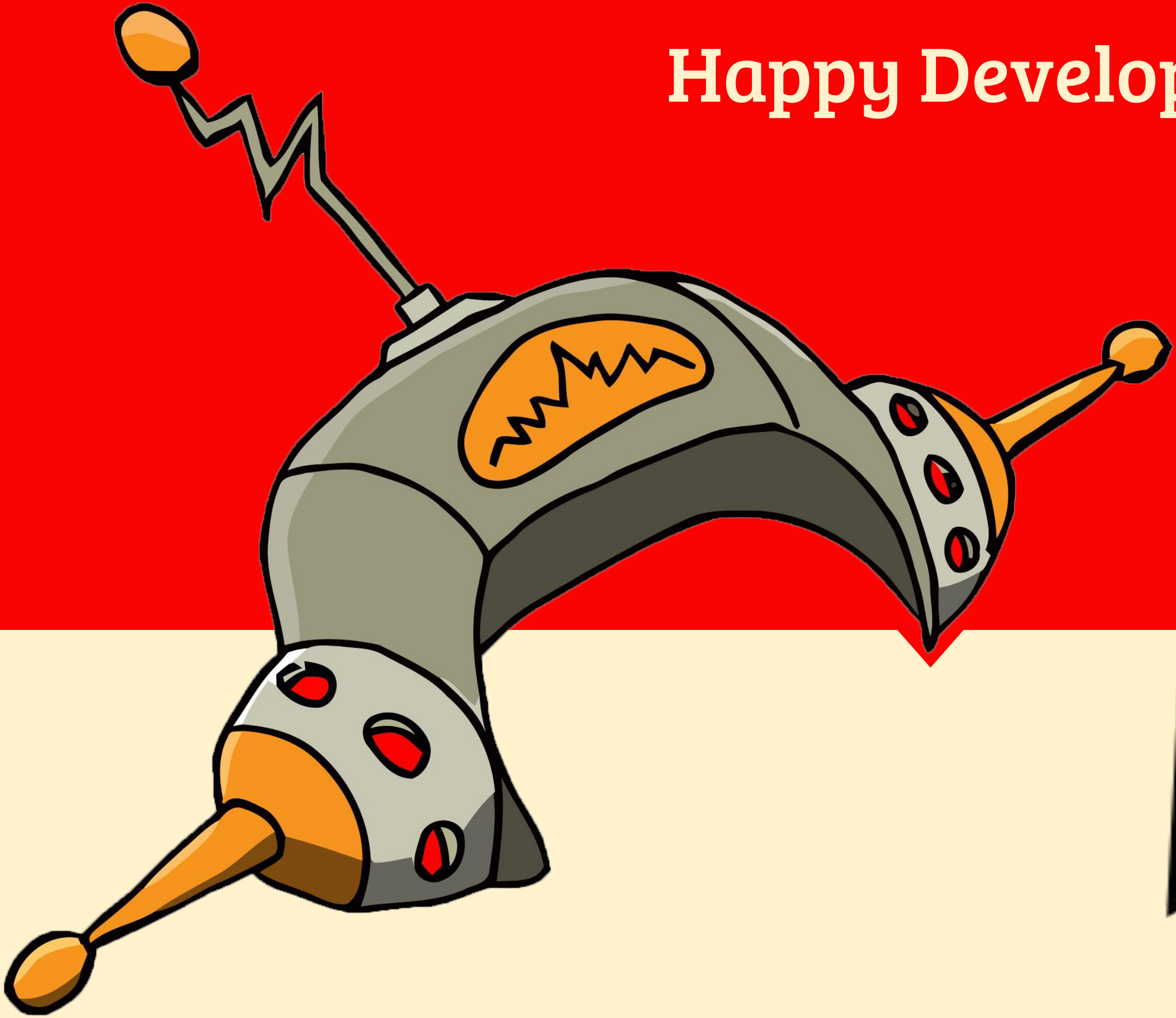
Happy Development with Hapi.js



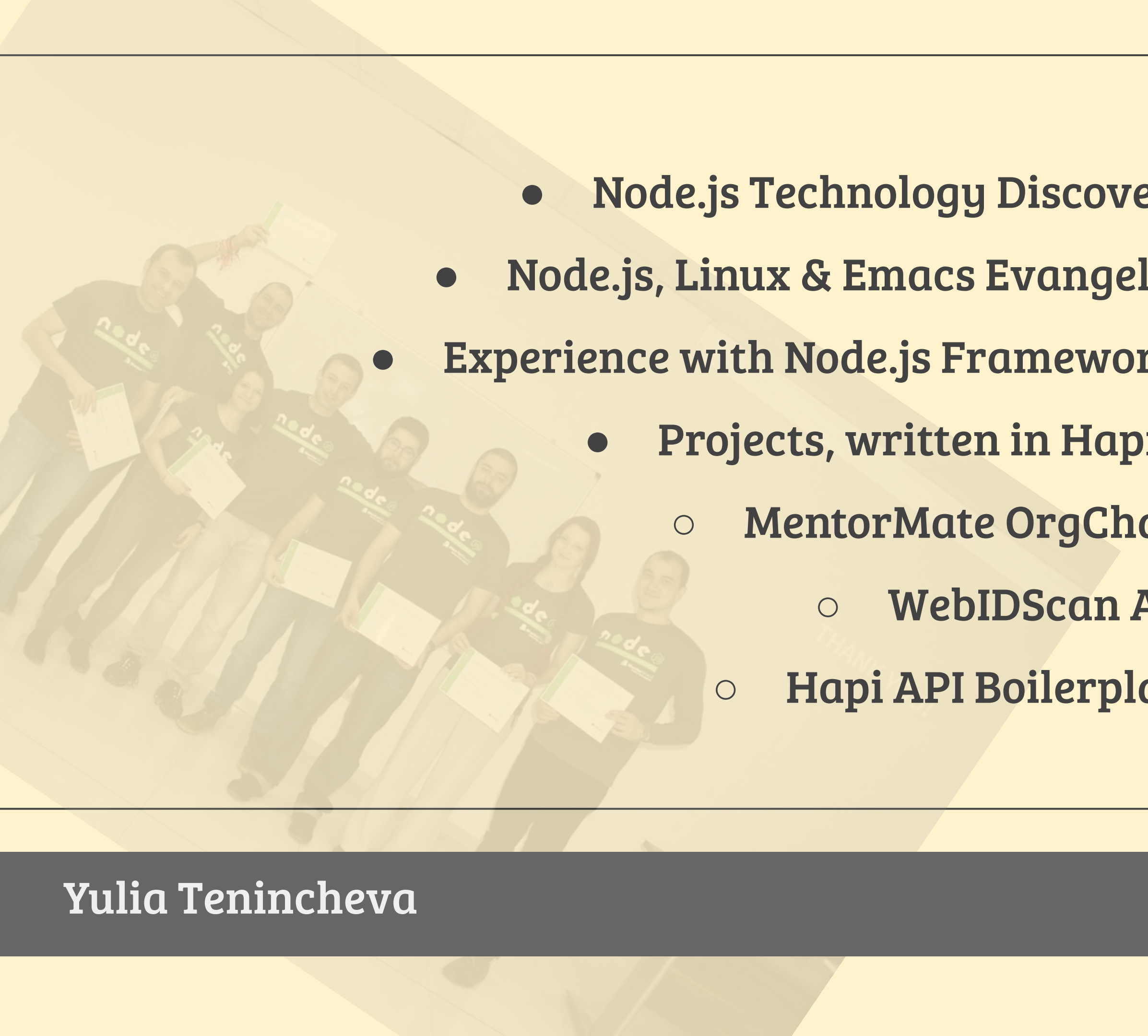
hapi

October, 2017

Recipe for Hapiness

1. When node hapi, learn Node.js
2. Try other frameworks and struggle with them a lot
3. Run

```
$ npm install -g makemehapi

$ makemehapi
```



- Node.js Technology Discovery
- Node.js, Linux & Emacs Evangelist
- Experience with Node.js Frameworks
- Projects, written in Hapi.js
  - MentorMate OrgChart
  - WebIDScan API
  - Hapi API Boilerplate

Yulia Tenincheva

Yulia Tenincheva

How (Agenda)

1. Introduction
2. Getting Started (basic server, routing, serving static content)
3. Response & Error Handling with Boom()
4. Validation with Joi
5. Plugins - The ❤ of Hapi
6. Opinionated by Glue
7. Quality Code, Security & Testing
8. Demo & Resources

Hapi is...

