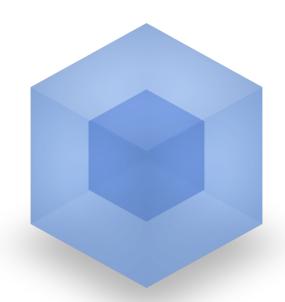
# Introduction to Webpack

Cerize @ Bananatag L&L





### **Development vs Production: expectation**



#### III, ORIENTAÇÕES GERAIS FARA O BOM ATENDORNIO

O descripcego é asia dis sasis grava atanção (no podea se vocaso da por um poso, a falta de cenda, a integrança sobre o Same e a terropia de futiros fogliase talcinalem descriptegados e interferen en sea custo de se oregado: e de se espesar fme se  $\gamma$ nomicate de posto de nomicanto se depute ala noma con tribalidas decimina : tristes, tras, também, initiates e se agrecios».

Em situações assim, um bem dia, um semio e un alto direntires ue clas de talacida. 🔾 podem alterar um pouco o seu transcr e fazer casa que ele ses mais compossaise é aux a atendimento. Ilso è tom para e traballador, que teri non profeste e a sonsi calcide e paro stendente, que poderá executor o sea trabalho con mais trasquisidale.

Todos es anos, o trabalho feito pelos arademes custivis par que militars de tabidados consigni inserção no mestado de todisão, sejan encantrasas por quilloção, promichanas de emprendefericas ou remonas solidas ou conjun caso a basilia de seguno-desemprego, em cossive: em que rão se possive) empreja a

On servições disposabilidados prios postas de aboxinação das paleiras de ejem a sua cos materializations upor this as SINE on, pole team, flar can car one safetime close o porto de una mantra difende da qui estanta ces sas oselas; e uni papasio sas s

O but underlyaj hebraj que artenquate que SU(0)e simpa o artenj<br/>aga e se ano a se dejation que irán enforeste.

#### after



Desirtuacies assert, on bornota, un sonica - Technius ananca, o trabalhe feit a pelos aton-



a emelhar distancina nociales de trata- distres (entrêto) para que milhanes de nutir order alterar in passo a sectour or trade it aboves consigning inserting and merstant process up not congresses and six trabation separt enconcertades. contracapatine de la aborque e tra- para qualificação, para investigas de amto hader que ses sus apopolítica e a contint. I precede dor simo ou, etizacionia solidária qui promote, equipo astardiete das poderáreies consiglam acesso ao beneficio do segunodesemprega, em ocacidos em que não foiproduct empression

De serviços disponétitivados pelos postas de standmento libra potenciali de alterar a vida des instruttradients que vila en SINS. ofarça e mais propinsário para os de-

#### 3.1. 0 Perfit Adequado para a Equipe SINE

A inagem do SINE, que um trabelhador ou um empregador que procuram um pesto de abensimento tenão, depende das instalações físicas do local em que foram atendidos e dia postura.



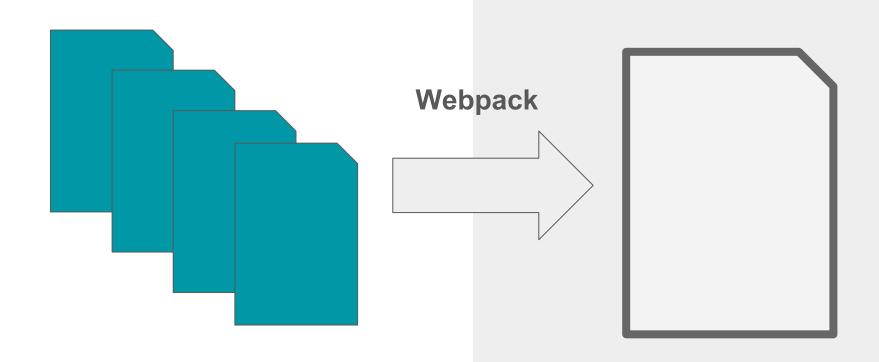


```
var greetings = 'Hello Bananatag!';
var n = 25200000;
var date = new Date('2016-09-29');
var dateAdjusted = new Date(date.getTime() + n);
console.log(greetings, 'Today is', dateAdjusted);
```

### Reality

The minified output (156 bytes, saved 45.07%

```
var greetings="Hello Bananatag!", n=252e5, date=new
Date("2016-09-29"), dateAdjusted =new
Date(date.getTime()+n); console.log(greetings, "Today is", dateAdjusted);
```



#### **Motivation**

- Fact 1: More and more code on the client side.... A big code base needs to be organized...
- Fact 2: To be manageable, large applications must be separated in parts
- **Fact 3**: Module systems offer the option to split your code base into modules.
- **Fact 4**: Module is the main **Javascript** design pattern

```
// CommonJs

require("module");
require("../file.js");
exports.doStuff = function()
{};
```

