

Creating Websites in NodeJS



Dariusz Krzeminski
sevisilex@gmail.com

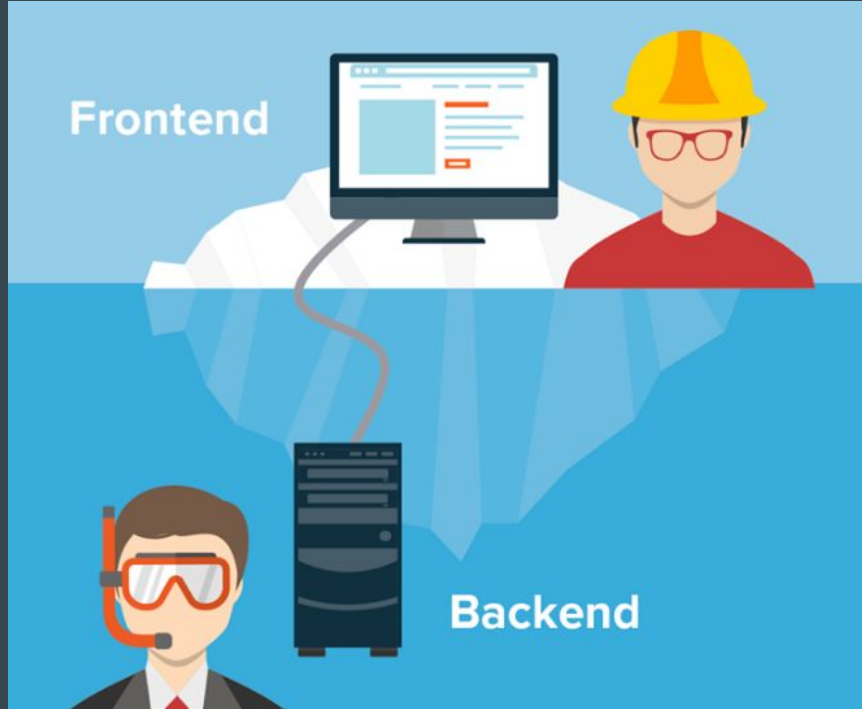
twitter.com/sevisilex
linkedin.com/in/dariuszkrzeminski78



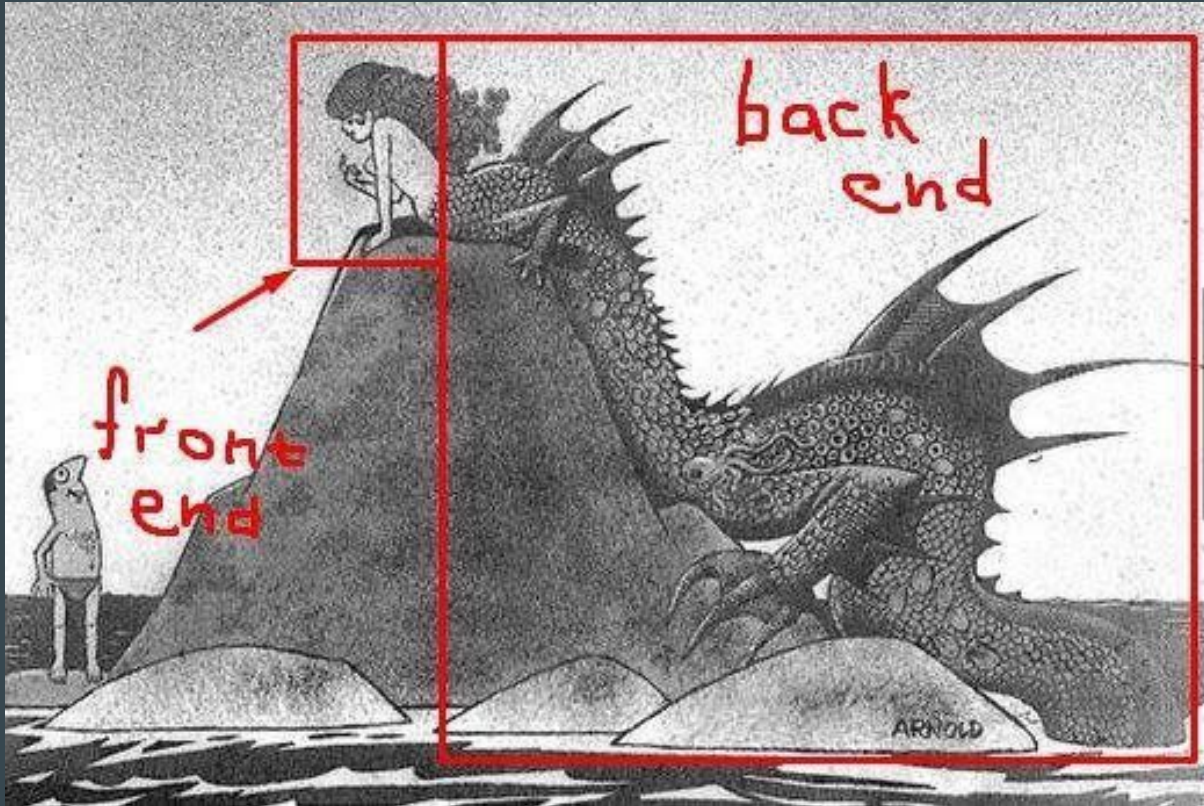
1.1 What is Nodejs

- NodeJS is an open source, cross platform runtime environment for server side and networking application.
- It is written in JavaScript and can run on Linux, Mac, Windows, FreeBSD.
- Uses an event-driven, non-blocking I/O model for building scalable network applications
- It used Google JavaScript V8 Engine to Execute Code.
- It is used by Groupon, LinkedIn, Microsoft, Yahoo, Paypal, ...
- It was created by Ryan Dahl in 2009.

