

The image features a dark, almost black background with several thin, white, parallel lines. These lines are oriented diagonally, creating a sense of movement and depth. The lines are scattered across the frame, with some appearing as single strokes and others as pairs or groups. The overall aesthetic is minimalist and modern.

WEBPACK

CHEN ZHEN

目录 • WEBPACK

1	简介
2	配置
3	WEBPACK和前端工程化



1

简介

什么是WEBPACK

WEBPACK的优势

WEBPACK的安装

从一个简单示例看WEBPACK

是什么

近期最火的一款模块加载器兼打包工具

加载顺序不好把握；
需要自己梳理可能的冲突和
依赖

`<script>`

实现一小部分；
所有浏览器支持需要时间

ES6 modules

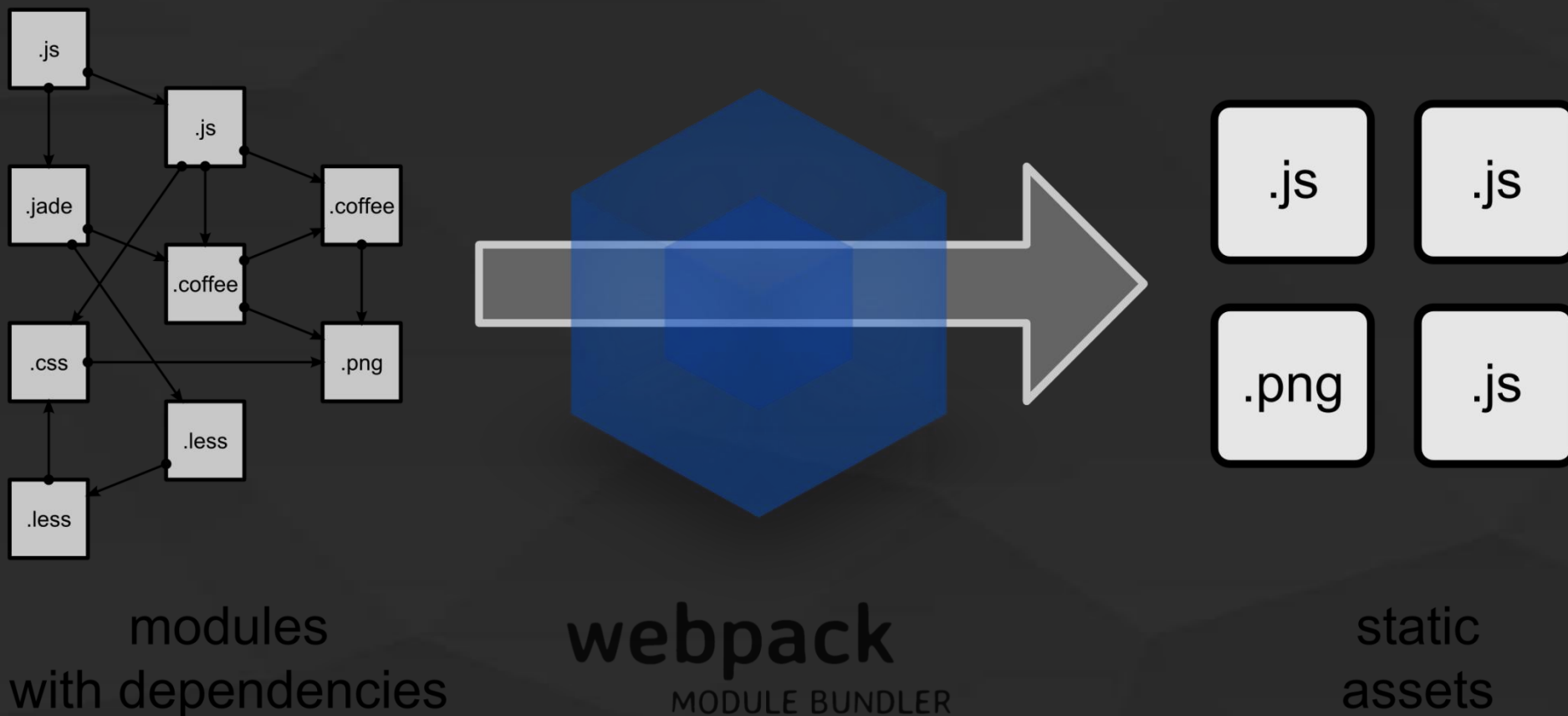
Commonjs/AMD

在浏览器端，网络的请求都是异步的，无法并行的
require多个module

webpack

灵活的模块管理；
还有很多其他**优势**.....

