

Développement d'un outil analysant du code Angular (TypeScript)

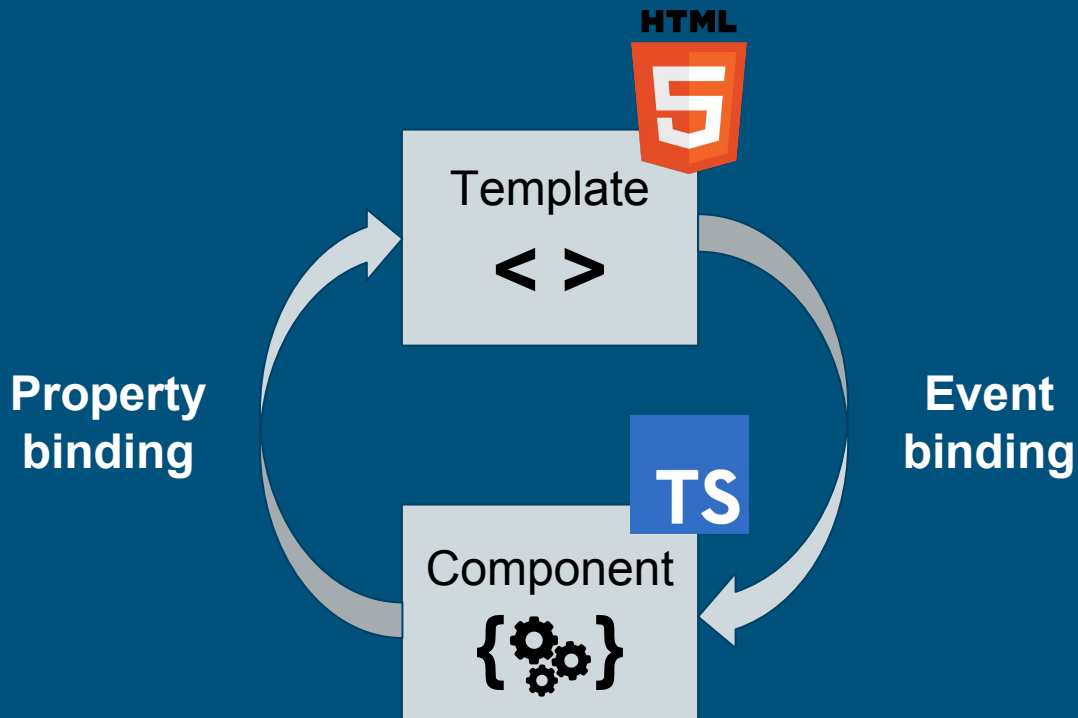
Projet Test Logiciel

Alexis BUSSENEAU
Grégoire LINOT
Mohamed RACHID
Adrien THÉBAULT

Présentation d'Angular (4.0)

- ❑ Framework JS open-source développé par Google
- ❑ Utilisé pour la partie “front-end” des applications web
- ❑ Basé sur du TypeScript (sur-ensemble de JavaScript)
- ❑ Repose sur le patron de conception MV* ou MVW (Model-View-Whatever)

Fonctionnement d'Angular



Problématique

- ❑ Détection des fonctions et attributs de la vue
- ❑ Différencier les données entrantes et sortantes entre les composantes
- ❑ Comment représenter l'architecture de l'application ?

Solution : Analyser le code statiquement

Objectifs

- ❑ Développer un outil d'analyse d'une application Angular
 - ❑ Analyser les “components”
 - ❑ Analyser le “template”
 - ❑ Détecter les bugs potentiels avant l'exécution

Réalisation n°1 : Analyser les composants

- ❑ Recherches sur le fonctionnement d'Angular
- ❑ Analyse de l'API du compilateur TypeScript
- ❑ Création d'un arbre syntaxique abstrait (AST)
- ❑ Analyse de l'arbre pour récupérer les données

Réalisation n°2 : Analyser le “template”

- ❑ Récupération du DOM du *template*
- ❑ Création d'expressions régulières pour matcher les tags
- ❑ Tests sur plusieurs applications trouvées sur internet

Réalisation n°3 : Détection des erreurs

- ❑ Comparer les attributs / fonctions utilisés entre le *template* et le *composant* associé
- ❑ Afficher à l'utilisateur les potentielles erreurs
- ❑ Représenter la structure des objets Angular

Difficultés rencontrées

- ❑ Manque de documentations de la part de Microsoft
- ❑ Mauvaise qualité du code d'exemple fourni
- ❑ Problème du typage facultatif de TypeScript - **any**
- ❑ Parcours des tableaux - ***ngFor**



Démonstration

Conclusion

- ❑ Projet fonctionnel
- ❑ Fonctionnalités expérimentales
- ❑ Evolution rapide d'Angular
- ❑ Sortie d'Angular 5.0

Des questions ?

