


Angular 4

architecture

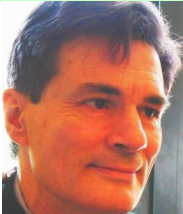
vertika.org by Jean Garutti

-- Angular 4 --

03/09/2017 -- 1



Angular 4 -- architecture



- *Pour élaborer votre application*
- *vous devez connaître l'architecture d'Angular*

Jean Garutti – vertika.org

by Jean Garutti

-- Angular 4 --

-- 2



Vous coupez tout en petits morceaux

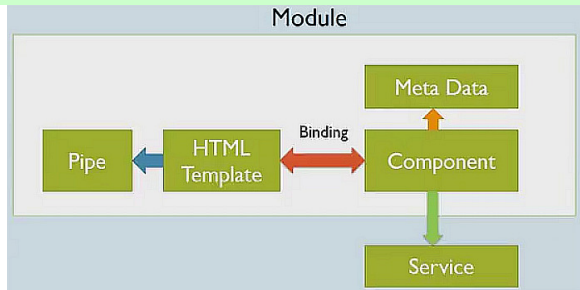


by Jean Garutti

-- Angular 4 --

-- 3

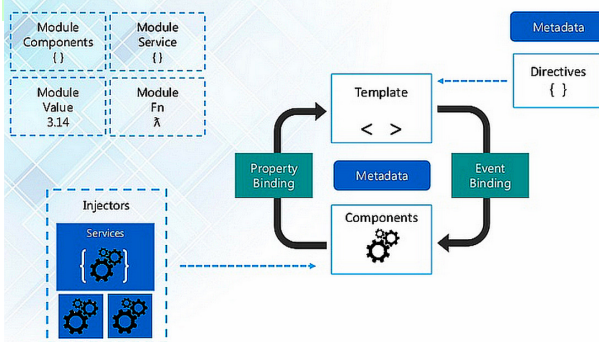
ANGULAR, des Modules....



Composant, est-ce utile ?

- lors de l'élaboration de votre application
- vous passerez la majeure partie du temps dans la construction des *composants*.

ANGULAR, Architecture



info@vertika.org

ANGULAR, 7 blocs...

Building Blocks of Angular

The diagram shows a blue background with white rounded rectangles. At the top, five boxes are arranged horizontally: 'Module', 'Component', 'Metadata', 'Template', and 'Data Binding'. Below these, two more boxes, 'Services' and 'Directives', are arranged horizontally and are slightly offset to the left and right respectively, creating a staggered effect.

by Jean Garutti -- Angular 4 -- -- 7

info@vertika.org

Composant, qu'est-ce que c'est ?

- Les composants Angular 4
- sont simplement des **classes**
- qui sont désignées comme « composant » par un **« décorateur de composants »**.

by Jean Garutti -- Angular 4 -- -- 8

info@vertika.org

Composant, qu'est-ce que c'est ?

- Chaque composant possède un **template** défini qui peut communiquer avec le code décrit dans la classe de composant.

by Jean Garutti -- Angular 4 -- -- 9

Template, qu'est-ce que c'est ?

- Vos vues sont définies dans les *modèles* HTML.
- Ces **Templates** sont définis dans le **@Component decorator**.
- Vous pouvez définir des *modèles* HTML
 - ☐ en ligne
 - ☐ externes (dans des fichiers HTML).

Template, qu'est-ce que c'est ?

- Vous pouvez afficher les données définies dans le composant
 - ☐ par interpolation,
- Vous pouvez utiliser divers conditionnements dans le modèle.

Composant, est structuré comment ?

```

1. // Imports
2. import { Component } from '@angular/core';
3. // Component Decorator
4. @Component({
5.   selector: 'app-root',
6.   1. templateUrl: './app.component.html',
     styleUrls: ['./app.component.css']
7. })
8. // Component Class
9. export class AppComponent {
10.   title = 'app works!';
11. }

```

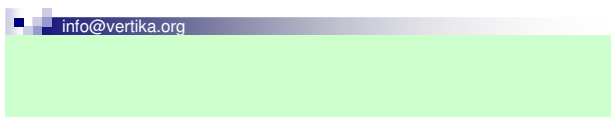
```
info@vertika.org
import { Component }
from '@angular/core';
```

Pour créer (une classe d')
un composant
Il faut importer le membre
Component
à partir de la bibliothèque
@angular/core.

```
info@vertika.org
import { Component }
from '@angular/core';
```

- Certains composants peuvent avoir
- **des dizaines d'importations**
- en fonction de leurs besoins.

- Vous importez tous les **services**
- que votre composant peut utiliser
- via l'**injection de dépendance**.



Merci !

Thanks !