是什么



webpack处理后，输出的静态文件只剩下 **js** 与 **png**，而css、less、jade其他的文件都合并到了js中。可以根据需求将一些图片自动转成base64编码的png文件；也可以打包成js文件。看设置~

优势

1

它同时支持commonjs和AMD规范（甚至混合的形式），方便系统进行旧代码迁移；

2

能模块化各种资源：例如JS（含JSX）、coffee、样式（含less/sass）、图片等；

3

Loaders可以使文件在编译时得到预处理：比如说模板的预编译，图片的base64处理；

4

丰富的和可扩展的插件可以适应多变的需求：例如支持React热插拔（见[react-hot-loader](#)）；

5

其他：如按需打包和按需加载、可灵活配置打包各个环节等。

安装

1

安装node.js环境 + npm

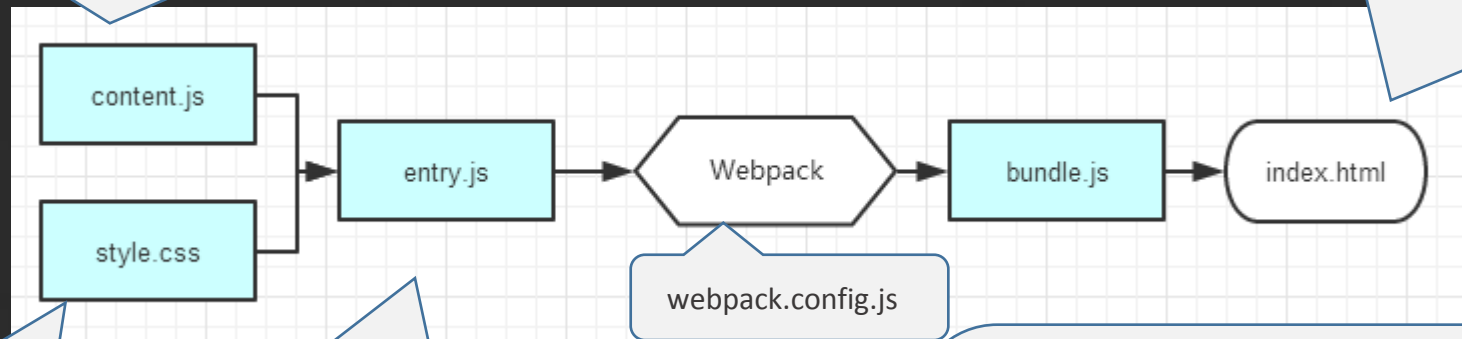
2

- `npm install webpack -g`
- 在项目中，把依赖写入package.json：
 `npm init`（创建package.json）
 `npm install webpack --save-dev`
（添加并下载依赖到node_modules文件）

——从一个简单示例看WEBPACK——

```
<html>
<head>
  <meta charset="utf-8">
</head>
<body>
  <script type="text/javascript" src="bundle.js" charset="utf-8"></script>
</body>
</html>
```

```
module.exports = "It works from content.js.";
```



```
body {
  background-color: yellow;
}
```