1.2 What different between Frontend and Backend



1.2 What different between **Frontend** and **Backend**

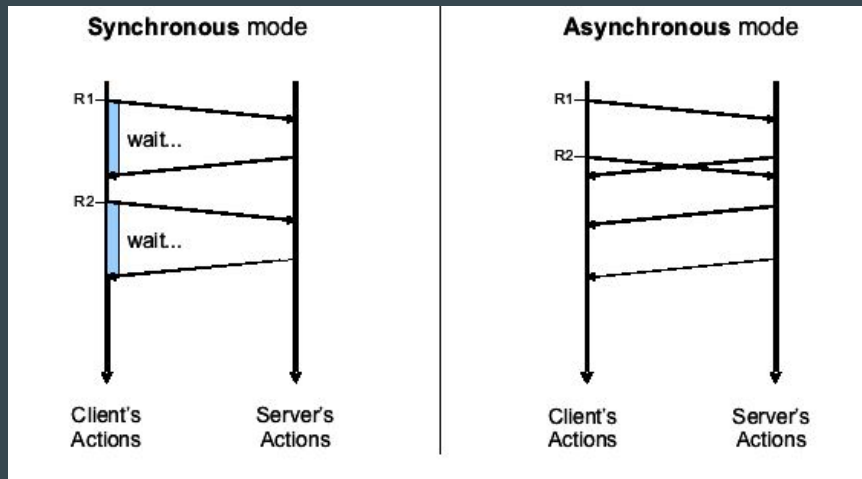
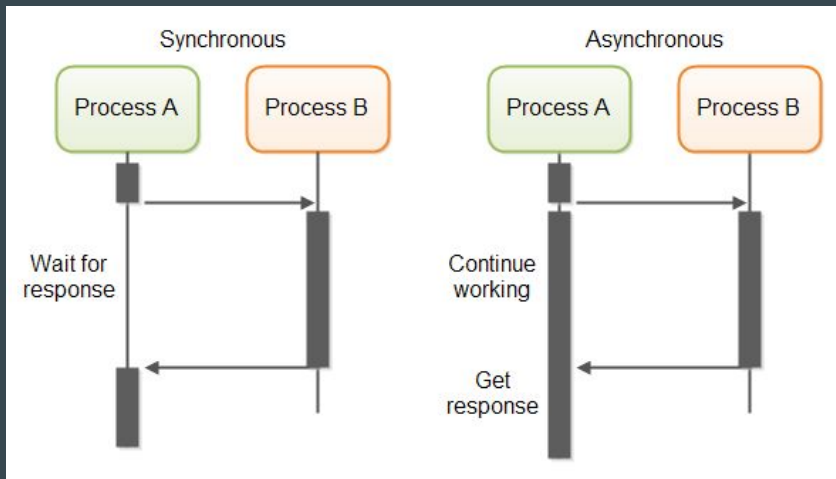


1.3 What is JavaScript

- Javascript is basic code.
- It is dynamic programming language.
- It is client side script to interact with user.
- It most commonly used as a part of web browser.



1.4 What is different **Synchronous** and **Asynchronous**



// Synchronous

```
<?php  
echo "hello";  
sleep(2);  
echo "world";  
?>
```

vs

// Asynchronous

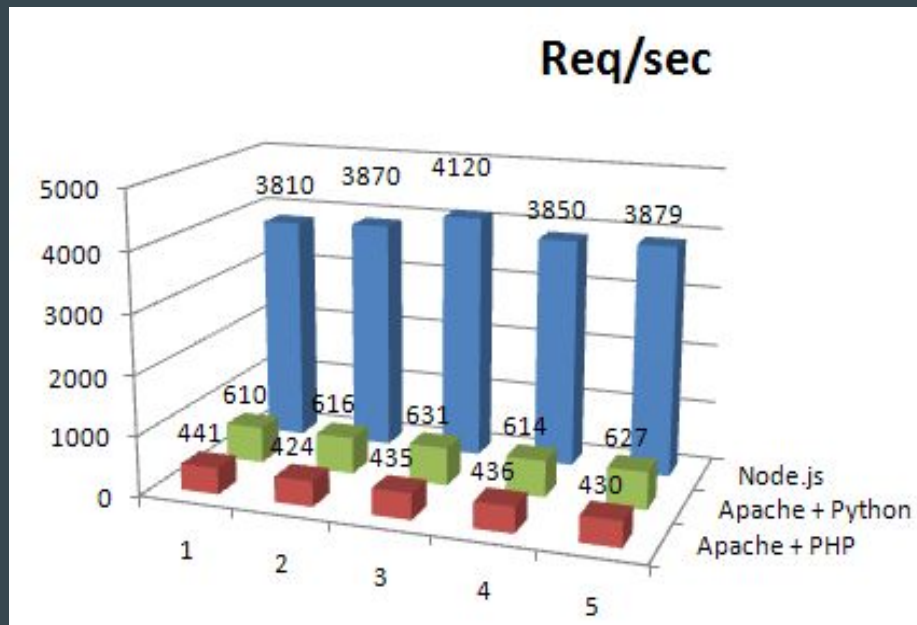
```
setTimeout(function() {  
    console.log('world');  
}, 2000);  
console.log('hello');
```

