

[Zapisz kopię w Evernote](#)

Ostatnia aktualizacja: 8 paź 2021

Etap 3 Tydzień 1 Dzień 3 - Zaawansowany routing

Zaawansowany routing

Na czym właściwie polega routing?

Routing mówiąc krótko to jest - kierowanie czegoś gdzieś. Czyli na przykład to jest już prosty routing:

```
app.get('/', (req : Request<P, ResBody, ReqBody, ReqQuery, Locals> , res : Response<ResBody, Locals> ) => {  
  res.send( body: 'Hi!');  
});
```

Jakie mamy możliwości: **metody i ścieżki**.

Jak korzystać z routingu w Express.js?

```
const express = require('express');  
const app = express();  
const port = 3000;  
  
app.get('/', (req, res) => {  
  res.send('<h1>Hello from backend!</h1>')  
});  
  
app.listen(port);
```

Parametry ścieżek

Czyi dowolne dane przesyłane w dowolnym miejscu.

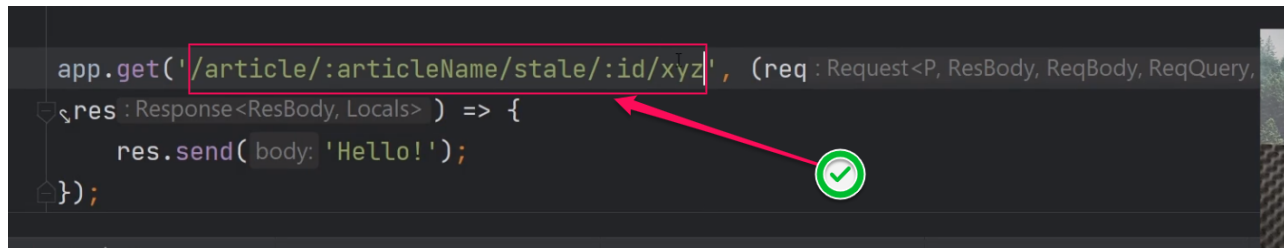
/stala/sciezka/:zmienna

/stala/inna/sciezka/:zmienna1/:zmienna1

```
app.get('/article/:articleName', (req : Request<P, ResBody, ReqBody, ReqQuery, Locals> , res : Re  
=> {  
  res.send( body: 'Hello!');  
});
```

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

```
app.get('/article/:articleName/stale/:id/xyz', (req : Request<P, ResBody, ReqBody, ReqQuery, Locals>, res : Response<ResBody, Locals>) => {  
  res.send( body: 'Hello!');  
});
```

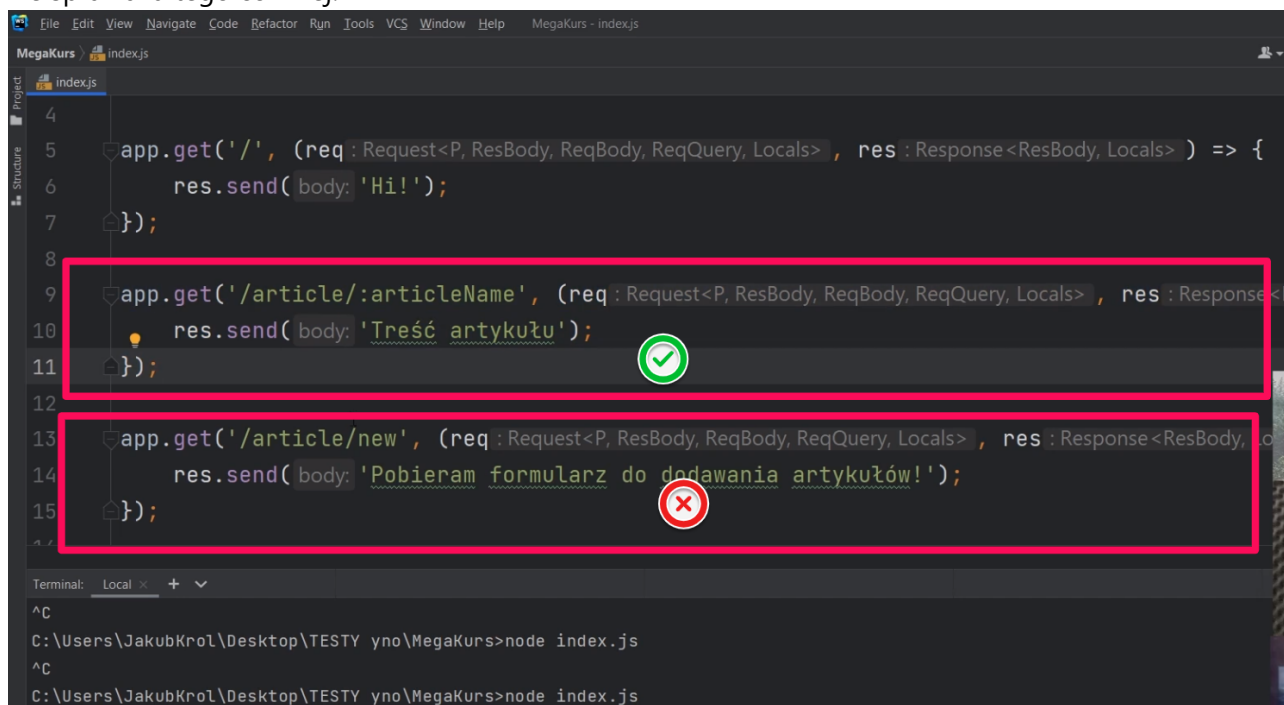


Kilka reguł: nazwy jak zmiennych; oddzielone od pozostałych elementów za pomocą /, . lub -. My zazwyczaj będziemy używali slesza czyli /

