

[Zapisz kopię w Evernote](#)

Ostatnia aktualizacja: 8 paź 2021

Etap 3 Tydzień 1 Dzień 4 - Przesyłanie plików

Slajdy:



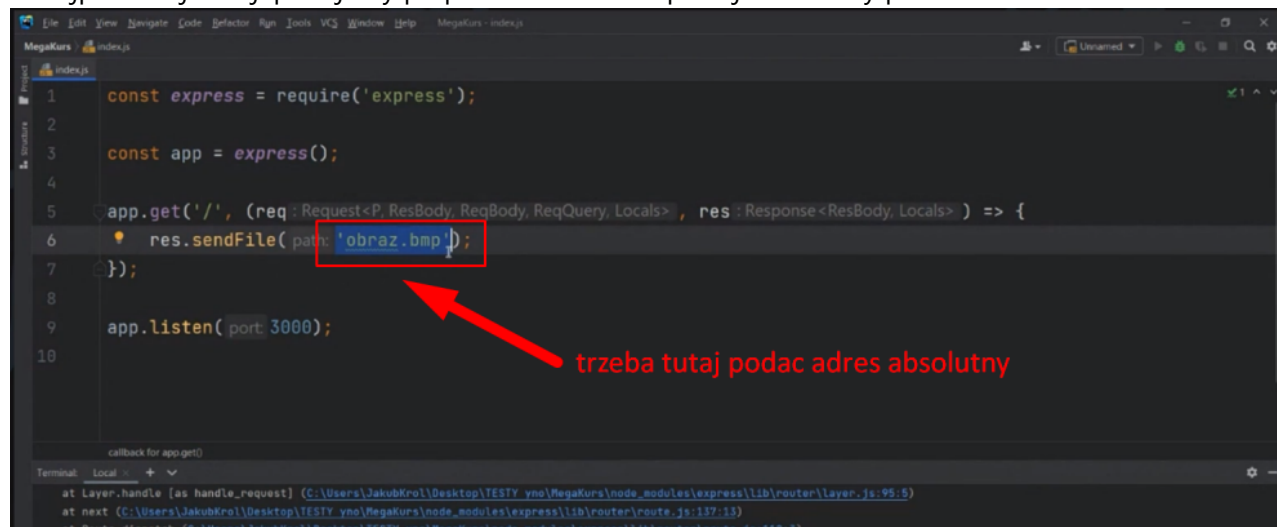
Temat przewodni: Przesyłanie plików.

Możemy przekazać Express.js informację, że chcemy przesłać do klienta plik. W przypadku standardowego serwera Node.js nie byłoby to łatwe, ponieważ trzeba ustalić typ MIME plik (<https://www.file-extension.info/pl/format/mime>), wczytać go, a następnie przesyłać lub strumieniować do przeglądarki.

Jednak w Express możemy skorzystać z metody:

```
res.sendFile();
```

W najprostszej wersji podajemy po prostu ścieżkę do pliku jaki chcemy przesłać.



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

do adresu absolutnego moduľ patch

```
app.get('/', (req : Request<P, ResBody, ReqBody, ReqQuery, Locals> , res : Response<ResBody, Locals> ) => {
  res.sendFile( path: 'obraz.bmp' );
});

app.listen( port: 3000 );
```

1 const express = require('express');
 2 const {join} = require('path');
 3
 4 const app = express();
 5
 6 app.get('/', (req : Request<P, ResBody, ReqBody, ReqQuery, Locals> , res : Response<ResBody, Locals>) => {
 7 res.sendFile(join(__dirname, 'obraz.bmp'));
 8 });
 9
 10 app.listen(port: 3000);
 11

