



Formation Angular 4 Avancé



Frédéric GAURAT

Une formation
Alphorm.com



Cursus Angular 4



Une formation
Alphorm.com



Une formation
Alphorm.com

Plan

Introduction

1. Comprendre le fonctionnement d'Angular
2. Les Reactive Forms
3. L'injection de dépendance et le développement de services
4. Les Pipes
5. Routing avancé
6. Le projet

Conclusion



Une formation
Alphorm.com

Public concerné

Développeurs et Architectes web



Une formation
Alphorm.com



Connaissances requises



Une formation
Alphorm.com

A vous de jouer !





Une formation
Alphorm.com

Présentation du projet de la formation



Frédéric GAURAT



Une formation
Alphorm.com

Plan

Le projet Todo List

Le serveur et la base de données : Docker

Présentation des outils



Le projet Todo List



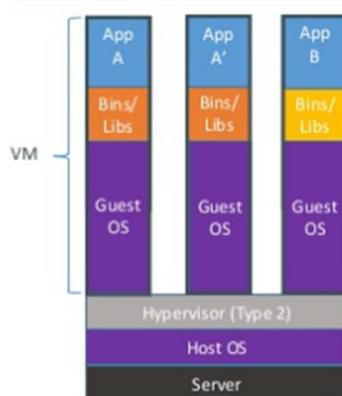
Helping you **select** an MV* framework

Une formation
Alphorm.com



Le serveur et la BDD : Docker

Containers vs. VMs



Containers are isolated,
but share OS and, where
appropriate, bins/libraries

...result is significantly faster deployment,
much less overhead, easier migration,
faster restart



Une formation
Alphorm.com



Une formation
Alphorm.com



Merci



Une formation
Alphorm.com

Rappel sur le fonctionnement de Javascript



Frédéric GAURAT



Une formation
Alphorm.com

Plan

Le fonctionnement de JavaScript
Les implémentations
Call Stack, Callback Queue et Event
Loop



Le fonctionnement de JS

JavaScript est un langage :

Single Threaded

Asynchrone

Il s'appuie sur des callbacks

Une formation
Alphorm.com



Les implémentations

Google's V8

Apple Safari

Carakan

Chakra


SpiderMonkey,

JägerMonkey,

IonMonkey



Une formation
Alphorm.com



Une formation
Alphorm.com

Call Stack

➔

```
function sayHello(){  
  return "Hello";  
}
```

➔

```
function printHello(){  
  var helloStr = sayHello();  
  console.log(helloStr);  
}
```

➔


```
printHello();
```

Stack


sayHello()

printHello()

main()



Une formation
Alphorm.com



TOUTE UTILISATION EN
DEHORS DE ALPHORM EST
INTERDITE

9



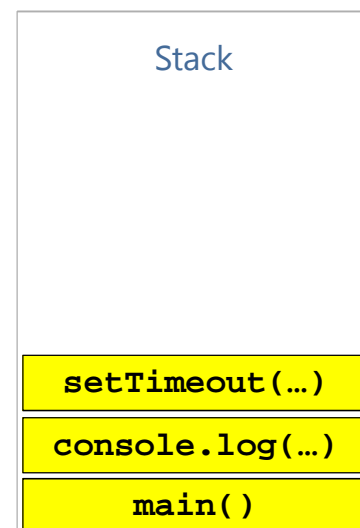
Callback Queue

```
➔ console.log("Hello");
```

```
➔ setTimeout(function(){
    console.log(" World");
},3000);
```