Podobne ścieżki

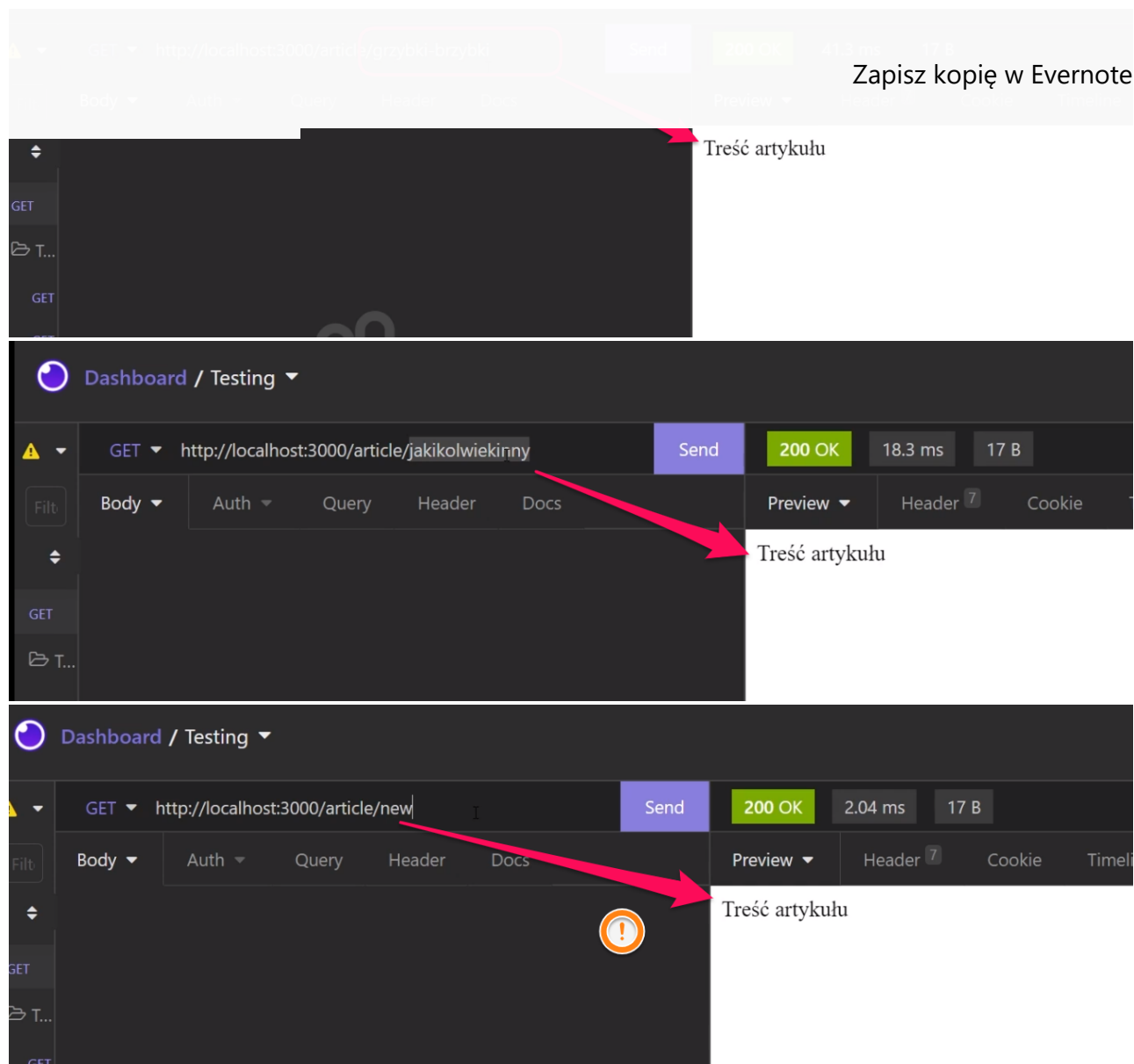
```
/hello/:name  
  
/hello/new-user
```

