Express.js

Node.js® is a JavaScript runtime built on Chrome V8 engine. Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient. Node.js' package ecosystem, npm, is the largest ecosystem of open source libraries in the world.

express

Express is a minimal and flexible Node.js web application framework that provides a robust set of features for web and mobile applications.

Express.js is used for creating server side of web application.

How to install?

```
$ npm install express
```

vs

express

Advantage of EXPRESS.JS

1. Has simple syntax .
2. Obvious API.
3. The biggest community.
4. A lot of modules.

1. Express.js is creation of StrongLoop
   An IBM Company

Frameworks built on Express

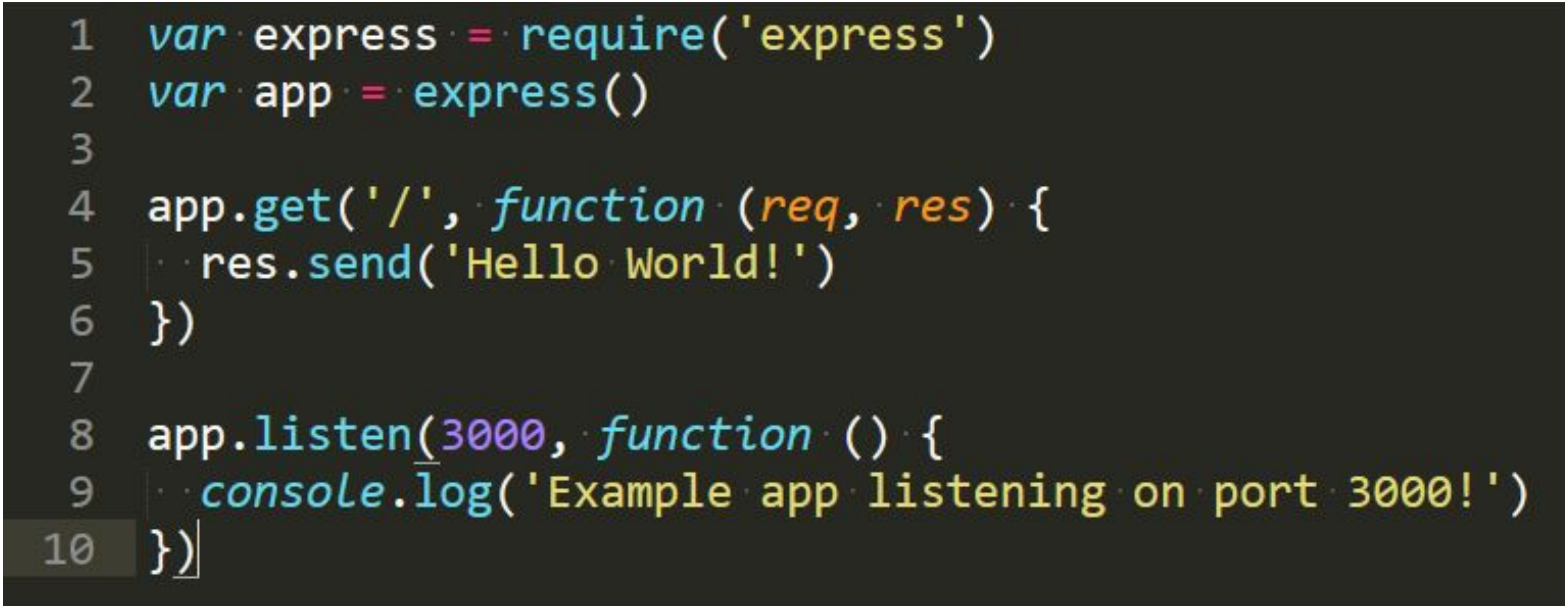
Several popular Node.js frameworks are built on Express:

- **Feathers**: Build prototypes in minutes and production ready real-time apps in days.
- **ItemsAPI**: Search backend for web and mobile applications built on Express and Elasticsearch.
- **KeystoneJS**: Website and API Application Framework / CMS with an auto-generated React.js Admin UI.
- **Kraken**: Secure and scalable layer that extends Express by providing structure and convention.
- **LEAN-STACK**: The Pure JavaScript Stack.
- **LoopBack**: Highly-extensible, open-source Node.js framework for quickly creating dynamic end-to-end REST APIs.
- **MEAN**: Opinionated fullstack JavaScript framework that simplifies and accelerates web application development.
- **Sails**: MVC framework for Node.js for building practical, production-ready apps.
- **Bottr**: Framework that simplifies building chatbot applications.

