

Building SPA with Node.js and Angular 2

04 Mar 17 | 1.00 PM - 5.00 PM

Location : Nextzy Chiang Mai

By



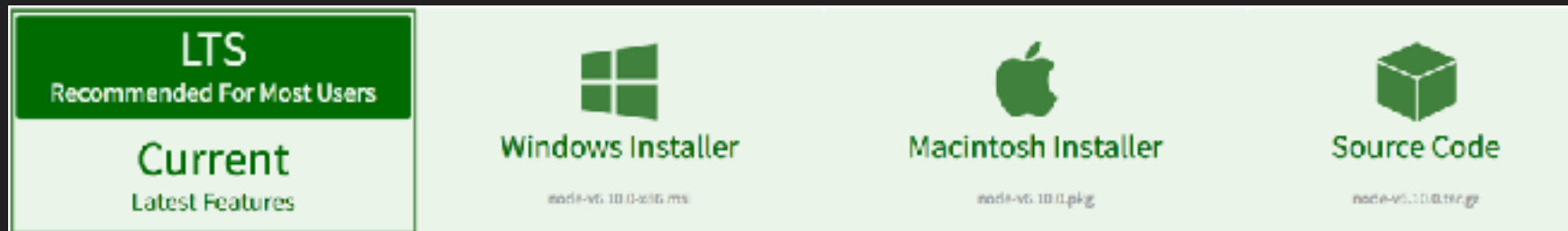


Phonbopit Sahakitchachawan
(Chai Phonbopit)

Software Engineer @Nextzy, Writer @DevAhoy



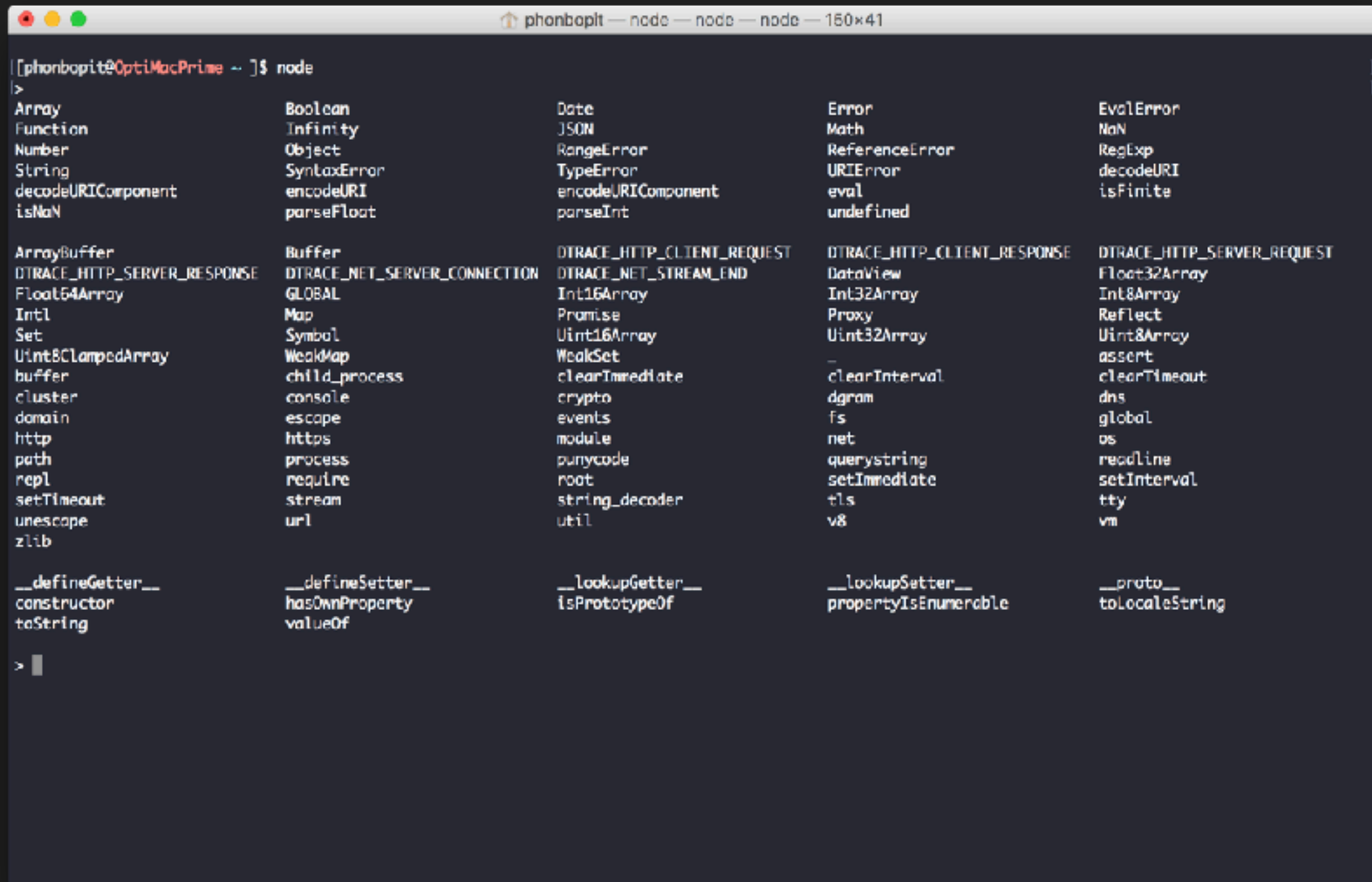
Install Node.js



```
brew install node
```