1.5 Benchmark - PHP vs NodeJS

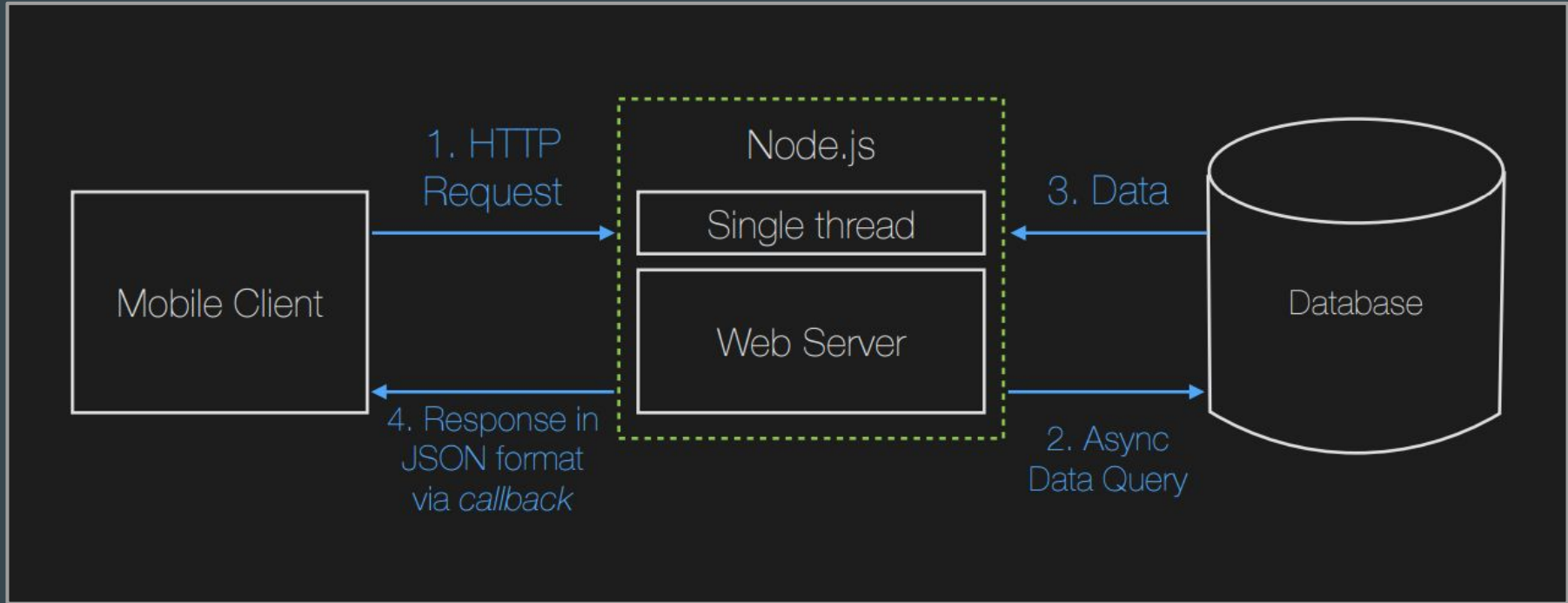
Node.js serves can nearly 3,900 requests/second.

for comparison:

- PHP 430 requests/second.
- Python 620 requests/second.



1.6 How it works - accessing data from a database



1.7 How to install NodeJS on **Linux** - <https://nodejs.org/en/>

```
cd /usr/local
wget https://nodejs.org/dist/v4.6.1/node-v4.6.1-linux-x64.tar.gz
sudo tar --strip-components 1 -xzf ./node-v4.6.1-linux-x64.tar.gz
```

```
darek@deafit:~$ node -v
v4.6.1
```

Built-in package manager contributes to the modularity of Node

darek@deafit:~\$ npm install fs		[filesystem]
darek@deafit:~\$ npm install express		[web-server]
darek@deafit:~\$ npm install ejs		[template]
darek@deafit:~\$ npm install mysql		[mysql-client]
darek@deafit:~\$ npm install body-parser		[post request]
darek@deafit:~\$ npm install method-override		[method request]

if you use Windows then use additional argument --no-bin-links

Here we go start!

...

Los geht's!