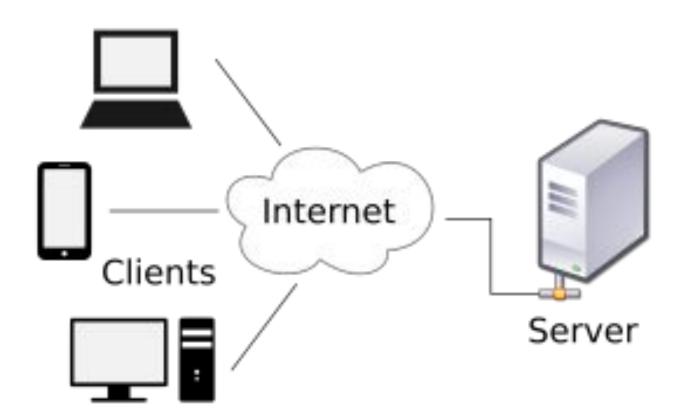
```
// ES6

export const sqrt = Math.sqrt;
export function square(x) {
    return x * x;
}
export function diag(x, y) {
    return sqrt(square(x) +
square(y));
```

#### **Motivation**

**Fact 4**: Until ES6, JavaScript did not have built-in support for modules, which means: NO MODULES IN THE BROWSER FOR YOU: (





# Webpack

Packs CommonJs/AMD modules for the browser.





Allows to split your codebase into multiple bundles, which can be loaded on demand.

Support loaders to preprocess files, i.e. json, jsx, es7, css, less, ... and your custom stuff.



It was authored by Tobias Koppers @sokra on Mar, 2012.

```
cats.js
                                                    webpack reads the entry point and analyzes
  var cats = ['dave', 'henry', 'martha'];
                                                    its dependencies, its dependencies'
  module.exports = cats;
                                                    dependencies, and so on.
app.js
  var cats = require('./cats.js');
  console.log(cats);
                                                                          webpack
                                                                               MODULE BUNDLER
                                               app.bundle.js
      webpack bundles the entry point
      and all its dependencies into a
                                                 !function(r){function t(o){if(n[o])return n[o].exports;
      single file.
                                                 var e=n[o]=\{i:o,l:!1,exports:\{\}\};return r[o].call(
                                                 e.exports,e,e.exports,t),e.l=!0,e.exports}var n={};
                                                 return t.m=r,t.c=n,t.p="",t(t.s=1)}([function(r,t){
                                                 var n=["dave","henry","martha"];r.exports=n},function
                                                 (r,t,n){cats=n(0),console.log(cats)}]);
```

#### Is that all?

- Preloaders
- Loaders
  - Can be chained (pipeline)
  - Only the final load must return js
  - Can be bound to Regex/extensions
  - Examples: **babel**, **css**, **sass**, **flow**
- Plugins
- SourceMaps
- Code Splitting



https://webpack.github.io/docs/list-of-loaders.html

#### **Installation and Use**

```
$ npm install webpack -g
$ npm install webpack-dev-server --save-dev
webpack ./app.js bundle.js
```

## webpack.config.js

```
var webpack = require('webpack');
var path = require('path');
var config = require('./configs/config');
module.exports = {
  entry: [
     path.join( dirname, 'web', 'src', 'js', 'client.js'),
     // path.join( dirname, 'web', 'static', 'css', 'style.css')
  output: {
     path: path.join( dirname, 'web', 'static', 'dist'),
     filename: 'bundle.js'
  resolve: {
     extensions: [", '.js', '.jsx']
  },
```

```
module: {
     loaders: [
           test: /.jsx?$/,
           loader: 'babel',
           exclude: /node_modules/
           test: /\.json$/, loader: 'json'
        },
        // {
             test: /\.css$/, loader: 'style!css?sourceMap'
        // },
        // {
             test: /\.s(a|c)ss$/, loader: 'style!css?sourceMap!sass?indentedSyntax'
        // },
        // {
             test: \(\lambda.\)(png|jpg)\(\struct\), loader: 'url?limit=8192' // inline base64 URLs for <=8k images,
direct URLs for the rest
        // }
```

```
plugins: [
  new webpack.DefinePlugin({
     'process.env': {
       // This has effect on the react lib size
       'NODE ENV': JSON.stringify('production'),
       'API SERVER': JSON.stringify('http://' + config.apiHost + ':' + config.apiPort)
  }),
  new webpack.optimize.OccurenceOrderPlugin(),
  new webpack.optimize.DedupePlugin(),
  new webpack.optimize.UglifyJsPlugin({
    minimize: true,
     warning: false,
     compress: {
       warnings: false,
       dead code: true,
       loops: true,
       unused: true,
       if return: true,
       join vars: true
```

#### References

https://webpack.github.io/docs/

https://medium.com/@dabit3/beginner-s-guide-to-webpack-b1f1a3638460#.73w1zc23

V

http://survivejs.com/webpack/developing-with-webpack/

#### My notes

CommonJS is a project with the goal of specifying an ecosystem for JavaScript outside the browser

The dev server uses Webpack's watch mode. It also prevents webpack from emitting the resulting files to disk. Instead it keeps and serves the resulting files from memory.

With Webpack dev server running, you will notice that if you go back to your app and make a change, the browser will automatically refresh (hot-loading).