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

API Docs



```
[phorbopit@OptiMacPrime ~]$ node
>
Array
Function
Number
String
decodeURIComponent
isNaN
Boolean
Infinity
Object
SyntaxError
encodeURIComponent
parseFloat
Date
JSON
RangeError
TypeError
encodeURIComponent
parseInt
Error
Math
ReferenceError
URIError
eval
undefined
EvalError
NaN
RegExp
decodeURI
isFinite

ArrayBuffer
DTRACE_HTTP_SERVER_RESPONSE
Float64Array
Intl
Set
Uint8ClampedArray
buffer
cluster
domain
http
path
repl
setTimeout
unescape
zlib
Buffer
DTRACE_NET_SERVER_CONNECTION
GLOBAL
Map
Symbol
WeakMap
child_process
console
escape
https
process
require
stream
url
DTRACE_HTTP_CLIENT_REQUEST
DTRACE_HTTP_CLIENT_RESPONSE
DTRACE_HTTP_SERVER_REQUEST
DTRACE_NET_STREAM_END
Int16Array
Promise
Uint16Array
WeakSet
clearImmediate
crypto
events
module
punycode
root
string_decoder
util
DTRACE_HTTP_CLIENT_RESPONSE
DataView
Int32Array
Proxy
Uint32Array
-
clearInterval
dgram
fs
net
querystring
setImmediate
tls
v8
DTRACE_HTTP_SERVER_REQUEST
Float32Array
Int8Array
Reflect
Uint8Array
assert
clearTimeout
dns
global
os
readline
setInterval
tty
vm

__defineGetter__
constructor
toString
__defineSetter__
hasOwnProperty
valueOf
__lookupGetter__
isPrototypeOf
__lookupSetter__
propertyIsEnumerable
__proto__
toLocaleString
```

- <https://nodejs.org/dist/latest-v6.x/docs/api/>

Node command

```
# Node REPL (Read Eval Print Loop)  
$ node  
# Run a node file  
$ node FILENAME.js  
  
# Install node dependency  
$ npm install AWESOME_PROJECT  
# ninstall  
$ npm uninstall AWESOME_PROJECT
```

Run a server

```
const http = require('http');

const hostname = '127.0.0.1';
const port = 3000;

const server = http.createServer((req, res) => {
  res.statusCode = 200;
  res.setHeader('Content-Type', 'text/plain');
  res.end('Hello World\n');
});

server.listen(port, hostname, () => {
  console.log(`Server running at http://${hostname}:${port}/`);
});
```

require & export

```
// hello.js  
module.exports = function() {  
  console.log('Hello');  
};  
  
// main.js  
const sayHello = require('./hello');  
sayHello();  
  
// run node  
$ node main.js  
Hello
```