mówimy: połącz te dwie ścieżki i dzięki temu powstanie ścieżka absolutna

```
at Layer.handle [as handle_request] (C:\Users\JakubKrol\Desktop\TESTY_yno\MegaKurs\node_modules\express\lib\router\layer.js:95:5)
at next (C:\Users\JakubKrol\Desktop\TESTY_yno\MegaKurs\node_modules\express\lib\router\route.js:137:13)
at Route.dispatch (C:\Users\JakubKrol\Desktop\TESTY_yno\MegaKurs\node_modules\express\lib\router\route.js:112:3)
at Layer.handle [as handle_request] (C:\Users\JakubKrol\Desktop\TESTY_yno\MegaKurs\node_modules\express\lib\router\layer.js:95:5)
at C:\Users\JakubKrol\Desktop\TESTY_yno\MegaKurs\node_modules\express\lib\router\index.js:281:22
at Function.process_params (C:\Users\JakubKrol\Desktop\TESTY_yno\MegaKurs\node_modules\express\lib\router\index.js:335:12)
at next (C:\Users\JakubKrol\Desktop\TESTY_yno\MegaKurs\node_modules\express\lib\router\index.js:275:10)
at expressInit (C:\Users\JakubKrol\Desktop\TESTY_yno\MegaKurs\node_modules\express\lib\middleware\init.js:40:5)
```

Przesyłanie plików - opcje.

Bardziej zaawansowana forma pozwala nam jako drugi parametr podać obiekt opcji.

```
app.get('/', (req : Request<P, ResBody, ReqBody, ReqQuery, Locals> , res : Response<ResBody, Locals> ) => {
  res.sendFile( path: 'nazwaPliku', options: {
    root: |
  });
});
```

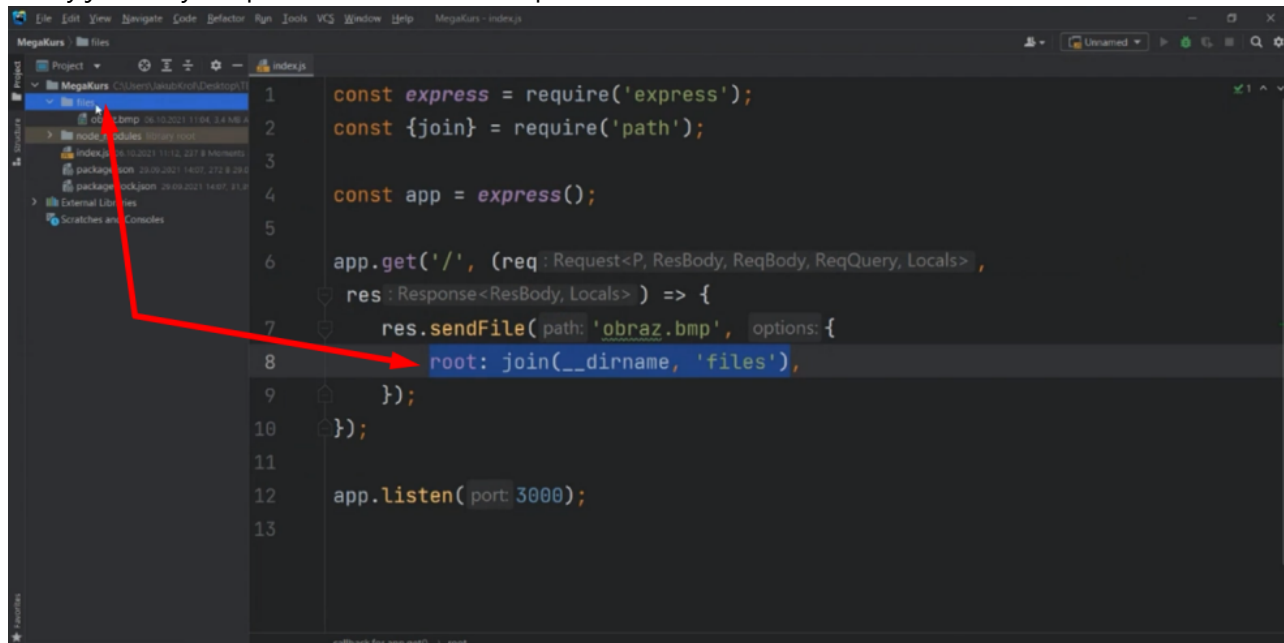
Zobaczmy na co pozwala:

- root
- lastModified

13:00

Zapisz kopię w Evernote

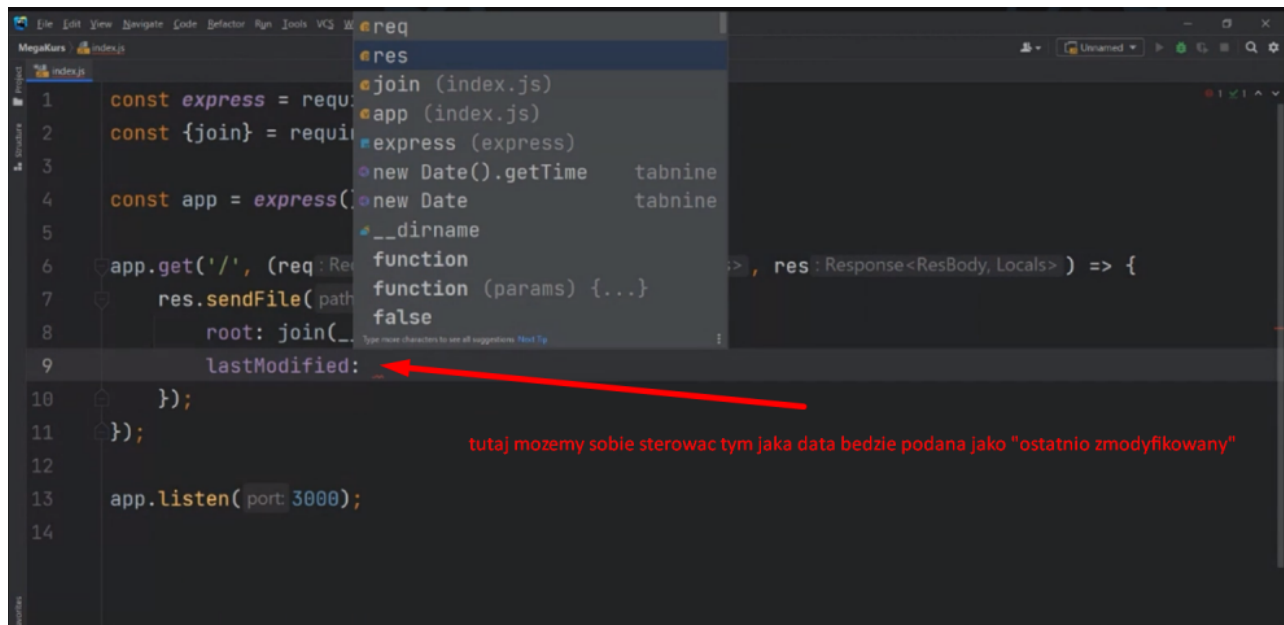
Do przesyłania plików najlepiej utworzyć sobie folder np. public albo files (jak to w przykładzie), bo wtedy jesteśmy bezpieczni ze nam nikt nie pobierze kodu.



```
1  const express = require('express');
2  const {join} = require('path');
3
4  const app = express();
5
6  app.get('/', (req: Request<P, ResBody, ReqBody, ReqQuery, Locals>,
7    res: Response<ResBody, Locals>) => {
8    res.sendFile(path: 'obraz.bmp', options: {
9      root: join(__dirname, 'files'),
10    });
11  });
12
13  app.listen(port: 3000);
```