⌚

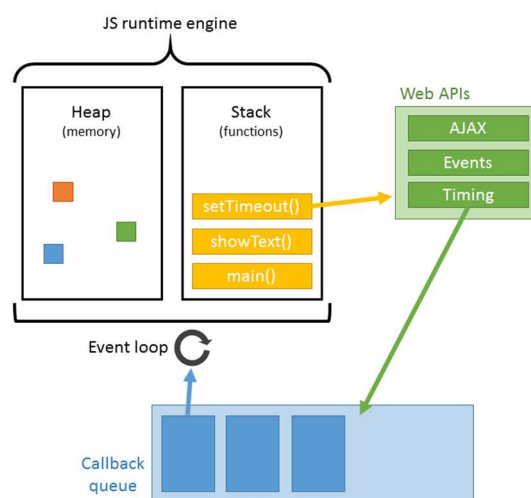
```
➔ console.log("end");
```



Une formation
Alphorm.com



Event Loop et Asynchrone



Une formation
Alphorm.com



Une formation
Alphorm.com



Merci



Une formation
Alphorm.com

Les Zones



Frédéric GAURAT



Une formation
Alphorm.com

Plan

- Qu'est que zone.js?
- Pourquoi utiliser zone.js?
- Fonctionnement de zone.js
- Utilisation de zone.js
- Zone.js dans Angular



Qu'est que zone.js?

Bibliothèque développée par l'équipe Angular (inspirée par **Dart**)

Utilisable hors d'Angular

Une zone est un contexte d'exécution

Une zone englobe l'exécution de code asynchrone

Une formation
Alphorm.com



Pourquoi utiliser zone.js

Garder des données entre différents appels asynchrones

Les frameworks peuvent l'utiliser pour détecter et intercepter des exécutions asynchrones

Facilite le debug, les tests et le profiling

Une formation
Alphorm.com



Fonctionnement de zone.js

zone.js « monkey patches » toutes les fonctions asynchrones du navigateur (*setTimeout(),...*)

« monkey patches » est un moyen de modifier un programme directement en mémoire

Une formation
Alphorm.com



Calcul du temps d'exécution d'une fonction
Calcul du temps d'exécution d'une fonction asynchrone

Une formation
Alphorm.com





Zone.js dans Angular

Angular s'appuie sur **NgZone**

NgZone est une implémentation de zone.js

NgZone ajoute une API basée sur des
Observables

Une formation
Alphorm.com

Merci



Une formation
Alphorm.com

Principe de détection et de notification des changements



Frédéric GAURAT



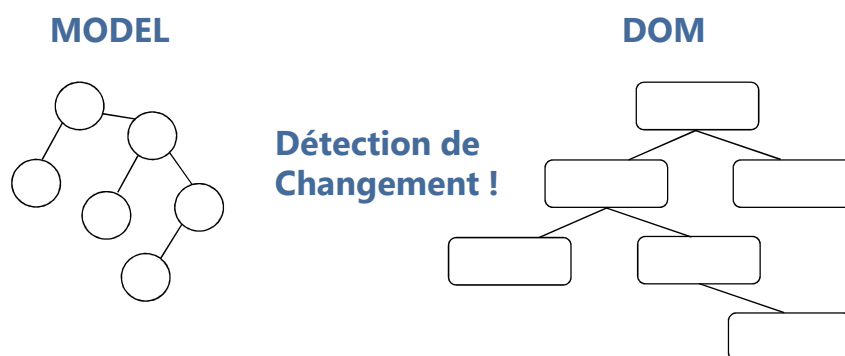
Une formation
Alphorm.com

Plan

- La détection de changement
- La cause du changement
- Notifier un changement
- Appliquer le changement



La détection de changement



Une formation
Alphorm.com



La détection de changement

Comment modifier le bon élément DOM en fonction de la modification du model ?

Comment le faire efficacement (manipulation d'arbre)

Une formation
Alphorm.com



La cause du changement

Events : click...

Ajax : récupération de data auprès du serveur

Timers : setTimeout(), setInterval()

Tous sont asynchrones !

