

[Zapisz kopię w Evernote](#)

Ostatnia aktualizacja: 8 paź 2021

Etap 3 Tydzień 1 Dzień 2 - Obiekt request

Slajdy z prezentacji:



Obiekt request

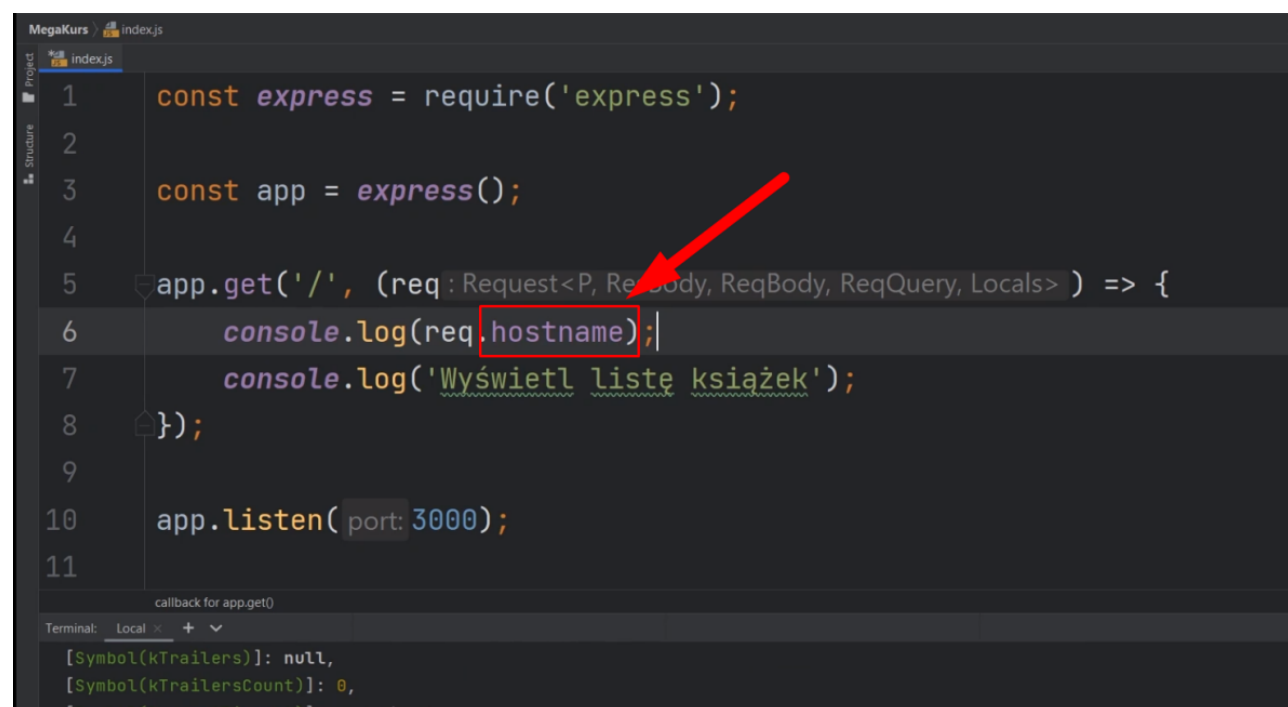
Koncepcja podobna do tej z obiektem req/request w Node.js - otrzymujemy parametr w callbacku każdego zdarzenia.

Najczęściej przyjęło się nazywać ten parametr jako **req**.

Jest on typu Request i reprezentuje każde zapytanie HTTP.