- Configuration > Code
- Feature Rich
- Reusable
- Secure
- Scalable
- Production ready
- ... just Awesome

Introducing Hapi

★ History

- 2011, Walmart Labs
- Black Friday Success

★ Contributors

- Eran Hammer

★ **Community**

Jul 31, 2011 – Jul 20, 2017

Contributions to master, excluding merge commits

Contributions: Commits

hueniverse #1
2,166 commits / 182,186 ++ / 167,828 --

geel #2
580 co

Hapi.Server()

```javascript
'use strict';

const Hapi = require('hapi');
const server = new Hapi.Server();

server.connection({ port: 3000, host: 'localhost' });

server.start(err => {

  if (err) {
    throw err;
  }

  console.log(`Server running at: ${server.info.uri}`);
});

```

Example

```javascript
'use strict';

const Hapi = require('hapi');
const server = new Hapi.Server();

server.connection([
  { port: 3000, host: 'localhost' },
  { port: 8080, host: 'localhost', labels: ['api'] }
]);

server.start(err => {
  if (err) throw err;

  server.connections.forEach(connection => {
    console.log(`server is listening on ${connection.info.uri}`);
  });
});
```

Example

Hello World!

```javascript
const Hapi = require('hapi');
const server = new Hapi.Server();

server.connection({ port: 3000, host: 'localhost' });

server.route({
  method: 'GET',
  path: '/',
  handler: function (request, reply) {
    reply('Hello World!');
  }
});

server.start(err => {
  if (err) throw err;
  console.log(`Server running at: ${server.info.uri}`);
});
```

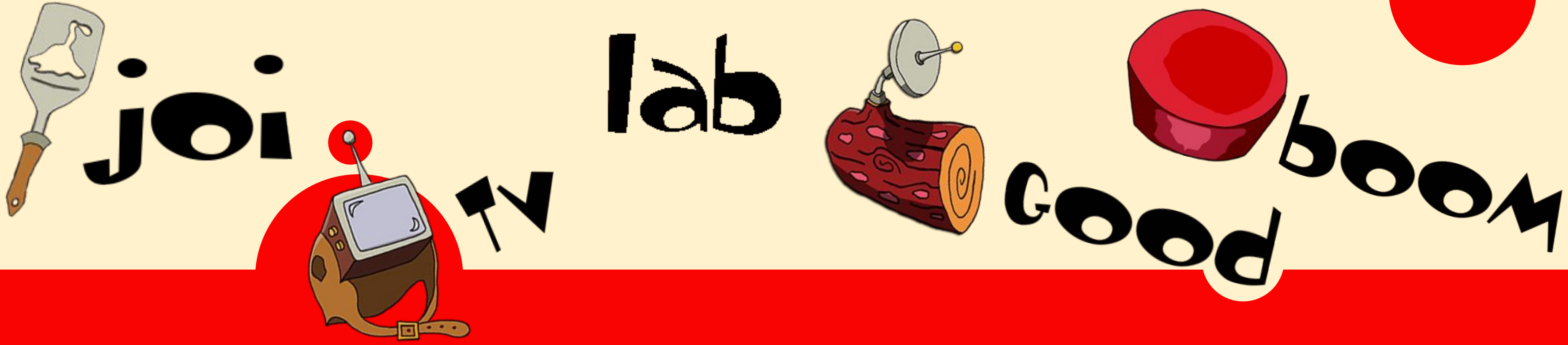
Example

```
server.route({ method: ['PUT', 'POST'], path: '/user', handler: ...});

server.route({ method: 'GET', path: '/hello/{user}', handler: hello});
server.route({ method: 'GET', path: '/hello/{user?}', handler: hello});

function hello(request, reply) {
  reply(`<h1>Hello, ${request.params.user}!</h1>`);
}

server.route({
  method: 'GET',
  path: '/hello/{user*2}',
  handler: function(request, reply) {
    const userParts = request.params.user.split('/');
    reply(`Heey, ${userParts[0]}, ${userParts[1]}!`);
  }
});
```

Example

```
const api = server.select('api');

api.route({
  method: 'GET',
  path: '/',
  handler: function(request, reply) {
    reply(`API index`);
  }
});

// all servers with a label of backend OR api
const myServers = server.select(['backend', 'api']);

// servers with a label of api AND admin
const myServers = server.select('api').select('admin');


```

Example



Plugins

Hapi Plugins

1. How to write a plugin
   - structure (server, options, next)
   - register method & attributes
   - internal vs external

2. How to load the plugin
   - external / internal
   - one by one / multiple
   - pass options
   - select a connection

```javascript
const myPlugin = {
  register: function (server, options, next) {


    // Do something


    next();
  }
};

myPlugin.register.attributes = {
  name: 'myPlugin',
  version: '1.0.0',
  multiple: false
};
```

Example

```javascript
const myPlugin = {
  register: function (server, options, next) {

    server.route({
      method: 'GET',
      path: '/doSomething',
      handler: function (request, reply) {
        reply.success();
      }
    });

    next();
  }
};

myPlugin.register.attributes = { name: 'myPlugin', version: '1.0.0' };
```