Une formation
Alphorm.com



Notifier un changement

Les zones ! (Et plus particulièrement ngZone)

La classe ApplicationRef surveille ngZone et déclenche un tick pour effectuer le changement

Une formation
Alphorm.com



Appliquer le changement

Chaque component a son propre détecteur de changement qui applique les modifications

Organisés en arbre, datas top to bottom : ce n'est pas un cycle

Une formation
Alphorm.com



Une formation
Alphorm.com



Merci



Améliorer les performances de la détection de changement



Frédéric GAURAT

Une formation
Alphorm.com



Plan

Performance

Immutable et Observables

Améliorer la détection de changement

Une formation
Alphorm.com



Performance

Angular doit vérifier chaque composant
lorsque le modèle change

Indiquer à Angular d'exécuter les détecteurs
de changement pour les parties qui ont
changé : utiliser les objets immutables et les
observables

Une formation
Alphorm.com



Immutable et Observables

Immutable : ne peuvent pas changer

Nouvelle référence de cet objet : clone

Observables : notifient par un événement la modification d'un objet

Une formation
Alphorm.com



Améliorer la détection de changement

Stratégie de détection de changement

OnPush : l'objet utilisé dans le component est en entrée uniquement et doit être considéré comme immutable

Si c'est un observable, nous pouvons indiquer à Angular une modification

Une formation
Alphorm.com



Une formation
Alphorm.com



Merci



Une formation
Alphorm.com

Les Reactive Forms



Frédéric GAURAT



Une formation
Alphorm.com

Plan

Les Reactive forms

Reactive forms vs Template Driven forms

Démonstration



Une formation
Alphorm.com

Les Reactive forms

Arbre d'éléments de formulaire créé directement dans le composant

Éléments de l'arbre liés (bind) aux composants du formulaire contenu dans le template

Formulaire manipulé entièrement par le composant, valeurs toujours **synchronisées**



Une formation
Alphorm.com

Comparaison

Reactive forms



Template Driven forms

Synchrones

Le développeur code l'ensemble des éléments du formulaire

Asynchrones

La création est laissée aux directives



Une formation
Alphorm.com



Merci



Une formation
Alphorm.com

Le FormBuilder



Frédéric GAURAT



Une formation
Alphorm.com

Plan

Le FormBuilder

La validation

Différents types d'éléments de
formulaire

Formulaire imbriqué



Une formation
Alphorm.com

Le FormBuilder

Facilite la création de formulaire

C'est un service :

- Qu'il faut injecter dans le constructeur
- Qui permet la création de FormGroup

Le **FormGroup** permet de stocker les éléments du formulaire sous la forme d'un objet : **name=>definition**



Une formation
Alphorm.com





La validation

La classe **Validators** définit les contraintes de base

Appliquer une règle de validation :

Compléter la **déclaration** d'un
FormControl

Dans un **FormGroup**

Une formation
Alphorm.com



Une formation
Alphorm.com





Types d'éléments de formulaire

Le FormBuilder autorise le binding sur la
totalité des éléments de formulaire HTML

Une formation
Alphorm.com



Une formation
Alphorm.com





Formulaire imbriqué

Le FormBuilder permet de décomposer des formulaires complexes

Une formation
Alphorm.com



Une formation
Alphorm.com



Merci



L'injection de dépendance



Frédéric GAURAT

Une formation
Alphorm.com



Plan

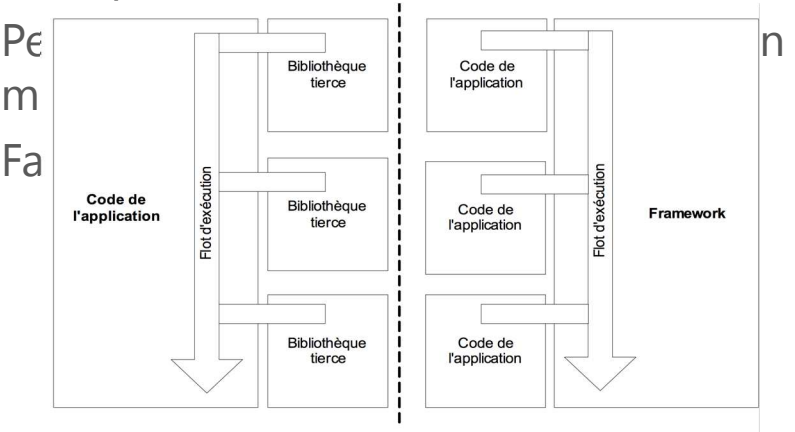
Qu'est ce que l'injection de dépendance?
Injection de dépendance dans Angular
Les services

Une formation
Alphorm.com



L'injection de dépendance ?