req.hostname

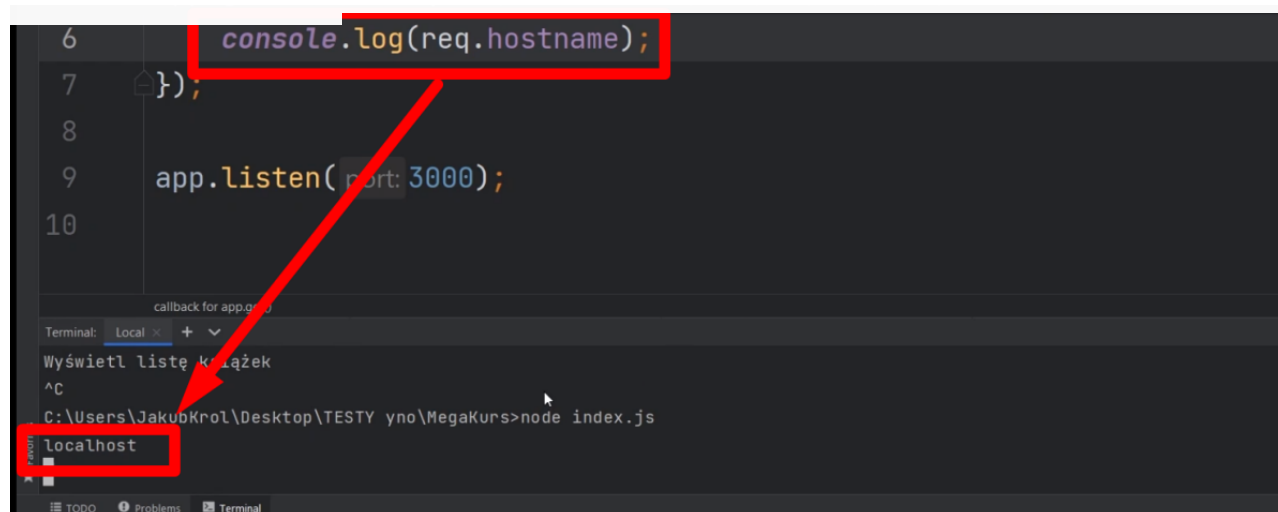
reprezentuje nazwę hosta - naszego serwera, na którym uruchomiona jest nasza aplikacja.



```
1  const express = require('express');
2
3  const app = express();
4
5  app.get('/', (req: Request<P, ReqBody, ReqQuery, Locals> ) => {
6    console.log(req.hostname);
7    console.log('Wyświetl listę książek');
8  });
9
10 app.listen(port: 3000);
11
```

Terminal: Local × + ▾

```
[Symbol(kTrailers)]: null,
[Symbol(kTrailersCount)]: 0,
[Symbol(RequestTimeout)]: undefined
```

[Warunki świadczenia usługi](#)[Polityka prywatności](#)[Zapisz kopię w Evernote](#)

The screenshot shows a code editor with the following JavaScript code:

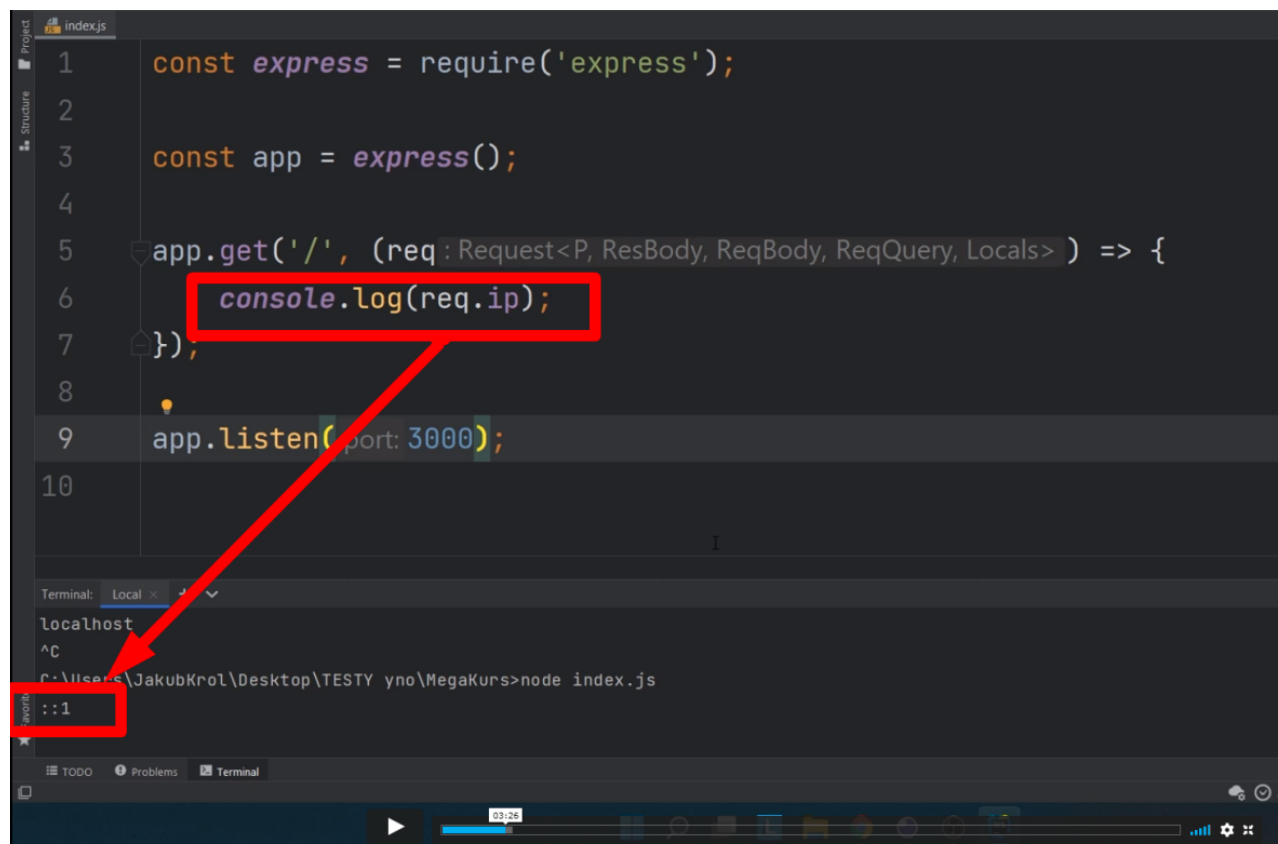
```
6 console.log(req.hostname);
7 });
8
9 app.listen(port: 3000);
10
```