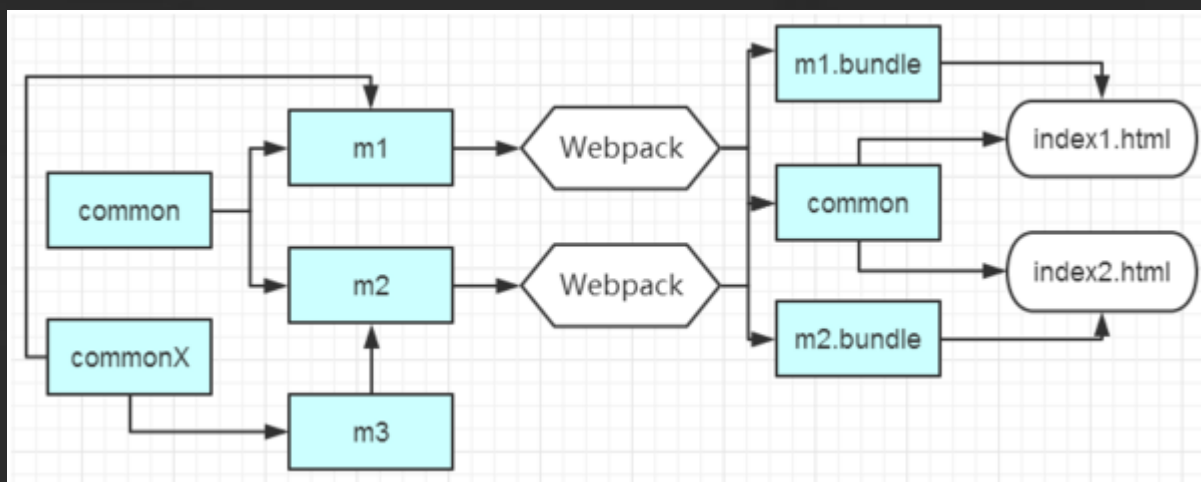
```
//样式文件同样以模块方式引入
require("!style!css!./style.css");
//以CMD引入content.js
var content = require("./content.js");

function a() {
  document.write(content);
};
a();
```

```
module.exports = {
  entry: "./entry.js",
  output: {
    path: __dirname,
    //打包输出文件
    filename: "bundle.js"
  },
  module: {
    //loaders引入加载器
    loaders: [
      { test: /\.css$/, loader: "style!css" }
    ]
  }
};
```