2.1 How to run a Nodejs - Calculator

```
darek@deafit:~$ node
```

```
> 1
```

```
1
```

```
> 2+3
```

```
5
```

```
> 4*3
```

```
12
```

```
> var a=3, b=7
```

```
> a*b
```

```
21
```

```
> var add = function(a, b) { return a+b; }
```

```
> add(3,4)
```

```
7
```

```
> add(a,b)
```

```
10
```

2.2 How print - Hello World

```
> 'DeafIT'
'DeafIT'
> var a = 'Hello', b = 'World'
> a
'Hello'
> a + b
'HelloWorld'
> a + ' ' + b + '!'
'Hello World!'
> _
```

2.3 Why use **setTimeout(...)** than **sleep(..)**

```
darek@deafit:~$ cat 11-hello-world.js
// echo('Hello');
// sleep(2000);          // <- blocking
// echo('World');

setTimeout(function() { // <- not blocking
    console.log('World');
}, 2000);
console.log('Hello');
```

```
darek@deafit:~$ node 11-hello-world.js
Hello
World
darek@deafit:~$ _
```

2.4 What different between `setTimeout(...)` and `setInterval(...)`

```
darek@deafit:~$ cat 12-hello-world2.js
```

```
setInterval(function() {  
    console.log('World');  
}, 2000);  
console.log('Hello');
```

```
darek@deafit:~$ node 12-hello-world2.js
```

```
Hello
```

```
World
```

```
World
```

```
World
```

```
...
```

```
^Cdarek@deafit:~$ _
```

2.5 What different `readFileSync(...)` and `readFile(...)`

```
darek@deafit:~$ cat 14-fileSystem_sync.js
```

```
// synchronous - blocking I/O
var fs = require('fs');
var text = fs.readFileSync('./readme.txt');
console.log(text);
```

```
darek@deafit:~$ cat 15-fileSystem_nosync.js
```

```
// asynchronous - non-blocking I/O
var fs = require('fs');
fs.readFile('./readme.txt', function(err, text) {
    console.log(text);
});
```

Display on the website

...

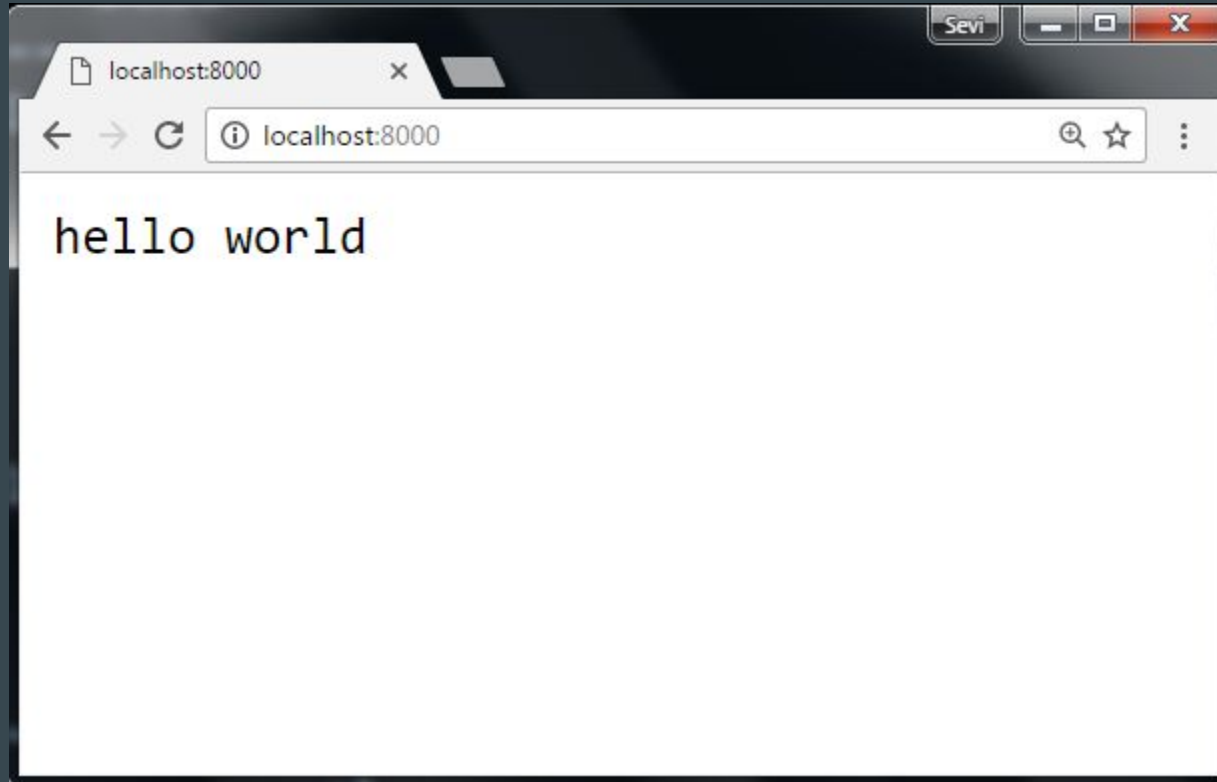
Anzeige auf der Website

3.1 How to display on the Websites - Hello World

```
darek@deafit:~$ cat 21-web-server.js
var http = require('http');
var s = http.createServer(function(req, res) {
  res.writeHead(200, { 'content-type': 'text/plain' });
  res.end('hello world\n');
});
s.listen(8000);

darek@deafit:~$ node 21-web-server.js
```

