

Angular 4 expressions de template

Règles à respecter

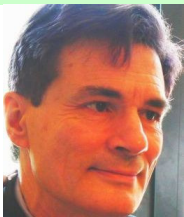
vertika.org by Jean Garutti

-- Angular 4 --

03/09/2017 -- 1

info@vertika.org

Expressions de template



- *Respectez les règles*
- *et Angular vous respectera*

Jean Garutti – vertika.org

by Jean Garutti

-- Angular 4 --

-- 2

info@vertika.org

Respectez les règles de la tribu



by Jean Garutti

-- Angular 4 --

-- 3

Aucun effet secondaire visible

- Une expression de **template** ne doit modifier aucun état de l'application
 - autre que la valeur de la propriété cible.

by Jean Garutti

-- Angular 4 --

-- 4

Aucun effet secondaire visible

- Cette règle est essentielle à la politique de «flux de données unidirectionnelles» d'Angular.
 - la lecture d'une valeur de composant ne doit pas modifier une autre valeur affichée.
 - la vue doit rester *stable* tout au long d'une seule passe de rendu.

by Jean Garutti

-- Angular 4 --

-- 5

Exécution rapide

- Angular exécute des expressions de **template**
- après chaque cycle de *détection d'un changement*.

by Jean Garutti

-- Angular 4 --

-- 6

Exécution rapide

- Les cycles de détection des changements sont déclenchés par de nombreuses activités asynchrones telles que
 - les résolutions de promesses,
 - les résultats http,
 - les événements temporisés,
 - les pressions de touches
 - et les mouvements de la souris

by Jean Garutti

-- Angular 4 --

-- 7

Exécution rapide

- Les expressions doivent se terminer rapidement
 - sinon l'utilisateur peut se lasser,
 - en particulier sur les périphériques plus lents.

by Jean Garutti

-- Angular 4 --

-- 8

Exécution rapide

- Utilisez les valeurs mises en cache
 - lorsque leur calcul coûte cher.

by Jean Garutti

-- Angular 4 --

-- 9

Simplicité

- Il est possible d'écrire des expressions de **template** assez complexes, mais vous devez les éviter.

Un nom de propriété ou un appel de méthode doit être la norme.

Simplicité

- Une négation booléenne *occasionnelle* (!) est acceptable.
 - ☐ sinon, confinez l'application et la logique métier dans le composant lui-même,
 - ☐ où il sera plus facile à développer et à tester.

Idempotence

- Une expression *idempotente* renvoie toujours exactement la même chose,
 - ☐ jusqu'à ce que l'une de ses valeurs dépendantes change.

Idempotence

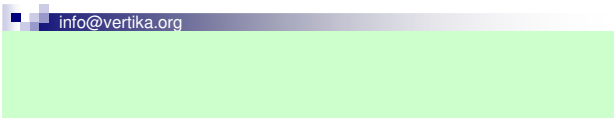
- Une expression *idempotente* est idéale
 - car elle est exempte d'effets secondaires
 - et améliore les performances de détection de changement d'Angular.

Idempotence

- Les valeurs dépendantes ne doivent pas changer
 - pendant un tour de la boucle d'événement.

Idempotence

- Si une expression idempotente renvoie une chaîne ou un nombre,
 - elle renvoie la même chaîne ou le même nombre
 - lorsqu'elle est appelée deux fois de suite.
- Si l'expression renvoie un objet (y compris un tableau),
 - il renvoie la même référence d'objet lorsqu'il est appelé deux fois de suite.



Merci !

Thanks !