——从一个简单示例看WEBPACK——

通过设计和配置，可以灵活切分代码，实现最优的按需打包和按需加载



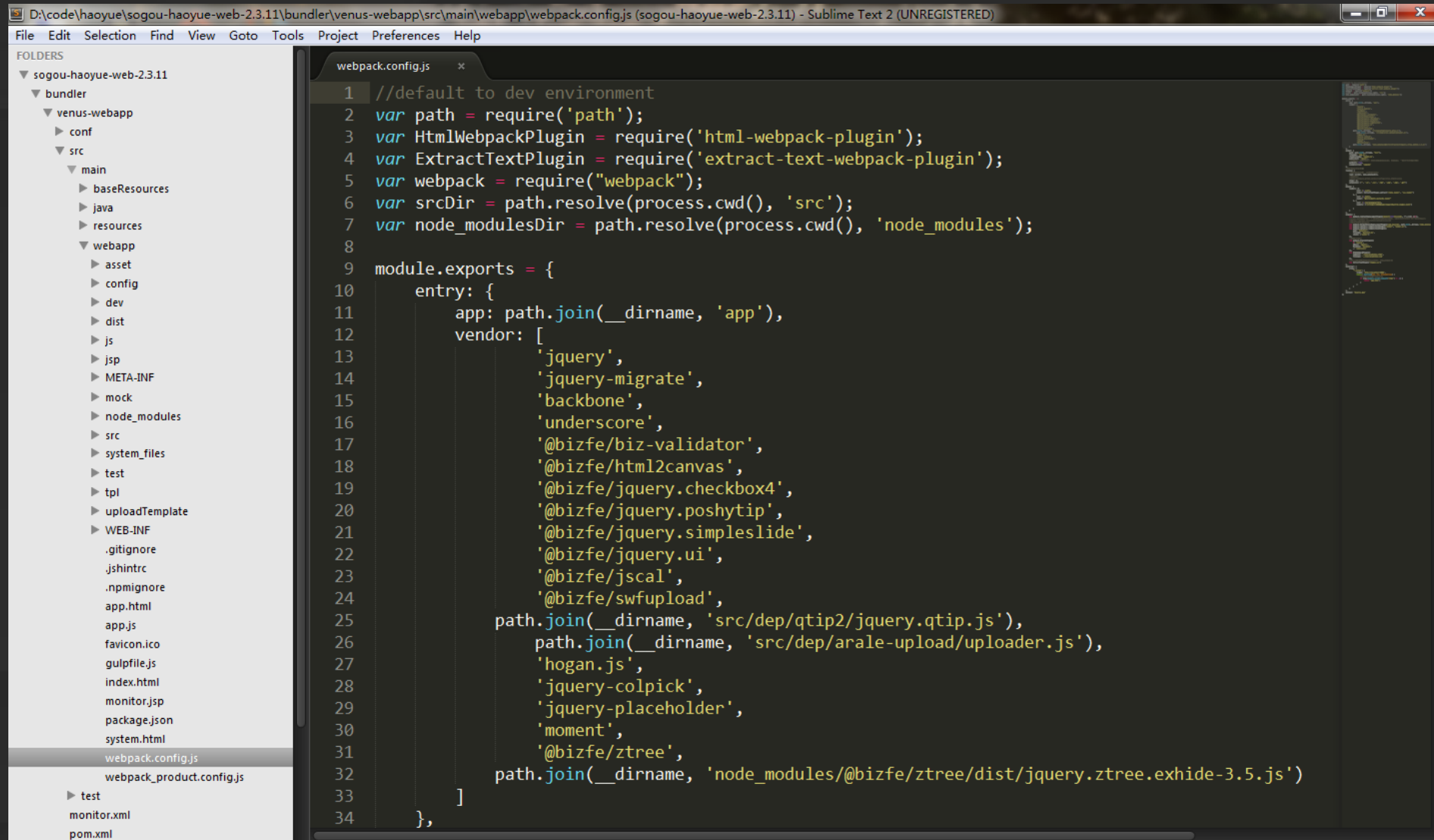


2

配置

entry
output
module
resolve
plugin
devServer
devtool

webpack.config.js



```
D:\code\haoyue\sogou-haoyue-web-2.3.11\bundler\venus-webapp\src\main\webapp\webpack.config.js (sogou-haoyue-web-2.3.11) - Sublime Text 2 (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

FOLDERS
▼ sogou-haoyue-web-2.3.11
  ▼ bundler
    ▼ venus-webapp
      ► conf
      ▼ src
        ▼ main
          ► baseResources
          ► java
          ► resources
          ▼ webapp
            ► asset
            ► config
            ► dev
            ► dist
            ► js
            ► jsp
            ► META-INF
            ► mock
            ► node_modules
            ► src
            ► system_files
            ► test
            ► tpl
            ► uploadTemplate
            ► WEB-INF
            .gitignore
            jshintrc
            .npmignore
            app.html
            app.js
            favicon.ico
            gulpfile.js
            index.html
            monitor.jsp
            package.json
            system.html
            webpack.config.js
            webpack_product.config.js
          ► test
            monitor.xml
            pom.xml

webpack.config.js
1 //default to dev environment
2 var path = require('path');
3 var HtmlWebpackPlugin = require('html-webpack-plugin');
4 var ExtractTextPlugin = require('extract-text-webpack-plugin');
5 var webpack = require("webpack");
6 var srcDir = path.resolve(process.cwd(), 'src');
7 var node_modulesDir = path.resolve(process.cwd(), 'node_modules');
8
9 module.exports = {
10   entry: {
11     app: path.join(__dirname, 'app'),
12     vendor: [
13       'jquery',
14       'jquery-migrate',
15       'backbone',
16       'underscore',
17       '@bizfe/biz-validator',
18       '@bizfe/html2canvas',
19       '@bizfe/jquery.checkbox4',
20       '@bizfe/jquery.poshytip',
21       '@bizfe/jquery.simpleslide',
22       '@bizfe/jquery.ui',
23       '@bizfe/jscal',
24       '@bizfe/swfupload',
25       path.join(__dirname, 'src/dep/qtip2/jquery.qtip.js'),
26       path.join(__dirname, 'src/dep/arale-upload/uploader.js'),
27       'hogan.js',
28       'jquery-colpick',
29       'jquery-placeholder',
30       'moment',
31       '@bizfe/ztree',
32       path.join(__dirname, 'node_modules/@bizfe/ztree/dist/jquery.ztree.exhide-3.5.js')
33     ]
34   },
}
```

entry

—— 模块的入口

字符串

在启动的时候加载

```
entry: './js/main.js'
```

数组

这个数组内所有模块会在启动的时候加载，数组的最后一个元素作为export

```
entry: [  
  'webpack/hot/only-dev-server',  
  './js/app.js'  
]
```