3.1 How to display on the Websites - Hello World

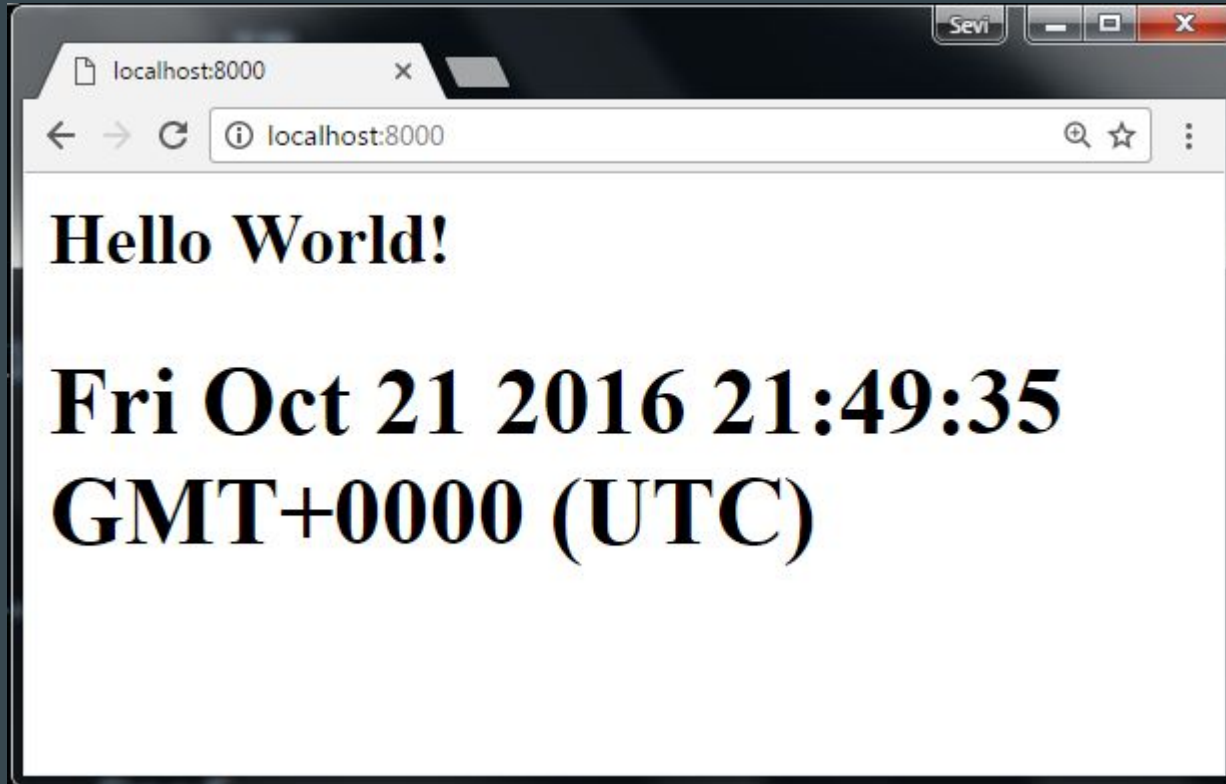


3.2 How to display **Html** on the Websites

```
darek@deafit:~$ cat 22-web-server2.js
var http = require('http');
var s = http.createServer(function(req, res) {
  var date = new Date();
  res.writeHead(200, { 'content-type': 'text/html' });
  res.end('<html><body>' +
    '<h2>Hello World!</h2>' +
    '<h1>' + date + '</h1>' +
    '</body></html>');
});
s.listen(8000);

darek@deafit:~$ node 22-web-server2.js
...
darek@deafit:~$ ab -n 10000 http://127.0.0.1:8000/
```

3.2 How to display **Html** on the Websites



4.1 How to display a template Ejs

```
darek@deafit:~$ cat 41-web-ejs.js
var express = require('express'),
    ejs = require('ejs'),
var app = express();

app.set('view engine', 'ejs');

app.get('/', function(req, res) {
    res.render('helloworld', { word: 'Hallo' });
});

app.listen(8000, function() {
    console.log('App listening on port 8000');
});
```