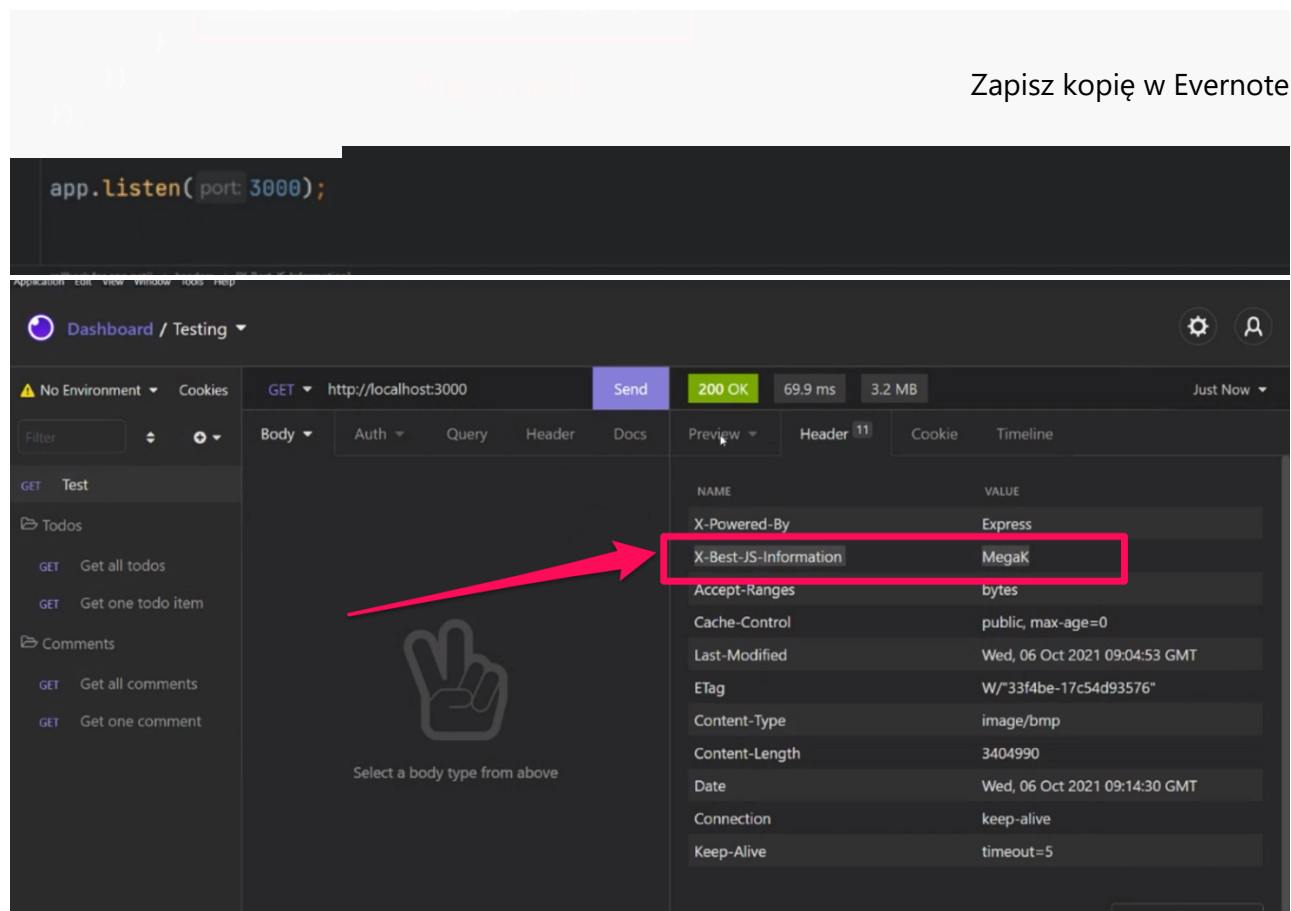
Ad 2.

lastModified



```
1  const express = require('express');
2  const {join} = require('path');
3
4  const app = express();
5
6  app.get('/', (req: Request<P, ResBody, ReqBody, ReqQuery, Locals>,
7    res: Response<ResBody, Locals>) => {
8    res.sendFile(path: 'obraz.bmp', options: {
9      lastModified: new Date().getTime(),
10    });
11  });
12
13  app.listen(port: 3000);
```