对象

可以构建多个bundle，实现按需使用

```
entry: {  
  hello: './js/hello.js',  
  form: './js/form.js'  
}
```

output

用于定义构建后的文件的输出：

```
output: {
  path: path.join(__dirname, "dist"),
  filename: 'app.js',
  chunkFilename: '[name].js',
  publicPath: '/dist/',
  //在模块id（通常是数字）前添加模块路径的注释，帮助阅读，一般在开发环境中需要
  pathinfo: true,
  jsonpFunction: 'sogouhy'
},
```

- path：输出文件地址，要求是绝对路径
- filename：用来配置生成的文件名。当我们在entry中定义构建多个文件时，filename可以对应的更改为[name].js用于定义不同文件构建后的名字。
- chunkFilename：未被列在entry中，却又需要被打包出来的文件命名配置。使用场景是：在按需加载（异步）模块的时候，文件是没有被列在entry中的，这时生成的文件名需要以chunkname配置
- publicPath：在浏览器中引用时，指定输出文件的公共网址地址
- pathinfo：是否支持pathinfo模式。pathinfo模式是php中的功能，返回文件路径的信息。设置为true时，支持这种写法：require(/* ./test */23)。默认为false
- jsonpFunction：异步加载chunks。默认是"webpackJsonp"

module

关于模块相关的配置：

```
module: {
  loaders: [{
    test: /\.css$/i,
    loader: ExtractTextPlugin.extract('style-loader', 'css-loader')
  }, {
    test: /\.tpl$/i,
    loader: "@bizfe/multi-mustache-loader"
  }, {
    test: /\.(jpe?g|png|gif)$/i,
    loaders: ['url?limit=10000&name=image/[hash:8].[name].[ext]']
  }
]
},
```

- loaders：为不同文件配置加载器
- preLoaders、postLoaders：语法同loaders，前置或者后置加载器
- noParse：在webpack中忽略对已知文件的解析。如果你确定一个模块中没有其它新的依赖，就可以配置这项，webpack 将不再扫描这个文件中的依赖。

resolve

解析模块的相关配置：

```
resolve: {  
  //解析查找的目录。必须是绝对路径  
  root: [srcDir, node_modulesDir],  
  //模块别名  
  //http://webpack.github.io/docs/configuration.html#resolve  
  alias: {},  
  extensions: ['', '.js', '.css', '.tpl', '.png', '.jpg', '.gif']  
},
```

- root：解析模块的路径，要求绝对路径
- alias：模块别名。把用户的一个请求重定向到另一个路径。不是最佳实践，故皓月晨星不使用该功能
- extensions：配置程序可以自行补全哪些文件后缀

plugin

webpack提供了丰富的组件用来满足不同的需求，
也支持自行实现一个组件来满足需求：

```
plugins: [  
  new webpack.ContextReplacementPlugin(/moment[\\/]locale$/, /^\.\/zh-cn$/),  
  //arale-upload包引用了一个其他更名过的jquery版本，造成打包时打进去了两个版本的jquery  
  //#https://github.com/aralejs/upload/issues/49  
  //因此对该版本的jquery  
  new webpack.NormalModuleReplacementPlugin(/spm-jquery$/, path.join(__dirname, 'node_modules/jquery/dist/jquery')),  
  new webpack.optimize.CommonsChunkPlugin("vendor", "vendor.js"),  
  new webpack.optimize.CommonsChunkPlugin({  
    name: 'manifest',  
    filename: 'manifest.js',  
    chunks: ['vendor']  
  }),  
  //暴露全局接口  
  new webpack.ProvidePlugin({  
    $: 'jquery',  
    jQuery: 'jquery',  
    Backbone: 'backbone',  
    _: 'underscore'  
  }),  
  new HtmlWebpackPlugin({  
    template: './jsp/cpcadindex.html',  
    filename: '../jsp/cpcadindex.jsp'  
  }),  
  //把css模块提取到独立的css文件中，在head标签引用  
  new ExtractTextPlugin('[name].css')  
],
```

