AYSEN UNLU

**LEARNING NODE IN ONE HOUR**

1. Node JS is an open source, cross platform runtime environment for executing JavaScript code outside of a browser. We use node to build back end services.
2. Node is easy to get started and can be used for prototyping and agile development. It can be used to build super-fast, highly scalable services.
3. It’s used in production in big companies such as Uber, Netflix, Walmart and PayPal. PayPal rebuilt Java, Spring based applications using node, they discovered that it’s built twice as fast with fewer people with 33% fewer lines of code and 40% fewer files, doubled the request per second while decreased average response time by 35%.
4. In Node, since JavaScript runs on both front-end and back-end your code is clearer, and your code base is consistent.
5. Node has the largest ecosystem of open source libraries available to you. You don’t have to build the building blocks from scratch, because there’s one already written, you can focus on your core application
6. In Node, V8 Engine is incorporated in C++ Code. Node provides some capabilities that are not available inside of a browser. It works with file system, network and so on
7. Node is asynchronous by default. We have a single thread to handle the requests. The thread doesn’t wait for results e.g. from DB to arrive to serve another request.
8. Modules built in the core of node: OS, FS, Events, HTTP
9. Every file is a module in node. Variables and functions defined in the module is scoped in the module. Module exports make the functions visible outside.

**BUILD RESTFUL API’s with NODE AND EXPRESS**

1. In large and complex applications, we have various endpoints. We don’t want to hardcode. Express is a fast and light-weight framework for building web applications.
2. REST stands for REpresentational State Transfer. CRUD operations are Create, Read, Update and Delete. These are conventions for building HTTP Services.
3. **HTTP Method /*GET***: e.g. with “GET /api/customers” request we get list of all customers or with “GET /api/customers/1” request, we get the customer object with id number 1.
4. **HTTP Method /*PUT:*** e.g. with “PUT/api/customers/1 request, we include customer object with updated properties in the body of the request. This request will update the customer with id number 1 with the updated fields of the customer object passed in the body. As a response, the customer object with updated fields will be returned.
5. **HTTP Method /*DELETE:*** e.g. with “DELETE /api/customers/1” request, we delete the customer with the id number of 1.
6. **HTTP Method /*POST:*** e.g with “POST /api/customers” request, we pass a customer object to be created and on the server side, an ID will be assigned to the customer and newly created customer object will be returned in response.
7. With Express framework, we can add more routs to our application while keeping our code maintainable.
8. **How to build Web Server Using Express:**

const express=require(‘express’);

const app=express();// This is an object of type Express

app.get(‘/’,(req,res)=>{res.send (“Hello World!!”)}) app.listen(3000,()=>{});//listening