JavaScript 101

Data Types

- Boolean
- Number
- String
- Null
- Undefined
- Object

```
var foo;           // undefined
var foo = 20;      // Number
var foo = 'Chai';  // String
var foo = false;   // Boolean
var foo = {};      // Object
var foo = null;    // null
```

Scope

```
foo = 'foo';  
var bar = 'bar';  
  
function hi() {  
  var foo = 'foo2';  
  var bar = 'bar2';  
  console.log('Hello ' + foo + ' ' + bar);  
}  
  
console.log(foo);    // foo  
console.log(bar);    // bar  
hi();                // Hello foo2 bar2
```

```
function hi() {  
  console.log(foo);  
  var foo = 'foo';  
}  
  
hi()    // undefined
```

Hoisting

```
hi(); // Error : hi is not a function
var hi = function() {
  console.log('Hello World');
}
```



```
var hi; // undefined
hi();

hi = function() {
  console.log('Hello World');
}
```

```
function sayHello() {
  for (var i = 1; i <= 1; i++) {
    var name = 'Chai';
  }

  console.log(name);
}

sayHello(); // Name
```



```
function sayHello() {
  var name; // undefined

  for (var i = 1; i <= 1; i++) {
    name = 'Chai';
  }

  console.log(name);
}

sayHello(); // Name
```

Functions

```
// function declaration
function foo() {
    return 0;
}
```

```
// function expressions
var foo = function() {
    return 0;
}
```

```
// call function
foo();
```

```
// IIFE
(function() {
    console.log('Hello');
})();
```

Array

```
// array
var foo = new Array();
var bar = [];
var numbers = [1, 2, 3, 4, 5, 6, 7, 8];
var arr = [1, false, 'OK'];

// access array
arr[1];

arr.push(2); // [1, false, "OK", 2]
arr.pop()    // [1, false, "OK"]

arr.forEach(function(data) {
  console.log(data);
});
```


Object

```
// object
var foo = new Object();
var bar = {};
var dog = {
  name: 'Jack'
};

dog.name      // 'Jack';
dog.age       // undefined

// assign 20 to variable age
dog.age = 20;

var dog = {
  name: 'Jack',
  bark: function() {
    console.log('bok bok!!');
  }
};

dog.bark();    // bok bok!!
```

Callback function

```
$("#btn-submit").click(function() {  
    alert("Button Clicked");  
});
```

```
function sayHi(name, callback) {  
    setTimeout(function() {  
        return callback(name);  
    }, 1000);  
}  
  
sayHi('Chai', function(name) {  
    console.log('name is ', name);  
});  
  
console.log('last line');
```

JavaScript ES6

Babel is a JavaScript compiler.

Use next generation JavaScript, today.

Put in next-gen JavaScript	Get browser-compatible JavaScript out
<pre>var obj = { shorthand, method() { return "😬"; } };</pre>	<pre>var obj = { shorthand: shorthand, method: function method() { return "😬"; } };</pre>
Check out our REPL to experiment more with Babel!	

<https://babeljs.io/>

let & const

```
let name = 'Chai';  
let name = 'C';           // 'name' has already been declared  
  
const foo;                // error missing initializer  
const bar = 'bar';  
const bar = 'bars';       // 'bar' has already been declared
```

```
for (var i = 0; i < 2; i++) {  
  console.log('var loop');  
}  
console.log(i)    // 2  
  
for (let j = 0; j < 2; j++) {  
  console.log('let loop');  
}  
console.log(j)    // ReferenceError j is not defined
```

```
const a = {  
  a: "a"  
}  
a.a = 'b';  
  
console.log(a);    // {"a": "b"}
```

import & exports

```
import someModule from 'module-name';  
import * as name from 'module-name';  
import { member } from 'module-name';
```

```
export {  
  name1,  
  name2,  
  name3  
};
```

```
// dog.js  
export default class Dog {  
  walk() {}  
}
```

```
// main.js  
import Dog from './dog';
```


String Interpolation

```
const name = 'Chai';

// ES5
console.log('Hello ' + name + '!');
// ES6
console.log(`Hello ${name}!`);

// ES5
console.log('<div class="container"><div class="row"><p>Hello World</p></div></div>');
// ES6
console.log(`
  <div class="container">
    <div class="row">
      <p>Hello World</p>
    </div>
  </div>
`);
```

