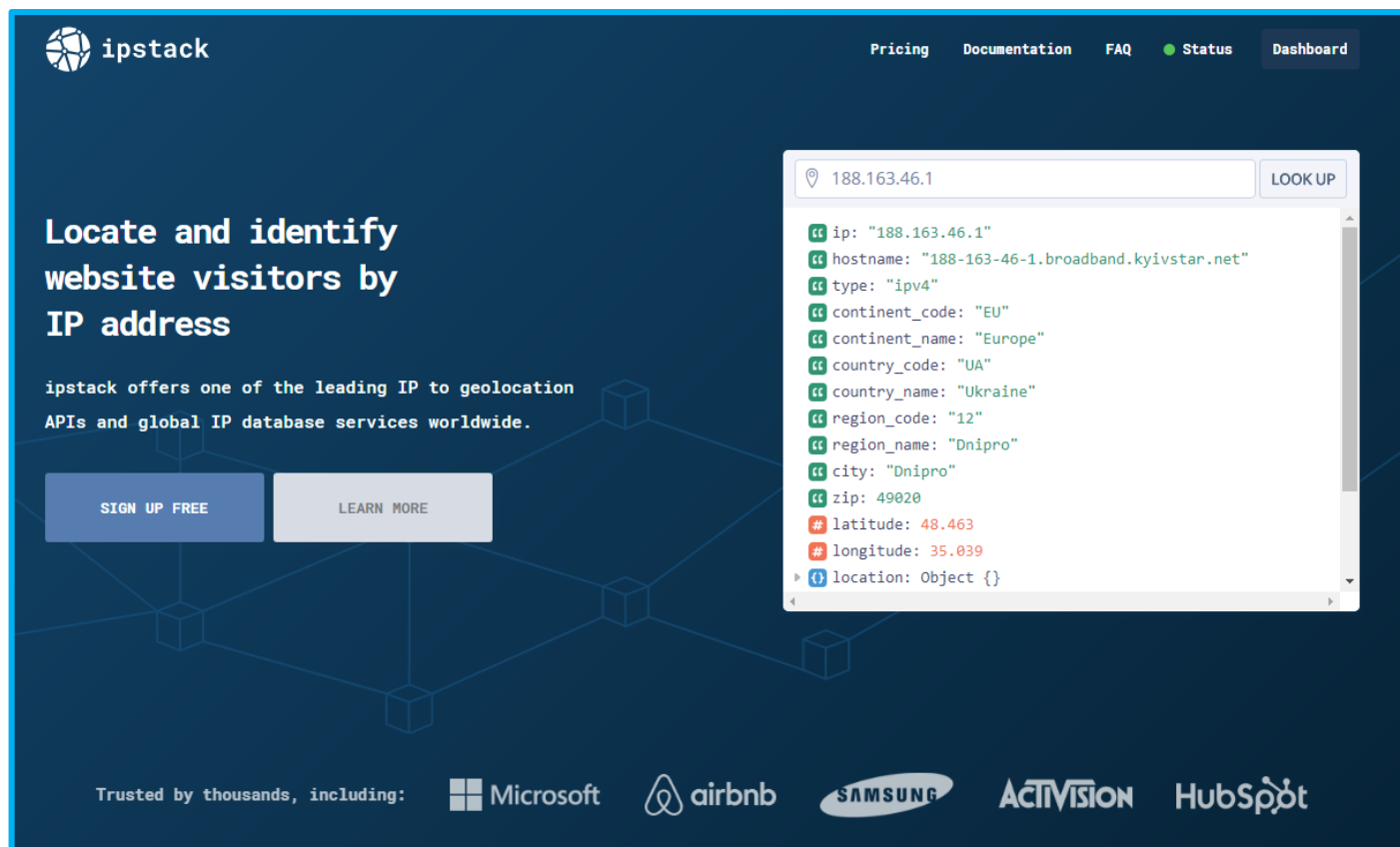
Сервис требует
регистрации и
использование ключа
при выполнении
запросов

9f118bfa230072d3603183e520cea4af

<https://openweathermap.org/>

Полезный API WHOIS-сервиса

API получения информации об IP адресе



The screenshot displays the ipstack website interface. At the top, the ipstack logo is on the left, and navigation links for Pricing, Documentation, FAQ, Status, and Dashboard are on the right. The main heading reads "Locate and identify website visitors by IP address". Below this, a subtext states "ipstack offers one of the leading IP to geolocation APIs and global IP database services worldwide." Two buttons, "SIGN UP FREE" and "LEARN MORE", are visible. On the right side, a search box contains the IP address "188.163.46.1" with a "LOOK UP" button. Below the search box, a JSON response is shown, detailing geolocation data for the specified IP.

```
{
  "ip": "188.163.46.1",
  "hostname": "188-163-46-1.broadband.kyivstar.net",
  "type": "ipv4",
  "continent_code": "EU",
  "continent_name": "Europe",
  "country_code": "UA",
  "country_name": "Ukraine",
  "region_code": "12",
  "region_name": "Dnipro",
  "city": "Dnipro",
  "zip": "49020",
  "latitude": 48.463,
  "longitude": 35.039,
  "location": Object {}
}
```

At the bottom, a section titled "Trusted by thousands, including:" lists logos for Microsoft, airbnb, SAMSUNG, ACTIVISION, and HubSpot.

*Сервис позволяют получить информацию в формате **JSON**. Но необходимо зарегистрироваться и получить ключ*

<https://ipstack.com/>

Домашнее задание
/узнать

Современный учебник Javascript

Перед вами учебник по JavaScript, начиная с основ, включающий в себя много тонкостей и фишек JavaScript/DOM.

[смотреть на Github](#)

Поделиться:

[НАЙТИ](#)

Содержание

Первые две части посвящены JavaScript и его использованию в браузере. Затем идут дополнительные циклы статей на разные темы.

Предварительные знания – лучший помощник в обучении, поэтому к следующему занятию жду, что **пройдёте разделы 1-й части 9.1-9.6**

<http://learn.javascript.ru/>

+1 Книга



Домашнее задание
/сделать

Домашнее задание #Е.1

<https://api.privatbank.ua/p24api/infrastructure?json&atm&address=&city=%D0%94%D0%BD%D1%96%D0%BF%D1%80%D0%BE>



По указанному выше URL возвращаются данные о банкоматах ПриватБанка в Днепре. Ваша задача при помощи геолокации определить какой банкомат ближе всего к пользователю и вывести его адрес, также выведите адрес текущего положения пользователя, и расстояние в метрах между ни и банкоматом (по прямой).

П.С. Это домашнее задание должно быть не только размещено на GitHub но и выложено на хостинге (выбор хостинга - на ваше усмотрение, но рекомендую узнать о хостинге GitHub Pages).

П.П.С. Вам может помочь сервис <https://opencagedata.com/>

