# Node.js

Express - it's like a fast-track ticket

christoffer.wallenberg@zocom.se



# Vad är express?

Do you mind if i give you the pitch?



## The pitch!

Fast, unopinionated, minimalist web framework for Node.js

## Vad betyder det?

- Ett minimalistisk lager ovanpå Node.js som ger ett antal funktioner
- Förenklar en del kring både att bygga en webbserver och ett API
- Väldigt populärt att använda när man bygger applikationer med Node.js
- Alla funktioner etc i Node.js går fortfarande att använda

## En enkel webbserver i Express

```
const express = require('express');
             const app = express();
             const port = process.env.PORT || 8000;
             app.get('/', (request, response) => {
Vad händer här?
                response.send('Hello World!');
            });
             app.listen(port, () => {
                console.log('Starting server');
            });
```

# Routing

# Routing innebär att vi kan tala om vad som ska hända vid olika url:er

Det har vi dels redan arbetat med

#### **Routing i express**

- method den HTTP-metod som används ex. GET, POST etc.
- path den url som efterfrågas
- **callback** den funktion som anropas när ett request matchar

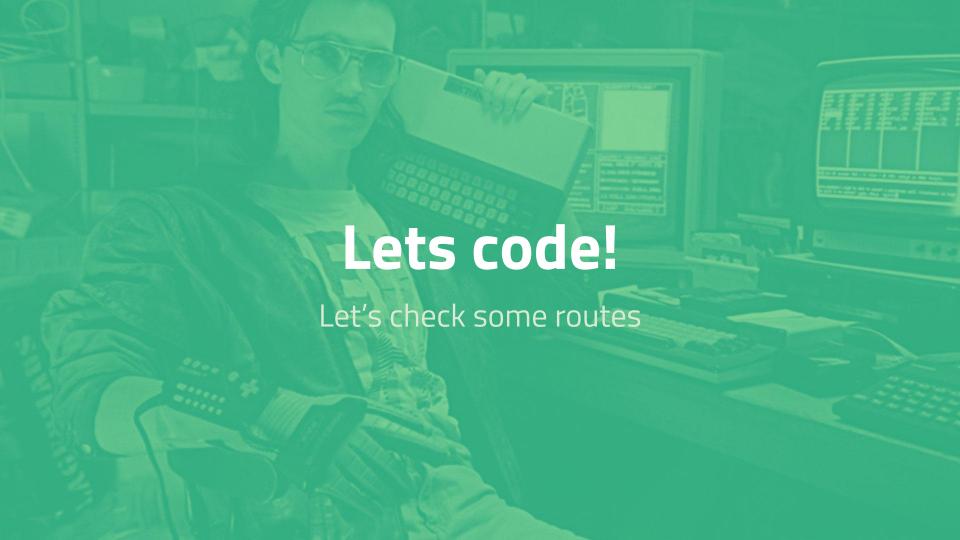
```
const express = require('express');
const app = express();
```

app.method(path, callback);

#### **Routing i express**

- Ifall man skriver in <u>http://localhost:8000/form</u> i webbläsaren, vad skrivs ut?
- Ifall man skriver in <u>http://localhost:8000/about</u> i webbläsaren, vad skrivs ut?
- Ifall man skriver in
   http://localhost:8000/about/test
   i webbläsaren, vad skrivs ut?
- Diskutera två och två i fem minuter

```
app.post('/form', (req, res) => {
   res.send('Post form');
});
app.get('/form', (req, res) => {
   res.send('Get form')
});
app.all('/about*', (req, res) => {
   res.send('About');
});
```



# Request / Response i express

- Request-objektet har följande egenskaper:
  - o url
  - o method
  - o params
  - query
- **Response-objektet** har följande metoder:
  - o send()
  - o sendFile()

```
app.get('/, (req, res) => {
    res.send('Hello World');
});
```

### Request - params

```
app.get('/plays/:play', (req, res) => {
    res.send(req.params.play); //Macbeth skrivs ut
});
```

Ifall man går in på url:en http://localhost:8000/plays/Macbeth

## Request - query

```
app.get('/plays', (req, res) => {
    res.send(req.query.play); //The Tempest skrivs ut
});
```

Ifall man går in på url:en http://localhost:8000/plays?play= The%20Tempest

## Statiska filer

Prepare to be blown away!

# Statiska filer kan vi skicka tillbaka direkt när klienten anropar dessa

Då behöver vi inte skapa en route för varje fil

#### Mappstruktur

- server.js vår
   Node.js-applikation
- public mapp med alla statiska filer f\u00f6r klienten

```
express-example - mapp
server.js - vår Node.js applikation
public - mapp med alla våra statiska filer
index.html
css - mapp
style.css
js - mapp
index.js - vår javascript i front-end
```

#### Statiska filer

```
const express = require('express');
const app = express();
const port = process.env.PORT || 8000;
app.use(express.static('public'));
```