Default parameters

```
// ES5
function sum(a, b) {
    if (b === undefined) {
        b = 10
    }
    return a + b;
}

// ES6
function sum(a, b=10) {
    return a + b;
}
console.log(sum(10)) // 20
```

Arrow Function

```
var numbers = [1, 2, 3, 4, 5];
```

```
numbers.forEach(function(val) {  
  console.log(val);  
});
```

```
// short hand for `function()` but does not bind `this`  
numbers.forEach((val) => {  
  console.log(val);  
});
```



Build API with Hapi.js

```
npm install hapi
```

Server with Hapi

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

server.connection({
  host: 'localhost',
  port: 3001
});

server.route({
  method: 'GET',
  path: '/',
  handler: (request, reply) => {
    return reply({message: 'Hello World'});
  }
});

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

Routing

```
server.route({
  method: 'GET',
  path: '/hello',
  handler: (request, reply) => {
    return reply({
      message: 'Hello'
    });
  }
});
```

```
server.route({
  method: 'GET',
  path: '/hello/{name}',
  handler: (request, reply) => {
    return reply({
      message: `Hello ${request.params.name}`
    });
  }
});
```

```
server.route({
  method: 'POST',
  path: '/hello',
  handler: (request, reply) => {
    const payload = request.payload;
  }
});
```


Request Object

```
// Params
// GET /api/players/chai
const params = request.params;
params.name // chai

// Query String
// GET /api/players?foo=foo&bar=baz
const query = request.query;
query.foo // foo
query.bar // baz
query.baz // undefined

// Payload
// POST /api/players {name: 'chai'}
// curl -X POST -d "{name: 'chai'}"
const payload = request.payload;
payload.name // chai
```

Validate with Joi

```
const schema = {  
  a: Joi.string()  
};
```

```
query: {  
  limit: Joi.number().integer().min(1).max(100).default(30),  
  sort: Joi.string(),  
  fields: Joi.string()  
},  
  
params: {  
  name: Joi.string().required()  
},  
  
payload: {  
  username: Joi.string().required(),  
  email: Joi.string().email(),  
  password: Joi.string().regex(/^[a-zA-Z0-9]{8,30}$/)  
}
```

Custom Errors with Boom

```
Boom.badRequest()    // 400
Boom.unauthorized()  // 401
Boom.notFound()      // 404

// GET /api/players
reply(Boom.badRequest('invalid query'));

{
  "statusCode": 400,
  "error": "Bad Request",
  "message": "invalid query"
}

// GET /api/auth/login

reply(Boom.unauthorized('invalid password'));

{
  "statusCode": 401,
  "error": "Unauthorized",
  "message": "invalid password"
}
```

Plugins

```
const inert = require('inert');  
server.register(inert, (err) => {  
  // next()  
});
```

```
server.register([  
  register: inert  
, {  
  register: vision  
, {  
  register: 'another-plugin',  
  options: {}  
}], (err) => {  
  // next()  
});
```

Example

- vision : Template Rendering
- inert : Static file handlers
- boom : HTTP Error object
- joi : Object validation
- lab : Test Framework
- code : Assertion library
- bell : 3rd Party for login
- good : Process monitoring

Project Structures

```
hapi — phonboplt@OptiMacPrime — ../example/hapi — -zsh — 73x33
[[phonboplt@OptiMacPrime hapi (master x)]$ tree -L 3
.
├── README.md
├── gulpfile.js
├── package.json
├── src
│   ├── auth
│   ├── club
│   │   ├── club.controller.js
│   │   ├── club.model.js
│   │   ├── club.route.js
│   │   ├── club.service.js
│   │   └── club.validate.js
│   ├── config.js
│   ├── constants.js
│   ├── player
│   │   ├── player.controller.js
│   │   ├── player.model.js
│   │   ├── player.route.js
│   │   ├── player.service.js
│   │   └── player.validate.js
│   ├── routes.js
│   ├── server.js
│   └── utils
│       ├── authen.js
│       ├── crypto.js
│       └── logger.js
└── test
    └── index.spec.js

6 directories, 21 files
[[phonboplt@OptiMacPrime hapi (master x)]$
```


JSON Web Token (JWT)