helloworld.ejs

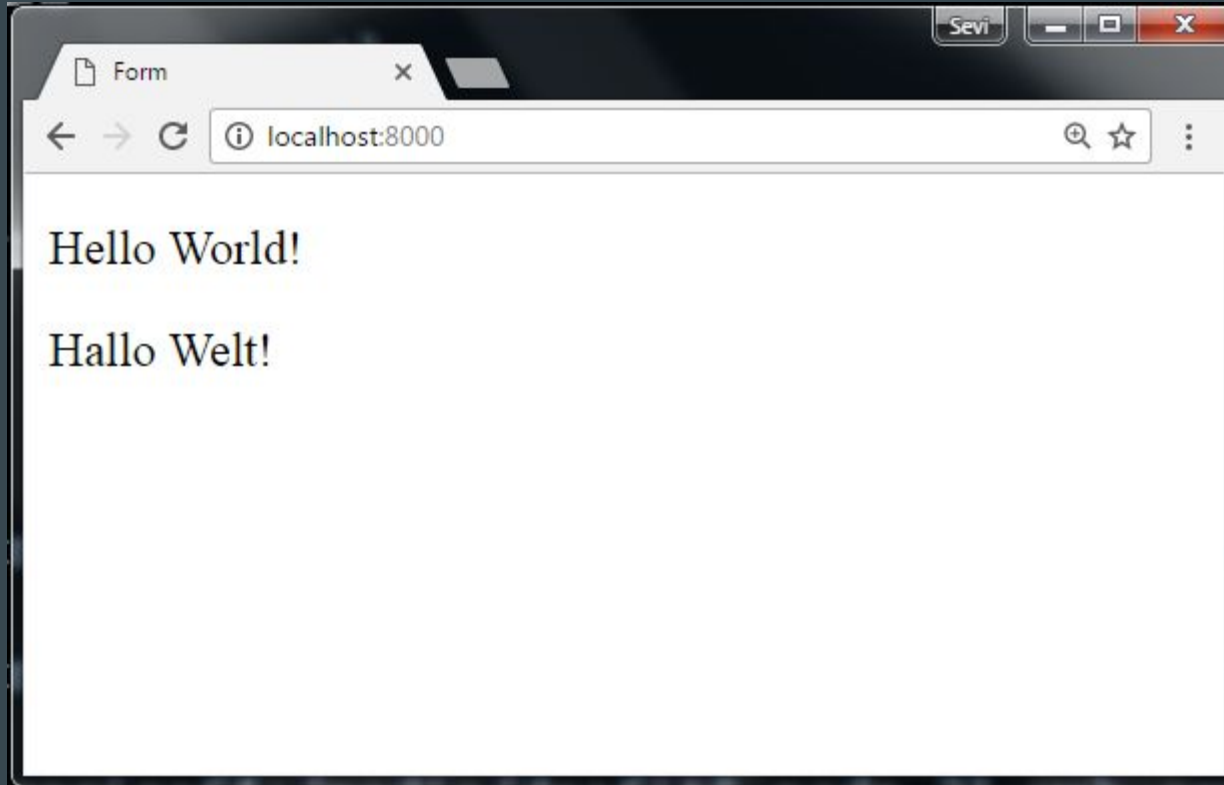
```
<!DOCTYPE html>
<html>
  <head>
    <title>Form</title>
  </head>
  <body>

    <p>Hello World!</p>

    <p><%= word %> Welt!</p>

  </body>
</html>
```

4.1 How to display a template Ejs



4.2 How to use the **route**

```
darek@deafit:~$ cat 31-web-expres.js
var express = require('express');
var app = express();

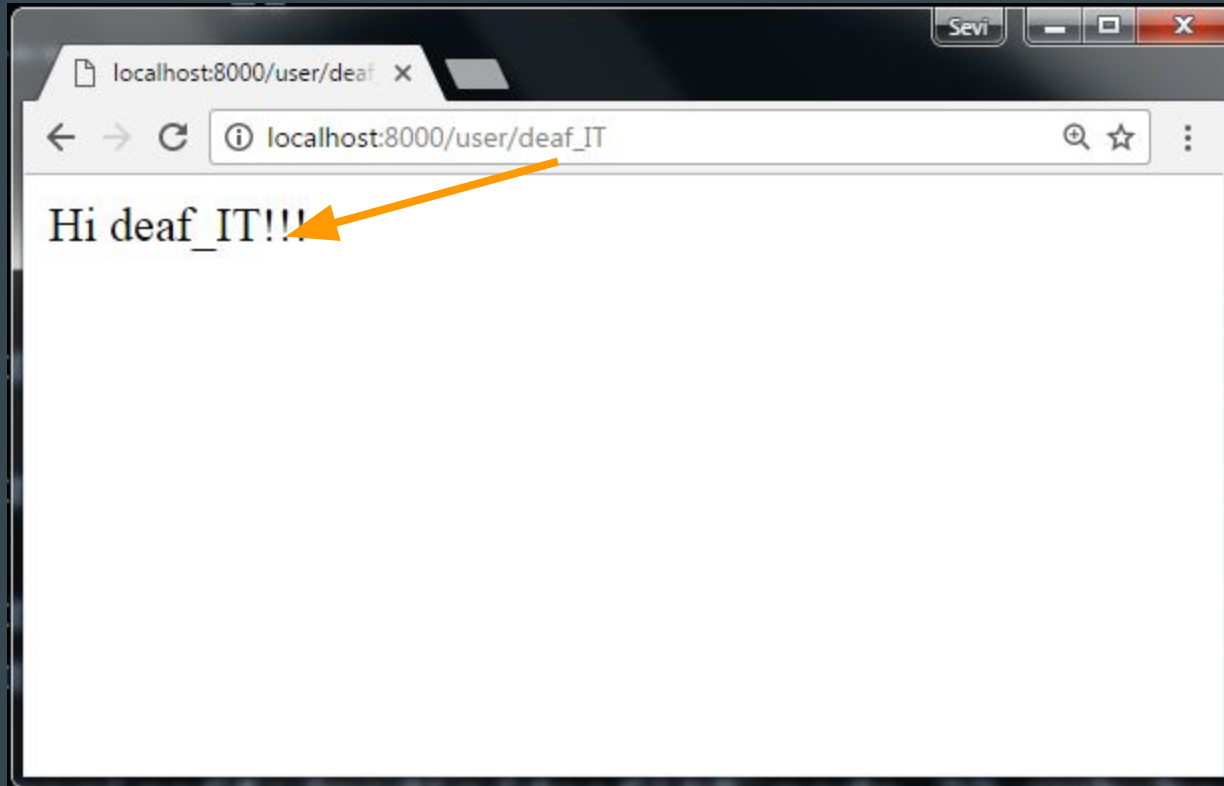
app.get('/', function(req, res) {
  res.send('Hello World!');
});
app.get('/home', function(req, res) {
  res.send('Home!');
});
app.get('/user/:user', function(req, res) {
  res.send('Hi ' + req.params.user + '!!!');
});
app.listen(8000, function() {
  console.log('App listening on port 8000');
});
```

```
http://local/
Hello World!
```

```
http://local/home
Home!!!
```

```
http://local/user/darek
Hi darek!!!
```