Example

```javascript
// load one plugin
server.register(require('myPlugin'), (err) => {
  if (err) {
    console.error('Failed to load plugin:', err);
  }
});

// pass options
server.register({
  register: require('myPlugin'),
  routes: { prefix: '/plugins' },
  options: {
    secret: process.env.SECRET
  }
}, (err) => {
    if (err) throw err;
});
```

Example

```javascript
// load multiple plugins
server.register([require('myPlugin'), require('yourPlugin')], (err) => {
  if (err) {
    console.error('Failed to load plugin:', err);
  }
});

// pass options
server.register({
  register: require('myPlugin'),
  routes: { prefix: '/plugins' },
  options: {
    secret: process.env.SECRET
  }
}, (err) => {
    if (err) throw err;
});
```

Example

Glue your plugins

Server composer for Hapi

- `server = new Hapi.Server(options)`
- one or more `server.connection(options)`
- one or more `server.register(plugin, options)`

Hapi Serving

Serving Static Content

1. Don’t use Hapi for that.
2. Inert plugin
3. File Handler => reply.file();
4. Directory Handler

Set your Vision

1. Supported Template Engines? Any!
2. Vision Plugin Configuration & API

> Example code from MentorMate OrgChart - Hapi Version PoC

Route Back

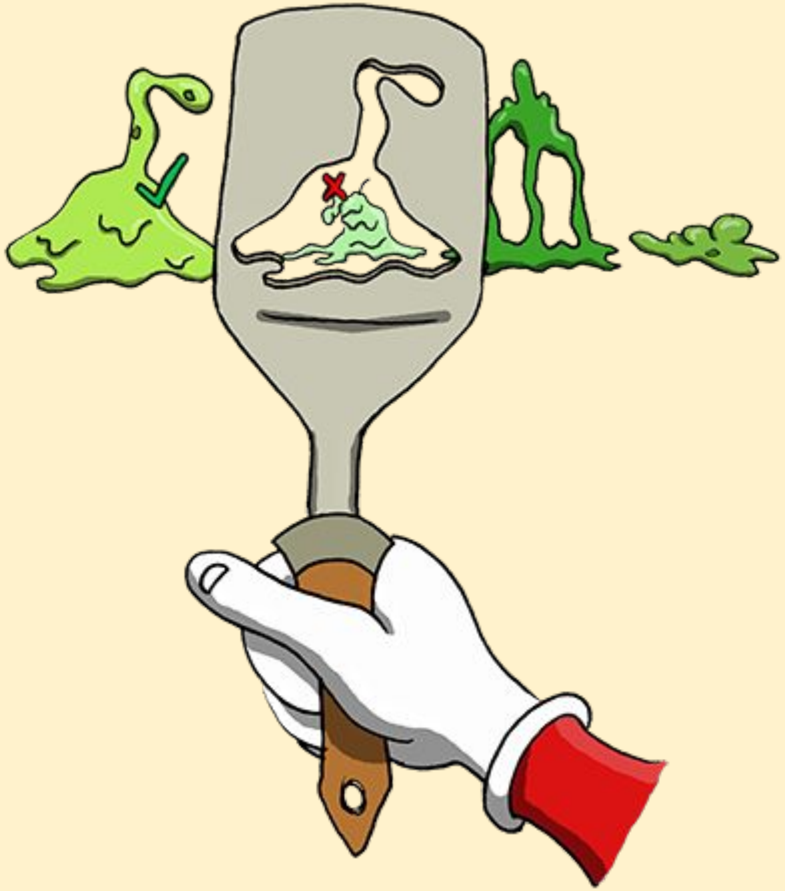
The joi of reusable validation

1. Trivial yet important for every API

2. Validating hapi routes with joi
   - Example code from Hapi API Boilerplate

3. Validation model schemas and ...everything
   - Example code from Hapi API Boilerplate

4. Documentation generation (wow!)
   - Demo with Swagger /documentation



From the Lab to Prod

1. Borrowed heavily from mocha

2. Lab.js & Code
   - Writing tests
   - Code coverage
   - Linting
   - Reporters

3. Testing hapi applications with lab

4. Continuous Integration Setup

DEMO

The Stack

Node.js - v8.x.x
Hapi.js + cool plugins
Redis
MongoDB / PostgreSQL
GitLab-CI with Docker
Swagger UI

Resources

- Project Starter with MongoDB
- Project Starter with Sequelize
- hapijs.com Docs, Tutorials, etc
- DWYL organization on GitHub
- API-Security-Checklist on GitHub

Thanks.

Yay. Gray slide.