Encoded

PASTE A TOKEN HERE

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9lIiwiaWF0IjoiYWRtaW4iOnRydWV9.TJVA95OrM7E2cBab30RMHrHDcEfxjoYZgeFONFh7HgQ
```

Decoded

EDIT THE PAYLOAD AND SECRET (ONLY HS256 SUPPORTED)

HEADER: ALGORITHM & TOKEN TYPE

```
{  "alg": "HS256",  "typ": "JWT"}
```

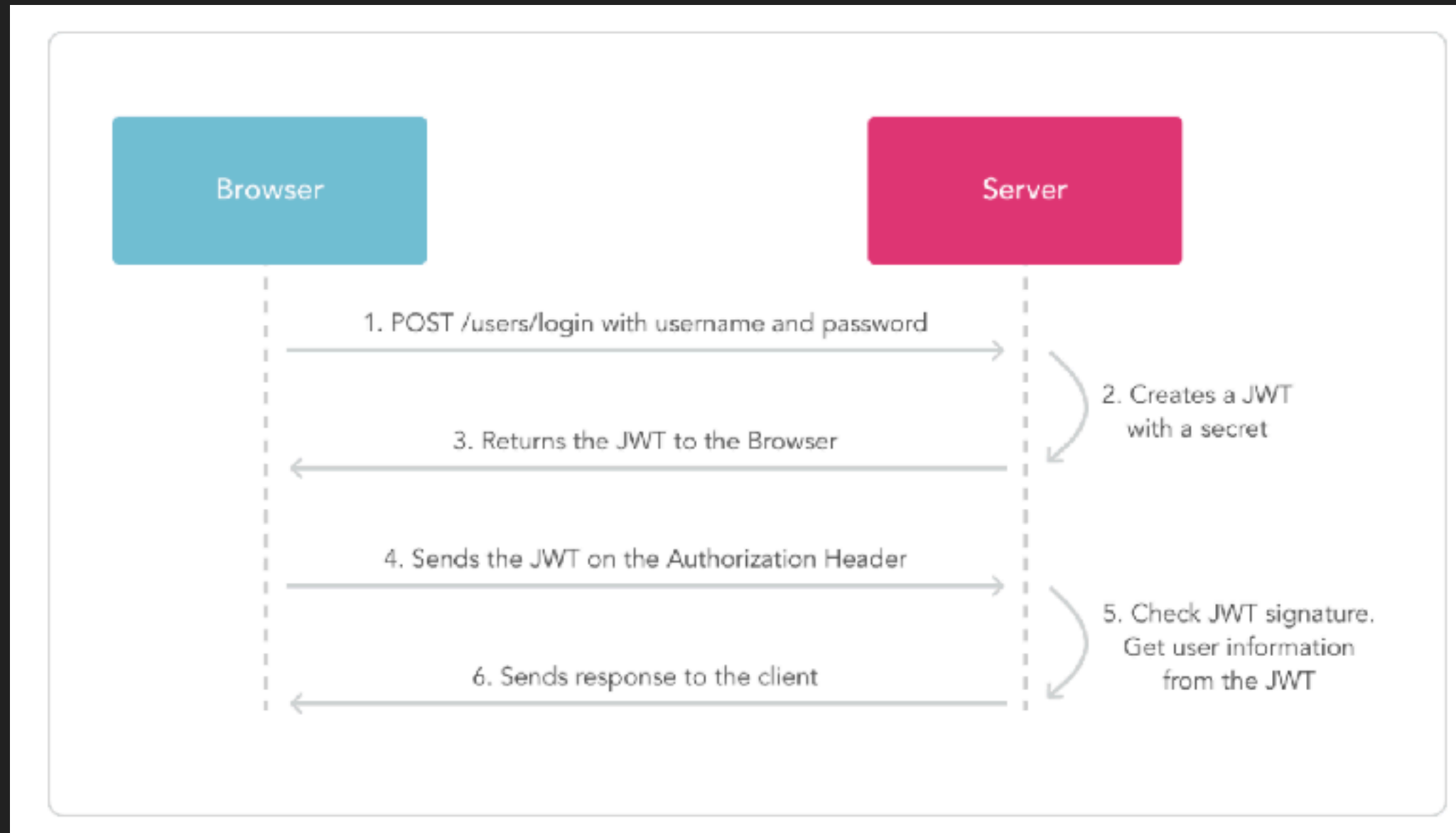
PAYLOAD: DATA