4.2 How to use the **route**



4.3 How to use the HTTP method for **CRUD**

CRUD is a short, mean CREATE, READ, UPDATE, DELETE

Nodejs can to use HTTP method, for example:

HTTP GET - read (for SQL: SELECT)

HTTP POST - create (for SQL: INSERT)

HTTP PUT - update (for SQL: UPDATE)

HTTP DELETE - delete (for SQL: DELETE)


But preferably using **AJAX** (Asynchronous JavaScript and XML) on CRUD.

```
darek@deafit:~$ cat 33-web-method.js
```

4.4 How to use the HTTP method for CRUD

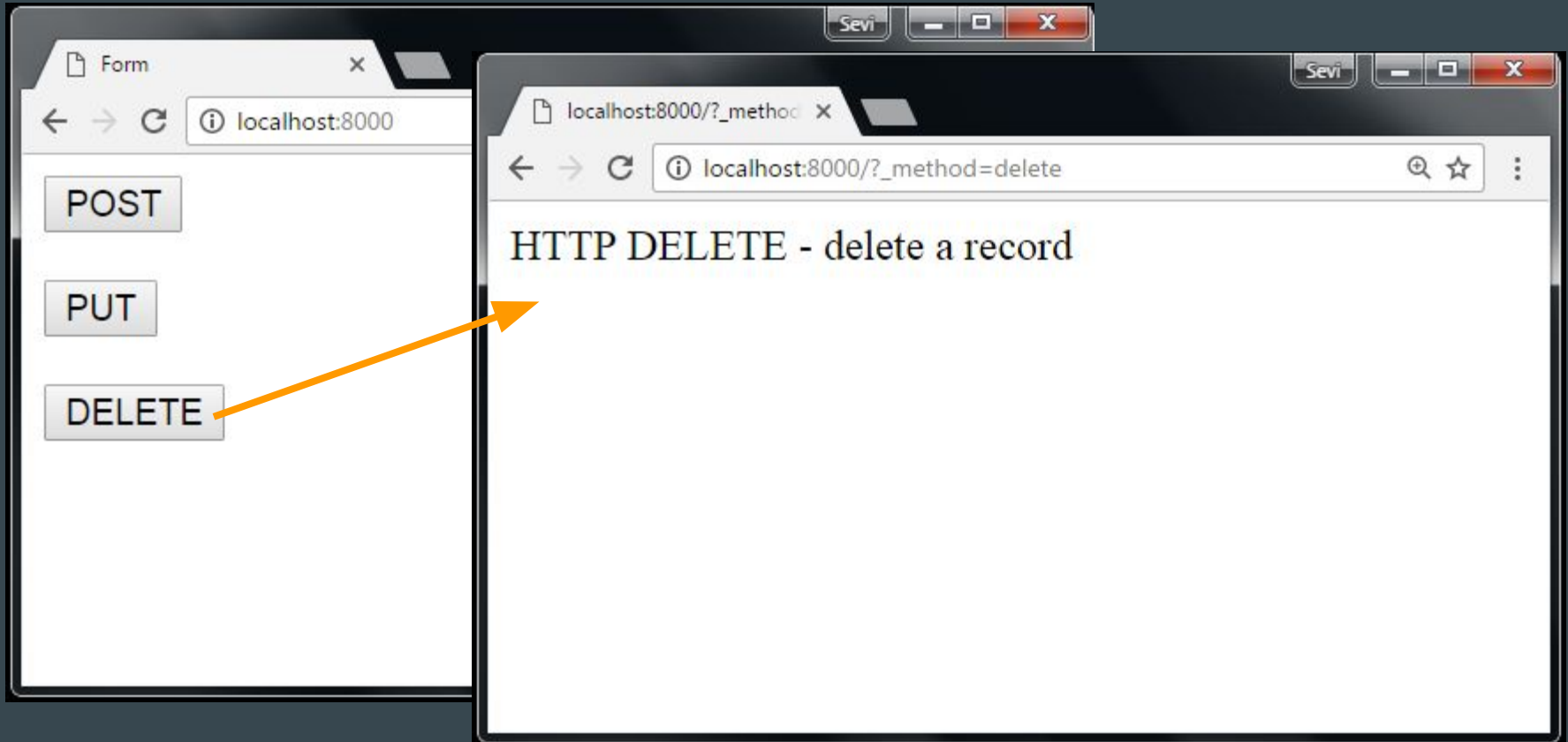
```
var express = require('express'),
    bodyParser = require('body-parser'),
    methodOverride = require('method-override'),
var app = express();
app.use(methodOverride('_method'));

app.get('/', function(req, res) {
  res.render('44-web-crud1');
});
app.post('/', function(req, res) {
  res.send('HTTP POST - create a new record');
});
app.put('/', function(req, res) {
  res.send('HTTP PUT - update a record');
});
app.delete('/', function(req, res) {
  res.send('HTTP DELETE - delete a record');
});
```



http://local/?_method=post
http://local/?_method=put
http://local/?_method=delete

4.4 How to use the HTTP method for CRUD



Connect to MySQL

...

