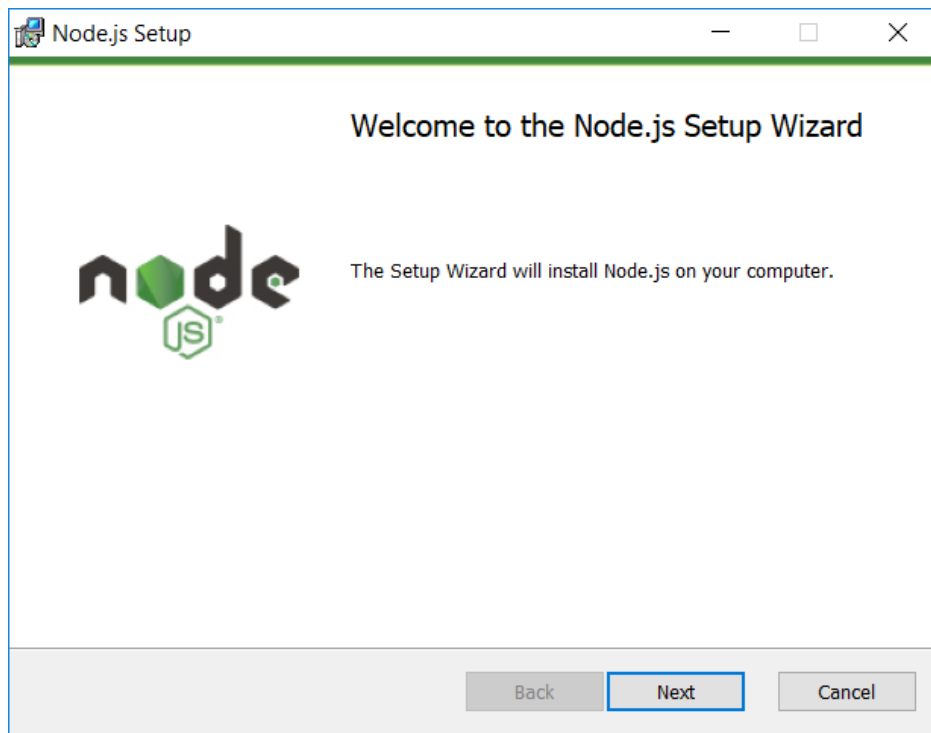


PRZYGOTOWANIE NODE.JS

INSTALACJA NODE.JS Z UPRAWNIENIAMI ADMINISTRATORA

1. Pobierz odpowiednią wersję instalatora ze strony <https://nodejs.org/en/download/>.
2. Uruchom go i przejdź standardową procedurę instalacyjną wybierając domyślne ustawienia.



3. Wyraż zgodę na wprowadzenie zmian na urządzeniu.

INSTALACJA NODE.JS BEZ UPRAWNIENI ADMINISTRATORA

1. Pobierz odpowiedni wersję **binary** ze strony <https://nodejs.org/en/download/>.
2. Rozpakuj do folderu, z którego będziesz go używać.
3. Ze względu na brak możliwości ustawienia w sposób permanentny ścieżki do folderu bez uprawnień administratora konieczne będzie przed każdym użyciem Node.js i Npm skonfigurowanie ścieżki na poziomie powłoki. W tym celu uruchom *cmd* i skonfiguruj zmienną *PATH* wskazując na lokalację pliku *node.exe*:
 - Np.: `PATH=%PATH%;C:\node\node-v8.12.0-win-x64`

SPRAWDZENIE INSTALACJI NODE.JS

1. Stwórz katalog projektu testowego (np.: test).
2. Przejdź do niego i spróbuj wywołać polecenia:
 - `node -v`
 - `npm -v`
3. Powinieneś zobaczyć informację o wersji:

```
C:\node\test
node -v
v8.12.0

C:\node\test
npm -v
6.1.0
```