Express or not Express ?

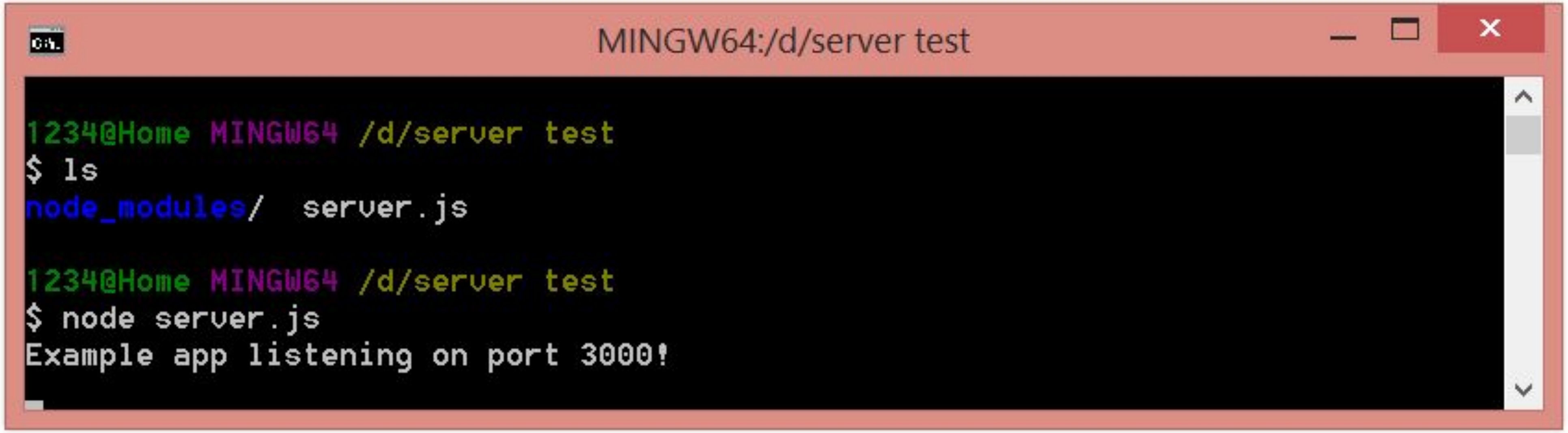
Let’s write some code

Simple server on Express.js

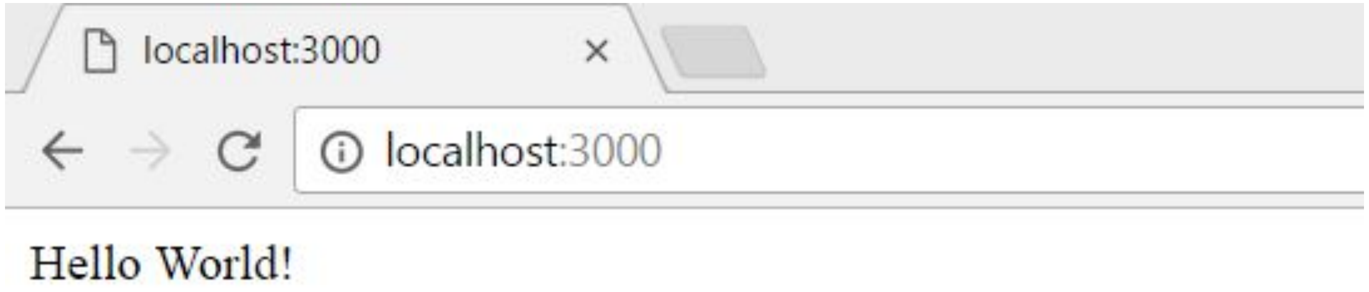


```javascript
var express = require('express')
var app = express()

app.get('/', function (req, res) {
  res.send('Hello World!')
})

app.listen(3000, function () {
  console.log('Example app listening on port 3000!')
})
```

Testing server



```
1234@Home MINGW64 /d/server test
$ ls
node_modules/  server.js

1234@Home MINGW64 /d/server test
$ node server.js
Example app listening on port 3000!
```



Hello World!

Some abstruse word for adventurous

- Middleware
- Routing
- Pattern Front controller
- Pattern MVC
- Pattern Chain of responsibility
- Templates
- Views (Partials & Layouts)
- Error handling

The End