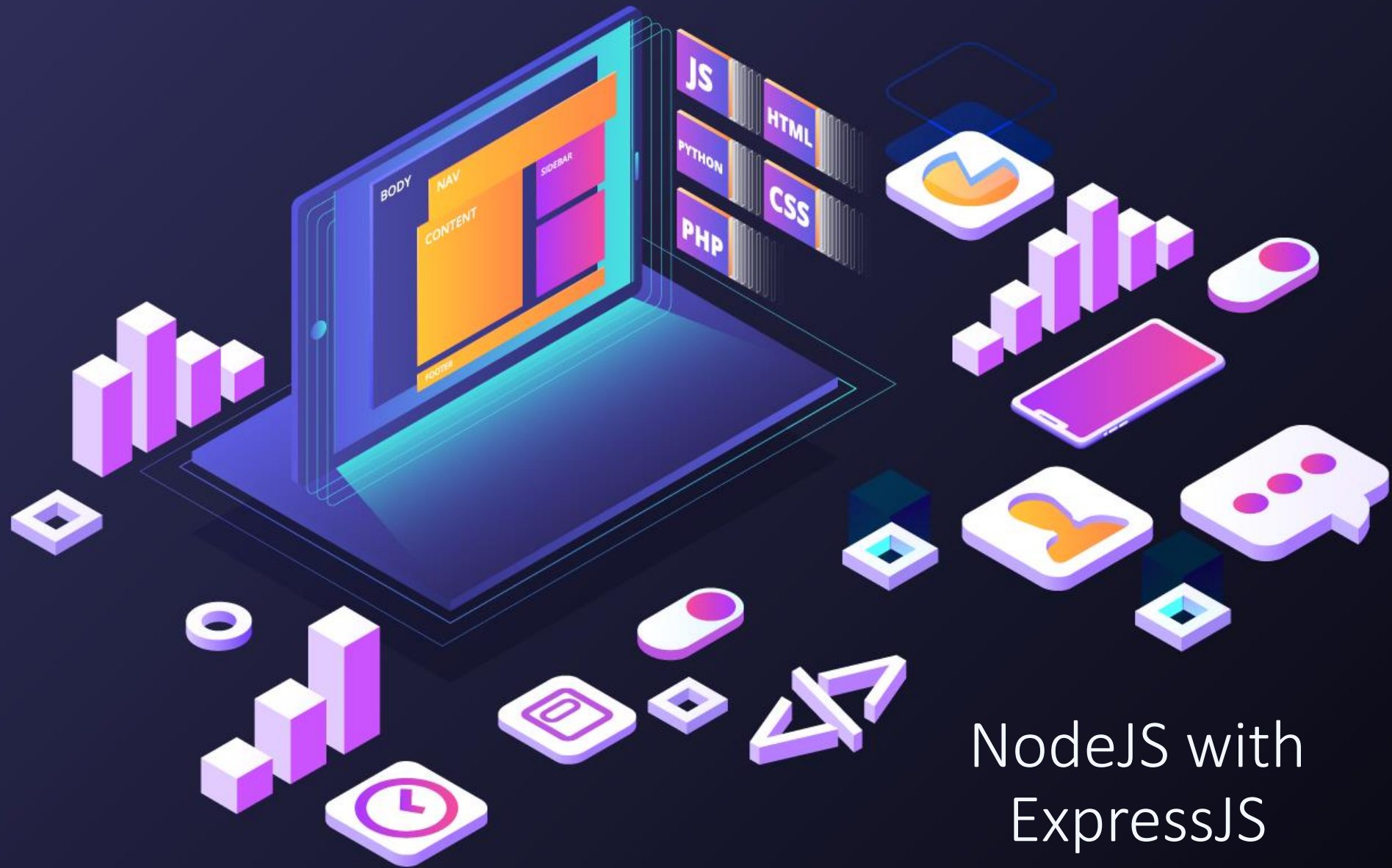


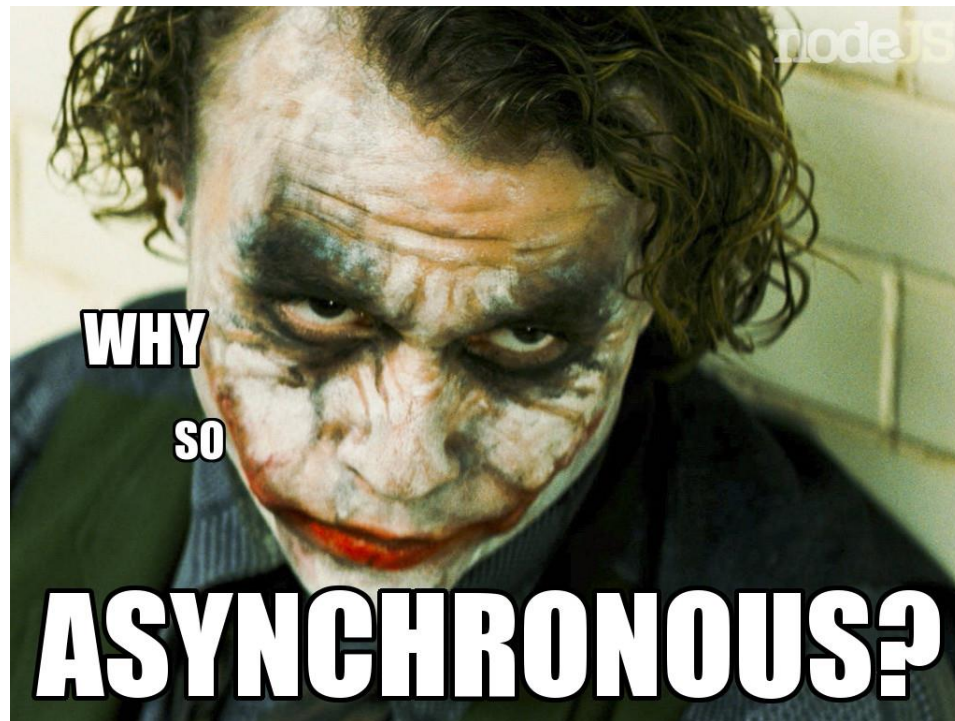


NodeJS with
ExpressJS
by Vlad Ungureanu

Agenda

NodeJS
and Express JS

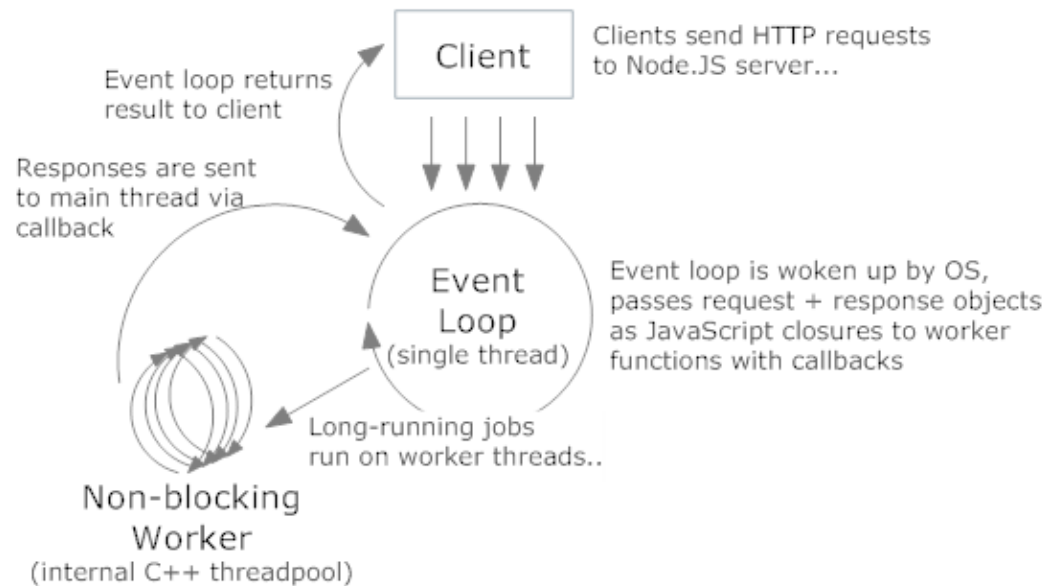
- What is Node JS
- Processing Model
- Express JS
- Express JS Server
- Rest Web Services
- DB Connectivity
- Handlebars
- Callback Hell
- Overview



What is Node JS ?