4. W ulubionym edytorze stwórz teraz w katalogu projektu krótki program:

```
console.log("Hello Node!")
```

5. Następnie uruchom go z użyciem node. Powinieneś zobaczyć wynik.

```
C:\node\test
node helloworld.js
Hello Node!
```

6. Zainstaluj Express.js wydając polecenie `npm install express`:

```
C:\node\test
npm install express
npm WARN saveError ENOENT: no such file or directory, open 'C:\node\test\package.json'
npm notice created a lockfile as package-lock.json. You should commit this file.
npm WARN enoent ENOENT: no such file or directory, open 'C:\node\test\package.json'
npm WARN test No description
npm WARN test No repository field.
npm WARN test No README data
npm WARN test No license field.

+ express@4.16.4
added 48 packages from 36 contributors and audited 121 packages in 7.577s
found 0 vulnerabilities
```

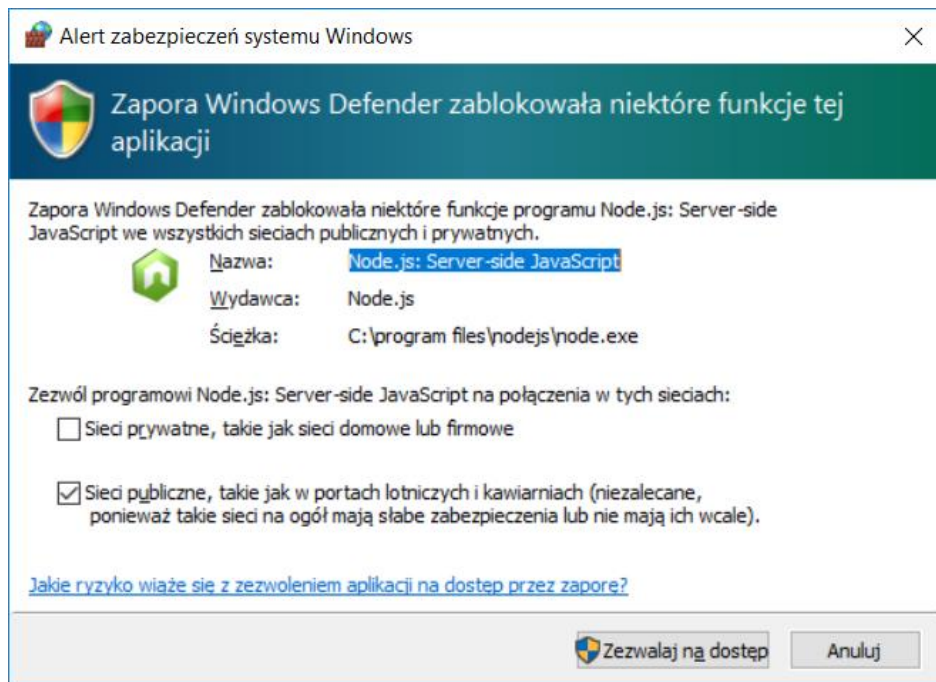
7. Zmodyfikuj aplikację testową do postaci:

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

app.get('/', (req, res) => res.send('Hello Node and Express!'))

app.listen(port, () => console.log(`Aplikacja uruchomiona na
porcie ${port}!`))
```

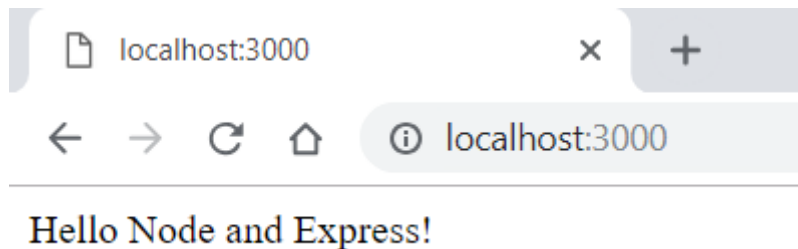
8. I uruchom ponownie. Przy uruchomieniu pojawić się może pytanie o zezwolenie na dostęp do połączeń. Pracując z lokalnej maszyny można wybrać anuluj.



9. W wierszu poleceń pojawi się po uruchomieniu komunikat:

```
C:\node\test
node helloworld.js
Aplikacja uruchomiona na porcie 3000!
```

10. Uruchom przeglądarkę i wpisz <http://localhost:3000>, gdzie 3000 to wskazany wcześniej numer portu. Powinna być widoczny wynik działania aplikacji webowej:



11. Jeżeli wszystko zadziałało oznacza to, że środowisko jest gotowe do pracy.