The terminal window below shows the command `node index.js` being executed, and the output `localhost` is displayed. A red box highlights the `localhost` output, and a red arrow points from the `console.log(req.hostname);` line in the code to the terminal output.

req.ip

req.ip to adres IP klienta - najczęściej więc komputera lub urządzenia mobilnego, z którego przeglądarki wysłano zapytanie do naszego serwera.

Czasami serwer proxy potrafi ukryć i nadpisać informację o IP.



The screenshot shows a code editor with the following JavaScript code:

```
1 const express = require('express');
2
3 const app = express();
4
5 app.get('/', (req : Request<P, ResBody, ReqBody, ReqQuery, Locals> ) => {
6   console.log(req.ip);
7 });
8
9 app.listen(port: 3000);
10
```

The terminal window below shows the command `node index.js` being executed, and the output `:::1` is displayed. A red box highlights the `:::1` output, and a red arrow points from the `console.log(req.ip);` line in the code to the terminal output.

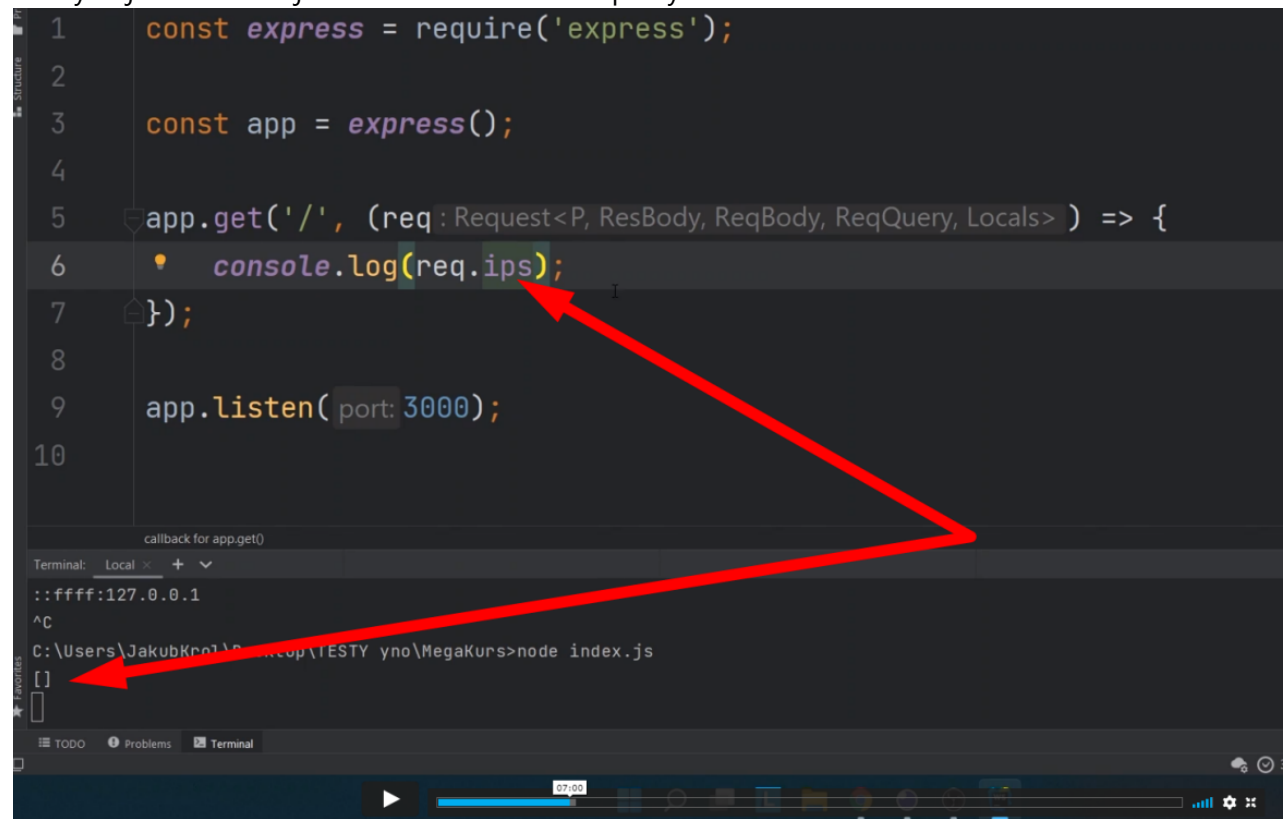
```
:::1  
::ffff:127.0.0.1
```

Zapisz kopię w Evernote

req.ips

W niektórych przypadkach (szczególnie gdy to proxy pochodzi z naszego własnego serwera) można odzyskać oryginalne IP. Znajduje się ono w tablicy req.ips.

Zazwyczaj zawiera ona jeden element - ale wiele proxy może oznaczać wiele elementów.



The screenshot shows a VS Code editor with a JavaScript file containing the following code:

```
1 const express = require('express');  
2  
3 const app = express();  