[List of plugins](#)

devServer

可用来配置webpack-dev-server行为

```
devServer: {  
  proxy: {  
    '*.action': {  
      target: 'http://localhost:8888',  
      bypass: function(req, res, proxyOptions) {  
        //处理jsp页面默认的动作  
        if (req.headers.accept.indexOf('html') !== -1) {  
          return 'app.html';  
        }  
      }  
    }  
  }  
},
```

devtool

开发调试工具：

```
devtool: '#source-map'
```

- 即使在开发时，我们也是加载webpack打包后的文件，通过sourcemap可以对打包后的代码进行调试。
- source-map是最原始的 source-map 实现方式，它在打包代码同时创建一个新的 sourcemap 文件，并在打包文件的末尾添加 `//# sourceMappingURL` 注释行告诉 JS 引擎文件在哪儿。



WEBPACK和前端工程化

各产品线开发流程和架构
工程化方案

将webpack用到前端工程化中
webpack和webpack-dev-server

工程化方案

Npm私服 : CNPM

Mock server : bizmock + webpack-dev-server

脚手架工具 : BizDP

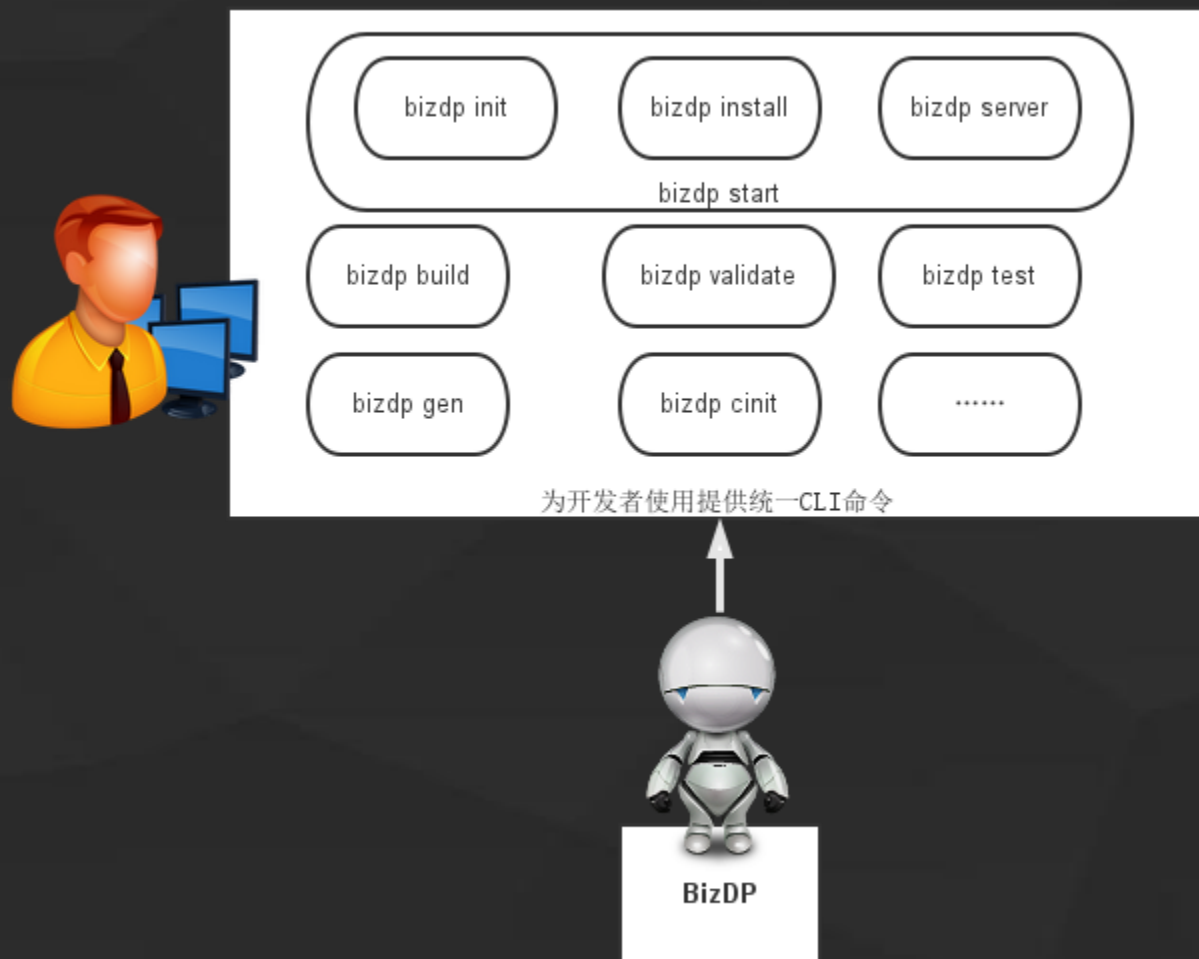
模块化规范 : commonJS

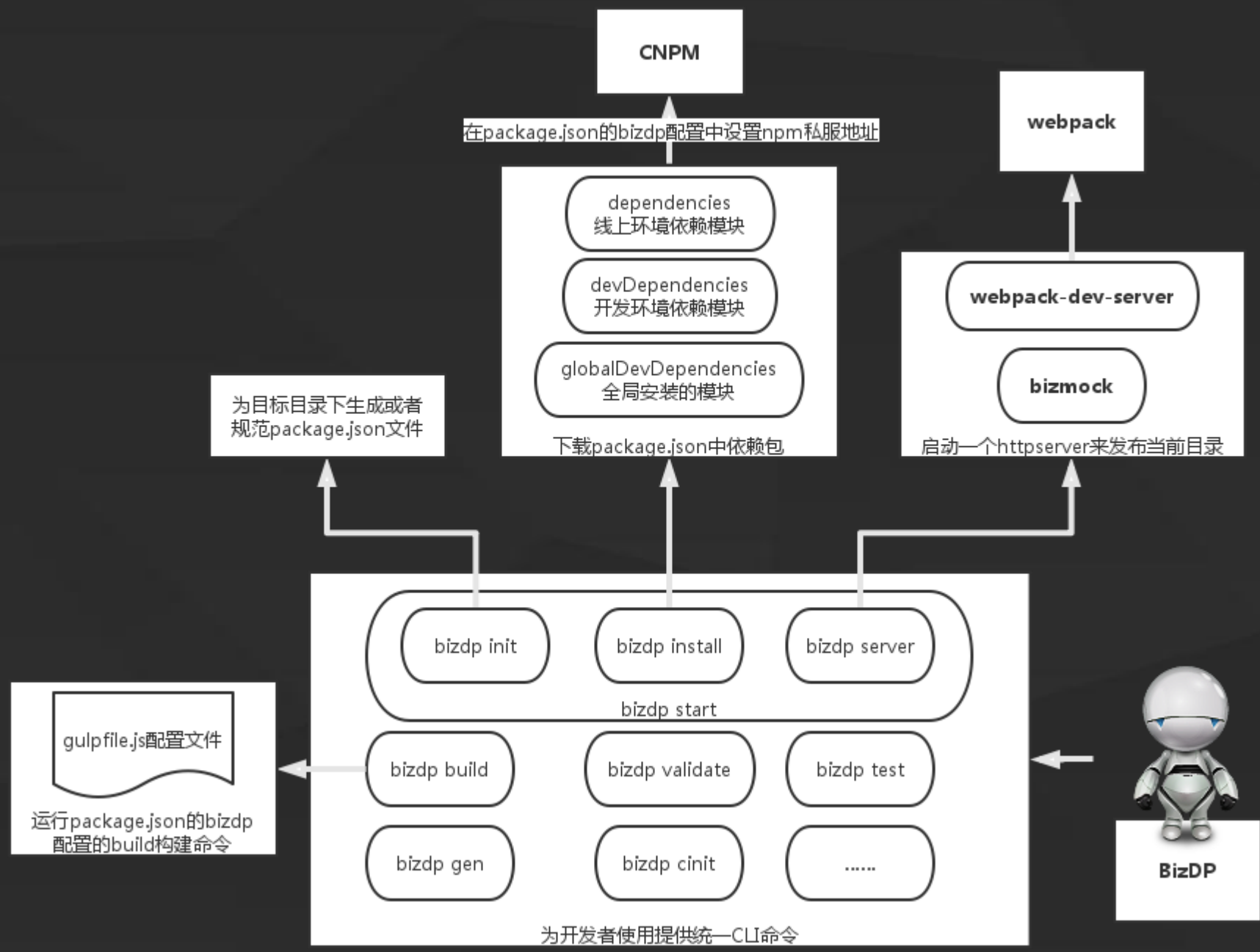
构建体系 : Webpack



NodeJs

将webpack用到前端工程化中





webpack和webpack-dev-server

```
"bizdp": {
  "npm": {
    "command": "npm",
    "registry": "http://10.134.27.134:7001"
  },
  "build": "gulp",
  "server": {
    "command": "webpack-dev-server",
    "options": {
      "hot": true,
      "color": true,
      "inline": true,
      "progress": true
    }
  },
