

Application Angular Modulaire



Vincent Elcrin

- Twitter - @VincentElcrin
- Github - velcrin
- Email - vincent.elcrin@bonitasoft.com

Application WEB



Larges Applications

- Complexité
- Réutilisabilité
- Maintenabilité
- Performance

Modules



Modules

- Organise le code
- Facilite lisibilité
- Facilite testabilité
- Controle la complexité



ES2015



ES2015 - Modules

*“Large ECMAScript programs are supported by modules which allow a program to be **divided into multiple sequences of statements and declarations**. **Each module explicitly identifies declarations** it uses that need to be provided by other modules and which of its **declarations are available for use by other modules**.”*

ES2015 - Modules

```
export default myFunctionOrClass;
```

```
import { member as alias } from "module-name";
```

```
* System.import("module-name").then(function() {});
```

<https://whatwg.github.io/loader/>

Des solutions aujourd'hui



webpack
MODULE BUNDLER

EcmaScript 5 Modules

- CommonJS - NodeJS
- Asynchronous Module Definition - AMD

Modules Angular 1.x - Limites

helloModule.js

```
angular.module('helloService', []);
```

helloFactory.js

```
angular.module('helloModule').factory('helloService', function ($log) {  
    'use strict';  
  
    this.sayHello = function () {  
        $log.log('Hello');  
    };  
});
```

Modules Angular 1.x - Limites

- Encapsulation sommaire
- Conflits de nommage
- Pas de dépendance asynchrone
- Syntaxe verbeuse et inexpressive
- Système de build complexe



webpack
MODULE BUNDLER

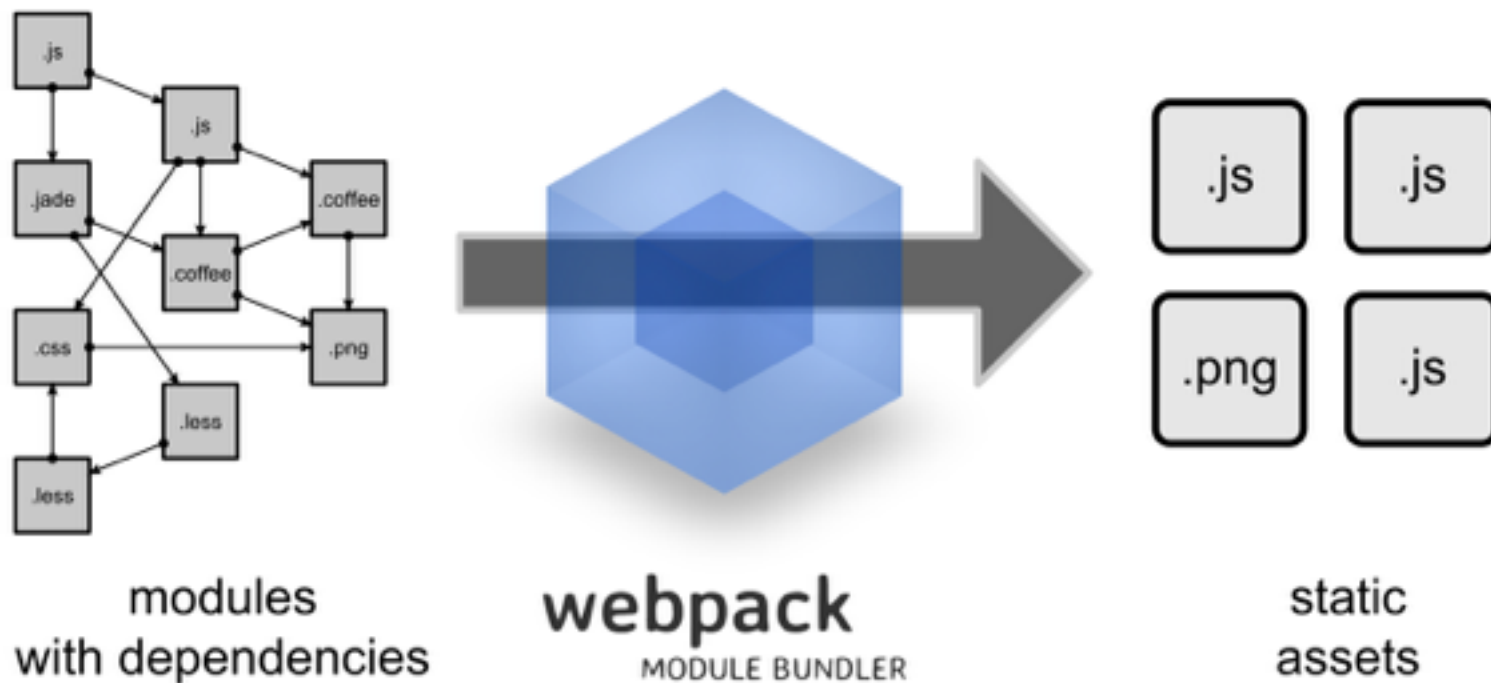


ES2015 - Modules



***“WebPack is a bundler for modules.** The main purpose is to bundle JavaScript files for usage in a browser, yet it is also capable of transforming, bundling, or packaging just about any resource or asset.”*

ES2015 - Modules



Features

- Entry Points
- Code Splitting
- Loaders
- Plugins

Modules Angular 1.x - Limites

helloModule.js

```
import angular from 'angular';
import HelloService from 'helloService.js';

angular.module('helloModule', []).service('helloService', HelloService);
```

helloService.js

```
export default class HelloService {

  constructor($log) {
    this.$log = $log;
  }

  sayHello() {
    this.$log.log('Hello');
  }
}
```

DEMO WebPack



Angular 1.x & WebPack

- Réelle encapsulation - [export](#)
- Syntax concise et explicite
- Code split & lazy loading via ocLazyLoad

<https://oclazyload.readme.io/docs/webpack>

WebPack



Angular 2

- `.module()` n'existe plus
- Basé sur la spécification ES2015
- `System.import().then()`

References

- <http://dmachat.github.io/angular-webpack-cookbook/index.html>
- <https://github.com/petehunt/webpack-howto>
- <https://github.com/velcrin/webpack-angular-es2015>



Merci!



Follow us on twitter
@bonitasoft



Visit us online
bonitasoft.com



Join our community
Bonitasoft.org



Download
Bonitasoft.com/downloads