```
{  "sub": "1234567890",  "name": "John Doe",  "admin": true}
```

VERIFY SIGNATURE

```
HMACSHA256(  
  base64UrlEncode(header) + "." +  
  base64UrlEncode(payload),  
  secret  
) ☐ secret base64 encoded
```

JSON Web Token (JWT)



Hapi Auth JWT

```
const HapiAuth = require('hapi-auth-jwt2');
const JWT = require('jsonwebtoken');

// register hapi auth jwt
server.register(HapiAuth);

server.auth.strategy('jwt', 'jwt', {
  key: 'mysecretKey',
  validateFunc: validate
});

server.route({
  method: 'GET',
  path: '/api/me',
  config: {
    auth: 'jwt',
    handler: (request, reply) => { reply({})}
  }
});
```

Hapi Auth JWT(2)

```
server.auth.default('jwt');

server.route({
  method: 'GET',
  path: '/api/me',
  config: {
    handler: (request, reply) => { reply({}) }
  }
});

server.route({
  method: 'POST',
  path: '/api/auth/login',
  config: {
    auth: false,
    handler: (request, reply) => { reply({}) }
  }
});
```

Hapi Auth JWT(3)

```
function validate(decoded, request, callback) {  
  // User.findOne({username: decoded.username}).then()...  
  
  if (decoded.name === 'chai') {  
    return callback(null, true);  
  } else {  
    return callback('User not found', false);  
  }  
}
```

Hapi Auth Response

```
HTTP/1.1 401 Unauthorized
Connection: keep-alive
Date: Sat, 04 Mar 2017 04:05:32 GMT
Transfer-Encoding: chunked
WWW-Authenticate: Token
cache-control: no-cache
content-encoding: gzip
content-type: application/json; charset=utf-8
vary: accept-encoding

{
  "error": "Unauthorized",
  "message": "Missing authentication",
  "statusCode": 401
}
```

Hapi Auth Response(2)

```
HTTP/1.1 401 Unauthorized
Connection: keep-alive
Date: Sat, 04 Mar 2017 04:05:48 GMT
Transfer-Encoding: chunked
WWW-Authenticate: Token error="Invalid token format"
cache-control: no-cache
content-encoding: gzip
content-type: application/json; charset=utf-8
vary: accept-encoding

{
  "attributes": {
    "error": "Invalid token format"
  },
  "error": "Unauthorized",
  "message": "Invalid token format",
  "statusCode": 401
}
```

Example

- <https://github.com/Phonbopit/hapi-auth-jwt-example>
- <https://github.com/Phonbopit/node-x-angular/tree/master/example/api>



Angular

Angular CLI

```
npm install -g typescript  
npm install -g @angular/cli  
brew install watchman
```

```
# Create new app  
ng new project_name
```

```
# Run a server  
ng serve
```

```
# Build dist folder  
ng build
```

Angular CLI

Scaffold	Usage
Component	<code>ng g component my-new-component</code>
Directive	<code>ng g directive my-new-directive</code>
Pipe	<code>ng g pipe my-new-pipe</code>
Service	<code>ng g service my-new-service</code>
Class	<code>ng g class my-new-class</code>
Interface	<code>ng g interface my-new-interface</code>
Enum	<code>ng g enum my-new-enum</code>
Module	<code>ng g module my-module</code>

App Module

```
@NgModule({
  declarations: [
    AppComponent,
    AboutComponent,
    HeroComponent,
    HeroListComponent,
    HomeComponent,
    NavbarComponent
  ],
  imports: [
    BrowserModule,
    FormsModule,
    HttpClientModule,
    AppRoutingModule
    //RouterModule.forRoot(routes)
  ],
  providers: [HeroService],
  bootstrap: [AppComponent]
})
export class AppModule { }
```

Components

```
ng generate component hello-world
```



```
installing component
  create src/app/hello-world/hello-world.component.css
  create src/app/hello-world/hello-world.component.html
  create src/app/hello-world/hello-world.component.spec.ts
  create src/app/hello-world/hello-world.component.ts
  update src/app/app.module.ts
```