  "validate": {
    "command": "jshint",
    "options": {},
    "src": "src"
  },
  "mock": {
    "port": 8888
  }
},
```

```
D:\code\haoyue\sogou-haoyue-web-2.3.11\bundler\venus-webapp\src\main\webapp>bizdp server
D:\code\haoyue\sogou-haoyue-web-2.3.11\bundler\venus-webapp\src\main\webapp
biz-mock is running, listening to port 8888
8176,8129,8138,8157,8192,8140,8155
[ 8176, 8129, 8138, 8157, 8192, 8140, 8155, 8121 ]
webpack-dev-server --hot --inline --progress --color --port 8121 --mockport 8888
 70% 2/2 build moduleshttp://localhost:8121/
webpack result is served from /dist/
content is served from D:\code\haoyue\sogou-haoyue-web-2.3.11\bundler\venus-webapp\src\main\webapp
 54% 34/46 build moduleswebpack: wait until bundle finished: /dist/app.css
webpack: wait until bundle finished: /dist/manifest.js
webpack: wait until bundle finished: /dist/vendor.js
 52% 47/66 build moduleswebpack: wait until bundle finished: /dist/app.js
Hash: 20384d8f3e7f71e18a93
Version: webpack 1.12.14
Time: 9984ms

   Asset      Size  Chunks             Chunk Names
manifest.js   24.1 kB    4  [emitted]  manifest
image/79d999e5.brushed-steel.jpg 17.1 kB    [emitted]
image/3eddf82d.ie6_bg.png    63.5 kB    [emitted]