Verbindung zu MySQL

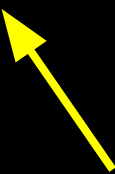
5.1 How to connect to MYSQL

```
darek@deafit:~$ cat 51-mysql.js
```

```
var mysql = require('mysql');
```

```
var pool = mysql.createPool({  
  connectionLimit : 10,  
  host: 'localhost',  
  user: 'root',  
  password: '',  
  database: 'deafit',  
});
```

```
pool.query('SELECT * FROM users', function(err, rows) {  
  for (var i = 0; i < rows.length; i++) {  
    console.log(rows[i].id, rows[i].firstname, rows[i].lastname);  
  }  
  pool.end();  
});
```




```
1 Darek Krzeminski  
2 Silvester Stalone  
3 Bill Gates
```

5.2 How to query in SQL (with escape string)

```
pool.query('SELECT * FROM users WHERE firstname = ?', ['darek'], function(err, rows) {
```

```
    -> SELECT * FROM users WHERE firstname = 'darek'
});
```



```
pool.query('SELECT * FROM users WHERE id IN (?)', [[2,3,4]], function(err, rows) {
```

```
    -> SELECT * FROM users WHERE id IN (2, 3, 4)
});
```



```
pool.query('SELECT * FROM users WHERE firstname LIKE ? OR id IN (?)',  
['%darek%'], [3, '4', 5], function(err, rows) {
```

```
    -> SELECT * FROM users WHERE firstname LIKE '%darek%' OR id IN (3, '4', 5)
});
```



5.3 How to query in SQL (with escape string)

```
var id = 3;  
pool.query('DELETE FROM users WHERE id = ?', [id], function(err) {...
```

```
DELETE FROM users WHERE id = 3
```



```
pool.query('INSERT INTO users (firstname, lastname, description) VALUES (?)',  
[['darek', 'krzeminski', 'nodejs']], function(err) {...
```

```
INSERT INTO users (firstname, lastname, description) VALUES ('darek',  
'krzeminski', 'nodejs')
```

```
pool.query('UPDATE users SET ? WHERE firstname = ?', [{lastname: 'Gates',  
description: 'Microsoft'}, 'Bill'], function(err) {....
```

```
UPDATE users SET lastname='Gates', description='Microsoft' WHERE  
firstname='Bill'
```

5.3 What is SQL incjetion

```
var firstname = 'darek';  
pool.query('DELETE FROM users WHERE firstname = ' + firstname, function ...
```



```
DELETE FROM users WHERE id = darek  <- Fail
```

```
pool.query('DELETE FROM users WHERE firstname = ?', [firstname], function ...
```

OR

```
pool.query('DELETE FROM users WHERE firstname = '+  
           pool.escape(firstname), function ...
```

```
DELETE FROM users WHERE id = 'darek'  <- Correct
```

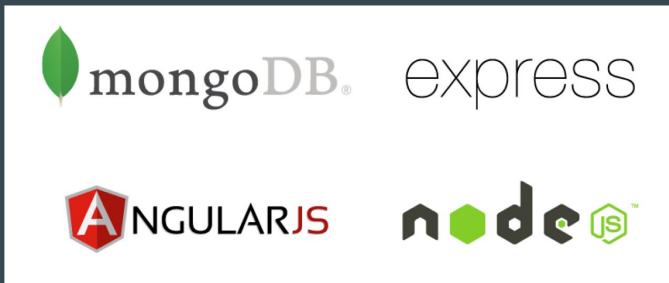

Framework?

...

6.1 Framework **Mean.io**

MEAN is a free and open-source JavaScript software stack for building dynamic web sites and web applications.

The **MEAN** stack makes use of **MongoDB**, **Express.js**, **Angular**, and **Node.js**. Because all components of the **MEAN** stack support programs are written in **JavaScript**, MEAN applications can be written in one language for both **server-side** and **client-side** execution environments.



Questions?

...

Frage? ;-)

Thank you for seeing us :D

...

Danke für Ihre Aufmerksamkeit

Dariusz Krzeminski
sevisilex@gmail.com

twitter.com/sevisilex
linkedin.com/in/dariuszkrzeminski78

References:

https://youtu.be/jo_B4LTHi3I

<https://nodejs.org/en/>

<https://expressjs.com/>

<https://npmjs.com/>

<https://ejs.co/>

<https://github.com/mysqljs/mysql>

<http://mean.io/>

...

Boring? You can play in Javascript ;-)

<http://play.elevatorsaga.com/>