4  
5 app.get('/', (req: Request<P, ResBody, ReqBody, ReqQuery, Locals> ) => {  
6   console.log(req.ips);  
7 });  
8  
9 app.listen( port: 3000 );  
10
```

Below the editor is a terminal window with the following output:

```
Terminal: Local x + v  
::ffff:127.0.0.1  
^C  
C:\Users\JakubKro1\Source\repos\TESTY yno\MegaKurs>node index.js  
[ ]  
[ ]
```

Two red arrows point from the `req.ips` property in the code to the terminal output, illustrating the connection between the code and the actual output.

```
const express = require('express');  
const app = express();  
const port = 3000;  
  
const allowedIps = ['localhost', '127.0.0.1', ':::1', '::ffff:127.0.0.1'];
```



req. method

Przechowuje metodę HTTP jaką wykonano zapytanie.

O ile w Node.js HTTP Server była to informacja istotna o tyle w Express jest łatwiejszy sposób. Jedyne miejsce gdzie może to być przydatne, to gdy użyjemy `app.all(...)`.

```
const express = require('express');
const app = express();
const port = 3000;

app.get('/', (req, res) => {
  console.log(req.method);
});

app.listen(port);
```

konsola:

```
GET
```

req.url, req.originalUrl, req.path

Zapisz kopię w Evernote

req.protocol i req.secure

Dzięki tym informacjom można stwierdzić czy serwer używa HTTP, czy bezpiecznego połączenia HTTPS.

Często serwer nasłuchuje jednocześnie na bezpiecznym i zwykajnym porcie.

Możemy wtedy wykorzystać **req.protocol** lub **req.secure**, żeby np. wykryć czy użytkownik odwiedził stronę z adresem http:// i przekierować go wtedy na https:// .