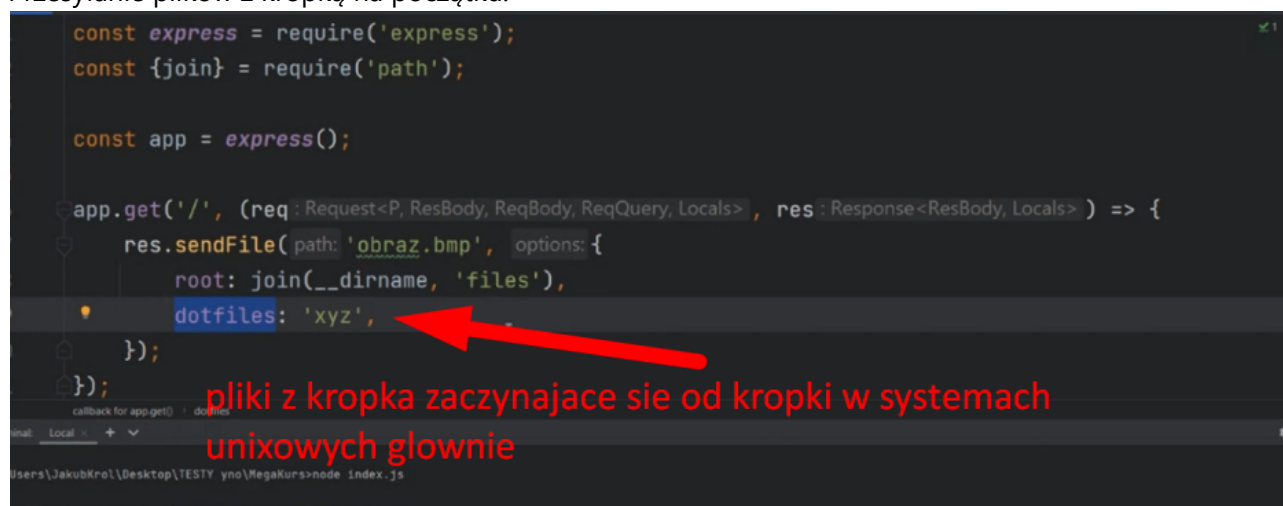
tutaj możemy sobie sterować tym jaka data będzie podana jako "ostatnio zmodyfikowany"



Ad 4.

dotfiles

Przesyłanie plików z kropką na początku.



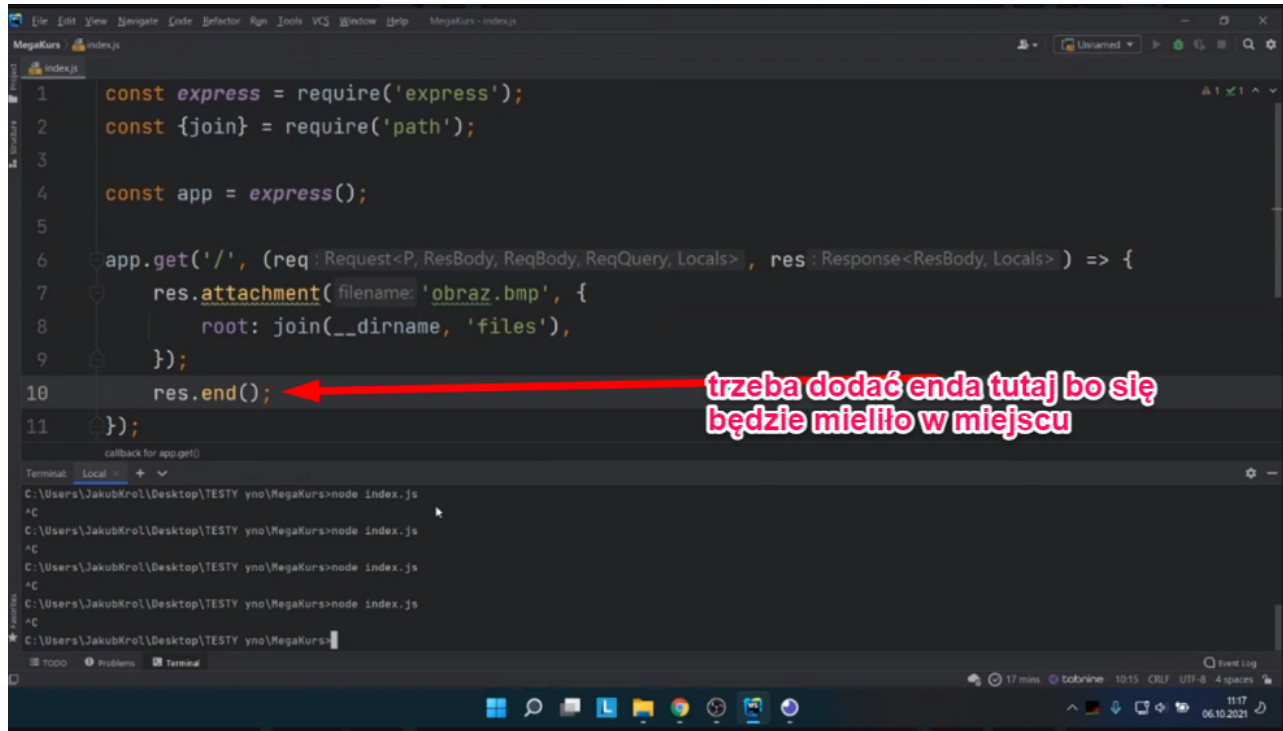
kolejną metodą do plików jest

```
res.attachment();
```