```
import { Component, OnInit } from '@angular/core';

@Component({
  selector: 'app-hello-world',
  templateUrl: './hello-world.component.html',
  styleUrls: ['./hello-world.component.css']
})
export class HelloWorldComponent implements OnInit {

  constructor() { }

  ngOnInit() {
  }

}
```



```
<app-hello-world></app-hello-world>
```

Function

```
// html
<button (click)="like()">Like</button>

// component.ts
export class Component implements OnInit {

  hero: any;

  constructor() { }

  like() {
    console.log('like');
  }

}
```

Routing

```
// html tag
<base href="/">

// import RouterModule
import { RouterModule, Routes } from '@angular/router';

// defined routes
const routes: Routes = [{
  path: '',
  component: YourComponent
}];

// pass routes to forRoot()
@NgModule({
  imports: [ RouterModule.forRoot(routes) ]
});

// rendering routed view
<router-outlet></router-outlet>
```

Routing

```
<a routerLink="/"></a>

<nav class="nav">
  <div class="nav-left">
    <a class="nav-item" routerLinkActive="active" routerLink="/">Home</a>
  </div>

  <div id="nav-menu" class="nav-right nav-menu">
    <a class="nav-item" routerLinkActive="active" routerLink="/heroes">
      Heroes
    </a>
    <a class="nav-item" routerLinkActive="active" routerLink="/about">
      About
    </a>
  </div>
</nav>
```


Service

```
import { Injectable } from '@angular/core';
import { Http } from '@angular/http';
import 'rxjs/add/operator/map';

@Injectable()
export class HeroService {

  constructor(private http: Http) { }

  findAll() {
    return this.http.get('http://localhost:5000/api/heroes')
      .map(res => res.json());
  }

  findOne(id) {
    return this.http.get(`http://localhost:5000/api/heroes/${id}`)
      .map(res => res.json());
  }
}
```

Service(2)

```
import { Component, OnInit } from '@angular/core';
import { HeroService } from '../hero.service';

@Component({
  selector: 'app-hero-list',
  templateUrl: './hero-list.component.html',
  styleUrls: ['./hero-list.component.scss']
})
export class HeroListComponent implements OnInit {

  heroes: any = [];

  constructor(private heroService: HeroService) { }

  ngOnInit() {
    this.heroService.findAll().subscribe(response => {
      this.heroes = response.data;
    });
  }

}
```

```
@NgModule({
  ...
  providers: [HeroService]
})
```

Hapi + Angular2

```
server.connection({  
  host: 'localhost',  
  port: 5000,  
  routes: {  
    cors: true,  
    files: {  
      relativeTo: path.join(__dirname, 'dist')  
    }  
  }  
});
```

Hapi + Angular2

```
const controller = require('./controller');

module.exports = [{
  method: 'GET',
  path: '/{path*}',
  config: controller.index
}, {
  method: 'GET',
  path: '/api/heroes',
  config: controller.heroes.findAll
}, {
  method: 'GET',
  path: '/api/heroes/{id}',
  config: controller.heroes.findOne
}]
```

Hapi + Angular2

```
const path = require('path');
const mock = require('./response');

const index = {
  auth: false,
  handler: {
    directory: {
      path: path.join(__dirname, '../dist'),
      listing: false,
      index: ['index.html']
    }
  }
};

const heroes = {
  findAll: {
    auth: false,
    handler: (request, reply) => {
      return reply({
        data: mock
      });
    }
  },
  findOne: {
    auth: false,
    handler: (request, reply) => {
      const id = request.params.id;
      return reply({
        data: mock.filter(hero => hero.id === id)[0]
      });
    }
  }
};

module.exports = {
  index,
  heroes
};
```

Hapi + Angular2

```
server.ext('onPostHandler', (request, reply) => {  
  const response = request.response;  
  if (response.isBoom && response.output.statusCode === 404) {  
    return reply.file('index.html');  
  }  
  return reply.continue();  
});
```

Hapi + Angular2