Jeżeli te ścieżki z powyższego obrazka, by miały takie same metody to wzięta pod uwagę zostaje pierwsza i kończy pracę. Trochę to działa jak if i else. Jeżeli pierwszy warunek zostaje spełniony to już nie sprawdza tego co niżej.



```
4  
5 app.get('/', (req : Request<P, ResBody, ReqBody, ReqQuery, Locals>, res : Response<ResBody, Locals>) => {  
6   res.send( body: 'Hi!');  
7 });  
8  
9 app.get('/article/:articleName', (req : Request<P, ResBody, ReqBody, ReqQuery, Locals>, res : Response<ResBody, Locals>) => {  
10   res.send( body: 'Treść artykułu');  
11 });  
12  
13 app.get('/article/new', (req : Request<P, ResBody, ReqBody, ReqQuery, Locals>, res : Response<ResBody, Locals>) => {  
14   res.send( body: 'Pobieram formularz do dodawania artykułów!');  
15 });
```

Terminal: Local x + v
^C
C:\Users\JakubKrol\Desktop\TESTY yno\MegaKurs>node index.js
^C
C:\Users\JakubKrol\Desktop\TESTY yno\MegaKurs>node index.js



Jeżeli ktoś się uprze i będzie chciał to tak mieć jak powyżej i żeby mu działało to trzeba będzie odwrócić kolejność. w kodzie.

```
8
9 app.get('/article/new', (req : Request<P, ResBody, ReqBody, ReqQuery, Locals> , res : Response<ResBody, Locals>
10   res.send( body: 'Pobieram formularz do dodawania artykułów!');
11 });
12
13 app.get('/article/:articleName', (req : Request<P, ResBody, ReqBody, ReqQuery, Locals> , res : Response<ResBody, Locals>
14   res.send( body: 'Treść artykułu');
15 });
```

Parametry opcjonalne

Zapisz kopię w Evernote

Mozemy również oznaczyć, że któryś parametr nie jest wymagany. Robimy tak często pod pozycjonowanie SEO, gdyż w systemie istotne jest ID, a nie tytuł.

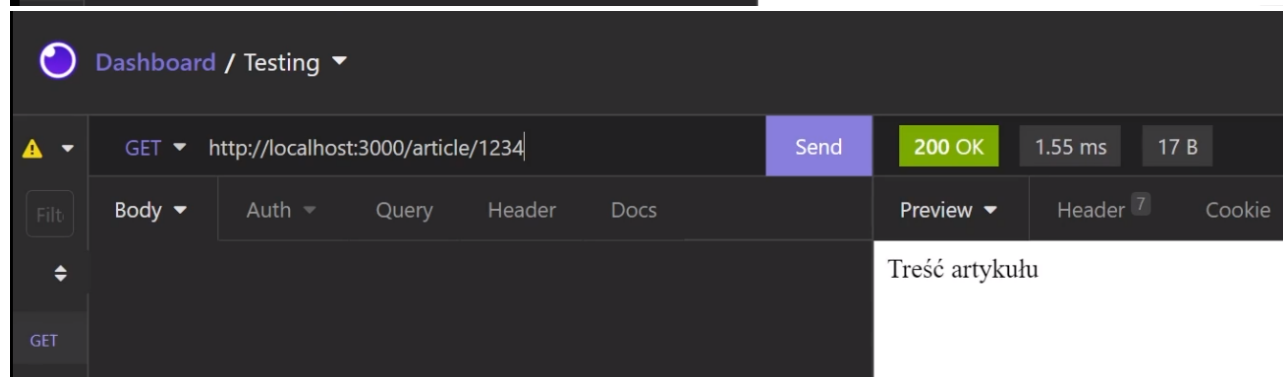
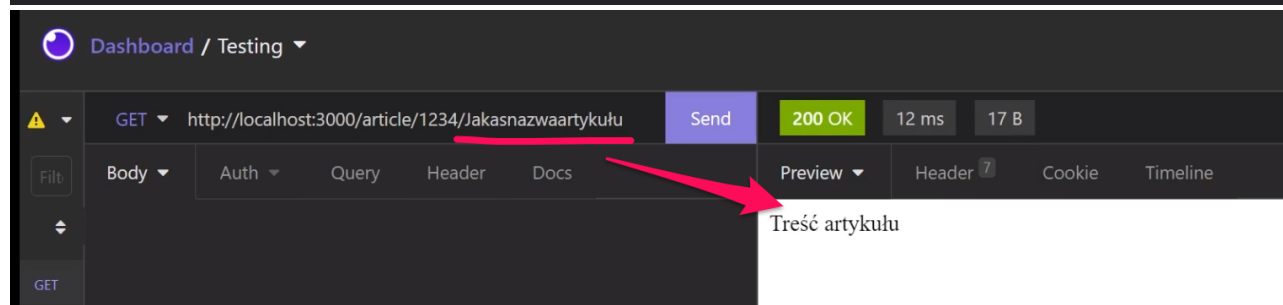
Dzięki temu ścieżka:

```
/article/:id/:title?
```

pasuje do poniższych adresów:

```
.../article/123  
.../article/123/To-jest-tytuł-artykułu
```

```
app.get('/article/:id/:articleName?', (req : Request<P, ResBody, ReqBody, ReqQuery, Locals> , res : Response<P, ResBody>) => {  
    res.send( body: 'Treść artykułu');  
});
```



Zapisz kopię w Evernote