Zapisz kopię w Evernote

kiedy podamy jako argument nazwę pliku - zostanie on przesłany jako załącznik- czyli klient, np przeglądarka - zapyta o zapis tego pliku, zamiast go wyświetlać.

Express.js w tym przypadku nie zakończy automatycznie połączenia, należy pamiętać o dodaniu **res.end()**.



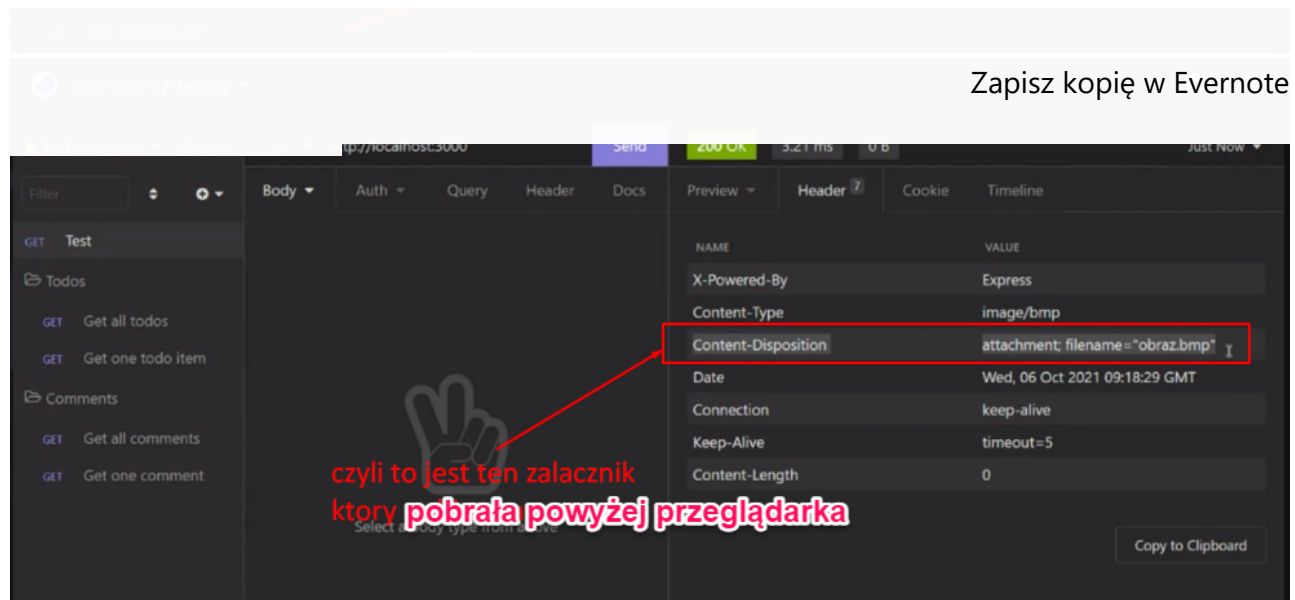
```
1  const express = require('express');
2  const {join} = require('path');
3
4  const app = express();
5
6  app.get('/', (req : Request<P, ResBody, ReqBody, ReqQuery, Locals> , res : Response<ResBody, Locals> ) => {
7    res.attachment(filename: 'obraz.bmp', {
8      root: join(__dirname, 'files'),
9    });
10   res.end();
11 });
```

trzeba dodać enda tutaj bo się będzie miało w miejscu

Terminal: Local

```
C:\Users\JakubKrol\Desktop\TESTY yno\MegaKurs>node index.js
^C
C:\Users\JakubKrol\Desktop\TESTY yno\MegaKurs>node index.js
^C
C:\Users\JakubKrol\Desktop\TESTY yno\MegaKurs>node index.js
^C
C:\Users\JakubKrol\Desktop\TESTY yno\MegaKurs>node index.js
^C
C:\Users\JakubKrol\Desktop\TESTY yno\MegaKurs>
```