image/2c6a035c.checker_bg.png   16 kB    [emitted]
image/524f504c.zTreeStandard.png 11.2 kB    [emitted]
app.js        977 kB    0, 4  [emitted]  app
ideaPub.js   395 kB    1, 4  [emitted]  ideaPub
report.js    61.8 kB    2, 4  [emitted]  report
vendor.js    1.83 MB    3, 4  [emitted]  vendor
image/16ff00a2.Generic_icon.png 12.7 kB    [emitted]
app.css       309 kB    0, 4  [emitted]  app
app.js.map   1.25 MB    0, 4  [emitted]  app
app.css.map   84 bytes  0, 4  [emitted]  app
ideaPub.js.map 499 kB    1, 4  [emitted]  ideaPub
report.js.map  82 kB     2, 4  [emitted]  report
vendor.js.map 2.2 MB    3, 4  [emitted]  vendor
manifest.js.map 25.6 kB    4  [emitted]  manifest
../jsp/cpcadindex.jsp 1.62 kB    [emitted]

chunk    <0>  app.js, app.css, app.js.map, app.css.map <app> 929 kB <3> [rendered]
```

webpack和webpack-dev-server

webpack-dev-server :

- ◆ 一个轻量的node.js express服务器；
- ◆ webpack官方提供的一个辅助开发工具；
- ◆ 可以自动监控项目下的文件，一旦有修改保存操作，开发服务器就会自动运行webpack 打包命令，帮我们自动将开发的代码重新打包。而且，如果需要的话，还能自动刷新浏览器，重新加载资源。

webpack和webpack-dev-server

```
webpack-dev-server --inline --hot
```

Hot Modules Replacemant —— 模块热部署：

修改后的代码不仅可以自动打包，而且浏览器不用完全刷新，只需要异步刷新，加载修改后部分代码即可，加载完成效果会马上反应在页面效果上。

欲知后事

且听下回分解