- Based on Chrome JavaScriptV8 engine
- Written in C
- Fast, non blocking
- Event driven server/framework

Node.JS Processing Model



- Minimal and flexible Node.JS web application framework that provides a robust set of features for web and mobile applications.
- A myriad of HTTP utility methods and middleware at your disposal, creating a robust API is quick and easy.
- Express provides a thin layer of fundamental web application features, without obscuring Node features

```
var express = require('express');
var session = require('express-session');
var bodyParser = require('body-parser');
var handlebars = require('express-handlebars').create({defaultLayout:'main'});
var app = express();
app.engine('handlebars', handlebars.engine);
app.set('view engine', 'handlebars');
app.use(express.static('public'));
app.use(bodyParser.json());
app.use(bodyParser.urlencoded({extended: true}));
var server = require('http').Server(app);
server.listen(3000);
require('./app/config/pg_datastore.js')(app);
var usersRepository = require('./app/repository/user-repository.js');
require('./app/controllers/user_skills.js')(app, userSkillsRepository);
```

```
app.get('/',function(req,res, next){  
  sess=req.session;  
  if(sess.userId) {  
    res.redirect('/home');  
  } else {  
    res.render('authentication');  
  }  
});
```

```
app.post('/login',function(req,res){  
  sess=req.session;  
  res.set('Content-Type', 'application/json');  
  userRepository.searchForUser(app, req.body.email,req.body.pass, res, sess);  
});
```



```
app.pg = require('pg');
app.connectionString = "postgres://nodejs:nodejs@localhost/techlounge";

exports.getCategories = function(app,res){
  var results = [];
  app.pg.connect(app.connectionString, function(err, client, done) {
    var query = client.query("SELECT * FROM tl_categories;");
    query.on('row', function(row) {
      results.push(row);
    });
    query.on('end', function() {
      client.end();
      res.send(results);
    });
  });
}
```

```
<!doctype html>
<html>
<head>
  <title>TechLounge: Learn, track, share</title>
</head>
<body>
  {{> menu}}
  <div class="header_container">
    {{{body}}}
  </div>
  <div class="info_container"></div>
  <div class="footer_container"></div>
</body>
</html>
```

In NodeJS:

```
res.render("home", {categories : categoriesData, badges: badgesData, statistics : data});
```

In Page:

```
<div class="rank_info" style="padding-top: 25px;">
  {{#each badges}}
    <div class="icon_holder">
      
      <span class="icon_title">{{this.name}}</span>
    </div>
  {{/each}}
</div>
```

```
<aside data-sidebar>
  <div class="menu_logo">
    TechLounge
  </div>
  <div class="padding">
    <a href="/home">Home</a>
    <br>
    <a href="/network">Network</a>
    <br>
    <a href="#" data-sidebar-button>Hide Sidebar</a>
    <br>
    <a href="/logout">Logout</a>
  </div>
</aside>
<div class="overlay" data-sidebar-overlay></div>
```

```
app.get('/home',function(req,res){
  sess=req.session;
  if(sess.userId) {
    categoryRepository.getLocalCategories(app, function(data){ // 1
      var categoriesData = data;
      badgesRepository.getLocalBadges(app, sess.userId, function(data){ // 2
        var badgesData = data;
        statisticsRepository.getLocalStatistics(app, sess.userId, function(data){ // 3
          res.render("home", {categories : categoriesData, badges: badgesData, statistics : data});
        });
      });
    });
  } else {
    res.redirect('/');
  }
});
```

- Pros

- Easy to learn
- Everything is asynchronous
- Very fast
- Concurrent request handling
- Symmetric error handling
- One environment
- Rapid development
- Single development language for server and client
- Excels at IO related features

- Cons

- Everything is asynchronous
- No standardization
- Everything goes, everywhere
- No scalability
- Callback Hell (especially for relational databases)
- Not suitable for heavy processing
- 1.5 GB of maximum RAM

THANK YOU!

Vlad Costel Ungureanu
ungureanu_vlad_costel@yahoo.com

This is a free course from [LearnStuff.io](https://learnstuff.io)
– not for commercial use –