Zapisz kopię w Evernote

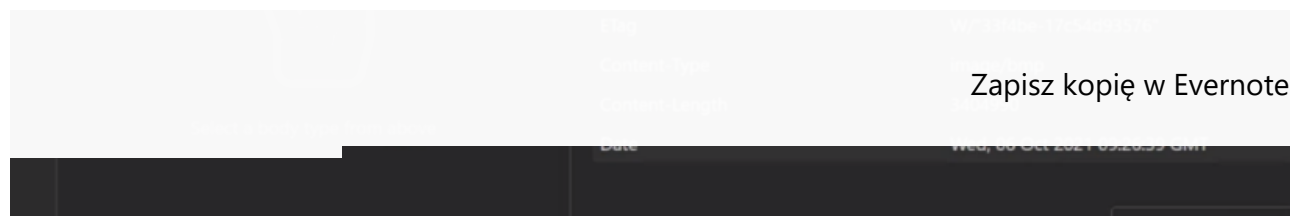


czyli to jest ten załącznik
który pobrała powyżej przeglądarka

NAME	VALUE
X-Powered-By	Express
Content-Type	image/bmp
Content-Disposition	attachment; filename="obraz.bmp"
Date	Wed, 06 Oct 2021 09:18:29 GMT
Connection	keep-alive
Keep-Alive	timeout=5
Content-Length	0

Copy to Clipboard

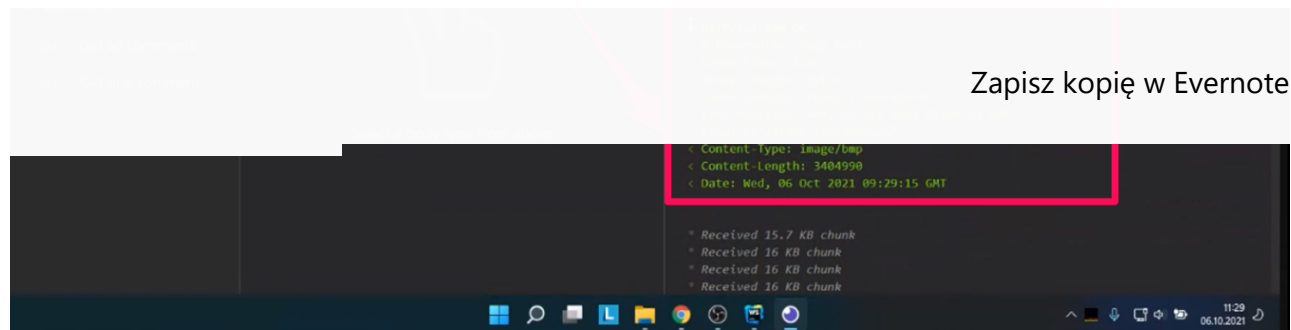
res.set();

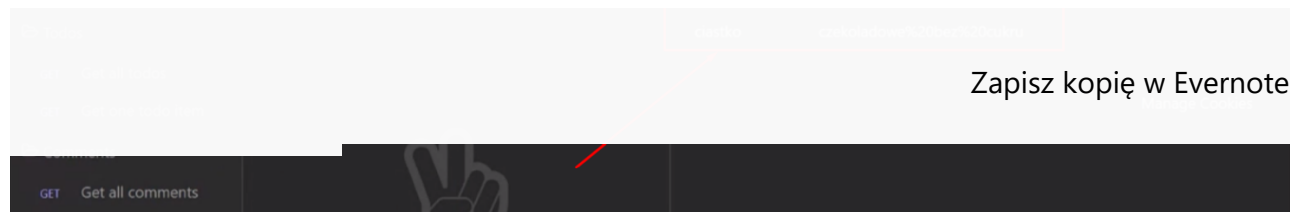


res.headerSent();

Zawiera informację czy nagłówki zostały już przesłane.

Trzeba pamiętać że najpierw ustawiać nagłówki, a potem treść odpowiedzi bo nie zadziała.





Zapisz kopię w Evernote

Zapisz kopię w Evernote