```
server.ext('onPostHandler', (request, reply) => {  
  const response = request.response;  
  if (response.isBoom && response.output.statusCode === 404) {  
    return reply.file('index.html');  
  }  
  return reply.continue();  
});
```

```
[phonbopit@OptiMacPrime node-x-angular (master)]$ ng build  
Hash: 3e304e594cb14b6d3931  
Time: 7529ms  
chunk    {0} polyfills.bundle.js, polyfills.bundle.js.map (polyfills) 153 kB {4} [initial] [rendered]  
chunk    {1} main.bundle.js, main.bundle.js.map (main) 20.8 kB {3} [initial] [rendered]  
chunk    {2} styles.bundle.js, styles.bundle.js.map (styles) 9.89 kB {4} [initial] [rendered]  
chunk    {3} vendor.bundle.js, vendor.bundle.js.map (vendor) 2.69 MB [initial] [rendered]  
chunk    {4} inline.bundle.js, inline.bundle.js.map (inline) 0 bytes [entry] [rendered]  
[phonbopit@OptiMacPrime node-x-angular (master)]$
```



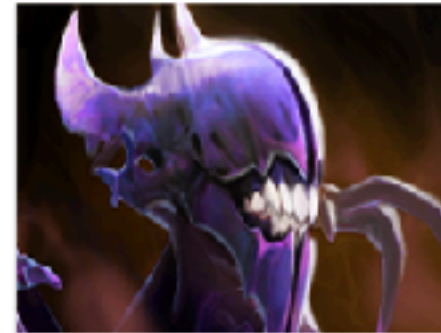

Anti-Mage

npc_dota_hero_antimage



Axe

npc_dota_hero_axe



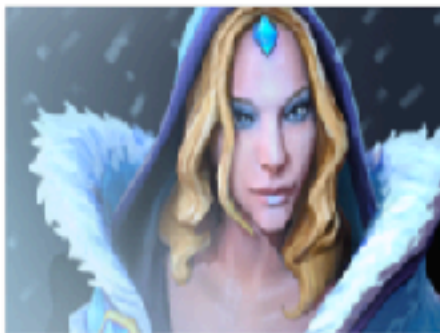
Bane

npc_dota_hero_bane



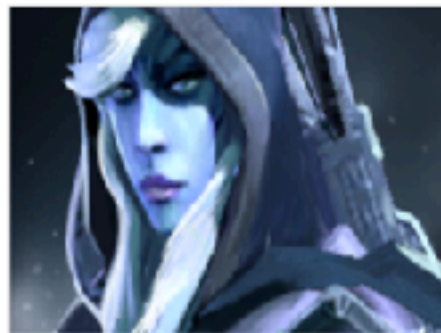
Bloodseeker

npc_dota_hero_bloodseeker



Crystal Maiden

npc_dota_hero_crystal_maiden



Drow Ranger

npc_dota_hero_drow_ranger



Earthshaker

npc_dota_hero_earthshaker



Juggernaut

npc_dota_hero_juggernaut



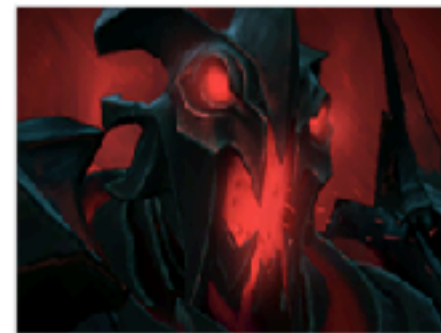
Mirana

npc_dota_hero_mirana



Morphling

npc_dota_hero_morphling



Shadow Fiend

npc_dota_hero_nevermore



Phantom Lancer

npc_dota_hero_phantom_lancer

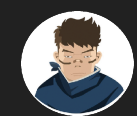
Blogs

- <https://ponyfoo.com/>
- <http://www.2ality.com/>
- <http://nodeweekly.com/>
- <http://javascriptweekly.com/>
- <https://blog.risingstack.com>
- <https://www.freecodecamp.com/>
- <https://www.babelcoder.com/>

References

- <https://angular.io/>
- <https://nodejs.org/en/>
- <https://hapijs.com/>
- <https://www.mongodb.com/>
- <http://exploringjs.com/>
- <https://babeljs.io/>

Thanks



Chai Phonbopit



DevAhoy