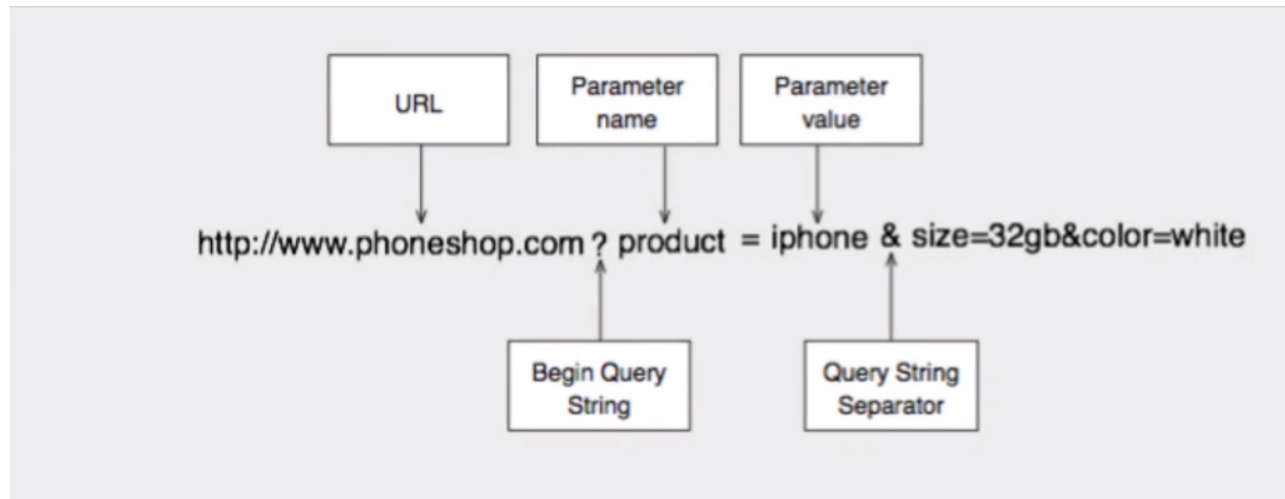
```
File Edit View Navigate Code Refactor Run Tools VCS Window Help MegaKurs - index.js
MegaKurs index.js
2
3 const app = express();
4
5 app.get('/ciekawyy', (req : Request<P, ResBody, ReqBody, ReqQuery, Locals> ) => {
6   console.log(req.protocol);
7 });
8
9 app.listen(port: 3000);

callback for app.get()

Terminal: Local x + v
^C
C:\Users\JakubKrol\Desktop\TESTY yno\MegaKurs>node index.js
{ url: '/ciekawyy', originalUrl: '/ciekawyy', path: '/ciekawyy' }
^C
C:\Users\JakubKrol\Desktop\TESTY yno\MegaKurs>node index.js
^C
C:\Users\JakubKrol\Desktop\TESTY yno\MegaKurs>node index.js
http
```

Zapisz kopię w Evernote

Parametry Adresu URL



```
< 'name=Bartek+%26+Kuba%3F'
> 'http://localhost:3000/?${qs.toString()}'
< 'http://localhost:3000/?name=Bartek+%26+Kuba%3F'

>
```

Zapisz kopię w Evernote

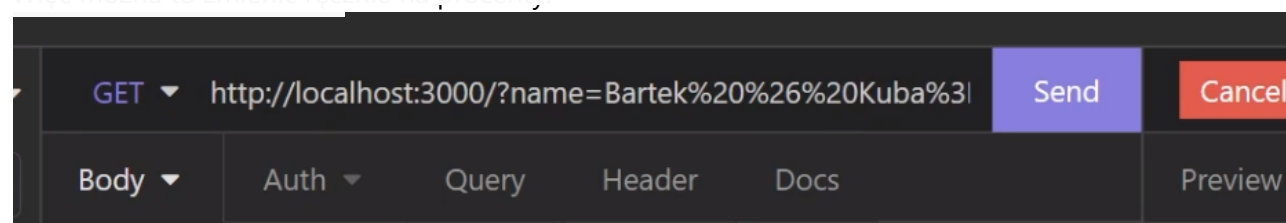
```
1  const {URLSearchParams} = require('url');
2
3  const qs = new URLSearchParams({
4    name: 'Bartek & Kuba?',
5  });
6
7  console.log(`http://localhost:3000/?${qs.toString()}`);
8
```

Terminal: Local + v

```
^C
C:\Users\JakubKrol\Desktop\TESTY yno\MegaKurs>node index.js
http://localhost:3000/?name=Bartek+%26+Kuba%3F
C:\Users\JakubKrol\Desktop\TESTY yno\MegaKurs>
```

Na powyższym obrazku jest pewien problem z plusami, które nie są dobrze od Węć można to zmienić ręcznie na procenty

Zapisz kopię w Evernote



Wstaw obraz

Zapisz kopię w Evernote

```
app.listen(port, () => {
```

[Zapisz kopię w Evernote](#)

```
C:\Users\JakubKrol\Desktop\TESTY yno\MegaKurs>node index.js  
http://localhost:3000/?name=Bartek%20%26%20Kuba%3F
```

```
C:\Users\JakubKrol\Desktop\TESTY yno\MegaKurs>node index.js  
{ name: 'Bartek & Kuba?' }  
{ name: 'MegaK!', part: 'Express.js?' }  
}
```

Zapisz kopię w Evernote

Zapisz kopię w Evernote

Zapisz kopię w Evernote

Zapisz kopię w Evernote