Principe basé sur l'inversion de contrôle



Une formation
Alphorm.com



Injection de dépendance

Angular a son propre mécanisme d'injection
: **l'injector**

Créé au boot de l'application

Utilisable dans la totalité de l'application

Configurer en enregistrant des providers

Les providers créent les services

Une formation
Alphorm.com



Les services

L'implémentation d'un service est
l'application du pattern Separation of
concerns (séparation des préoccupations)

Un service contient de la logique métier et
sera injecté dans des composants de
l'application

Un service est un singleton

Une formation
Alphorm.com



Les services

Un service est une simple classe marquée par le décorateur **@Injectable()**

Le décorateur @Injectable indique que la classe est disponible pour être injectée

Une formation
Alphorm.com

Merci



Une formation
Alphorm.com

Développement d'un service



Frédéric GAURAT



Une formation
Alphorm.com

Plan

Le service Logger
Configuration de l'injector
Utilisation du service



Le service Logger

Objectif : créer un Logger

- Il affichera les messages dans la console
- Il stockera les messages dans un tableau

Une formation
Alphorm.com



Une formation
Alphorm.com





Configuration de l'injector

Il peut être configuré dans 2 endroits :

1. **Le module** : le service est visible dans tous les composants du module
2. **Le composant** : le service n'est utilisable que dans ce composant

Une formation
Alphorm.com



Une formation
Alphorm.com



Utilisation du service

Injection du service

Une formation
Alphorm.com



Une formation
Alphorm.com



Merci



Une formation
Alphorm
com

Notion de pipes



Frédéric GAURAT



Plan

Qu'est ce qu'un pipe ?
Utilisation d'un pipe

Une formation
Alphorm.com



Qu'est ce qu'un pipe ?

Le pipe est le nouveau nom donné au filter d'AngularJS
Il permet de transformer une donnée à afficher



Une formation
Alphorm.com



Utilisation d'un pipe

Le pipe s'utilise de la même façon que dans AngularJS

En utilisant le caractère '|'

Suivi du filtre à utiliser et éventuellement des paramètres

Une formation
Alphorm.com



Une formation
Alphorm.com



Merci



Utilisation de pipes prédéfinis



Frédéric GAURAT

Une formation
Alphorm
com



Plan

Les pipes d'Angular
UpperCasePipe et LowerCasePipe
DatePipe
CurrencyPipe
PercentPipe

Une formation
Alphorm.com



UpperCasePipe et LowerCasePipe

Transforme du texte en majuscule
ou minuscule

```
{{ "hello" | uppercase }}  
{{ "HELLO" | lowercase }}
```

Une formation
Alphorm.com



Une formation
Alphorm.com



Une formation
Alphorm.com

DatePipe

Transforme une expression de date

Date object

Timestamp

Chaîne au format ISO

En une chaîne au bon format (suivant la locale)

```
{{1498119767000|date:'medium'}}
```



Une formation
Alphorm.com



Une formation
Alphorm.com

CurrencyPipe

Formate un nombre en devise monétaire

```
{{1.234|currency:'USD':true}}
```



Une formation
Alphorm.com



Une formation
Alphorm.com

PercentPipe

Formate un nombre en pourcentage (en fonction de la locale)

```
{{1.234 | percent}}
```



Une formation
Alphorm.com



Merci



Une formation
Alphorm.com

Développement d'un pipe personnalisé



Frédéric GAURAT



Une formation
Alphorm.com

Plan

Le décorateur @Pipe
L'interface PipeTransform
Implémentation du pipe



Le décorateur @Pipe

Le décorateur @Pipe est ajouté à la classe du pipe
Il prend 2 paramètres optionnels :

- name : (string) le nom du pipe
- pure : (boolean) état interne

Une formation
Alphorm.com



L'interface PipeTransform

L'interface PipeTransform contient une seule méthode

```
transform(value: any, ...args: any[]): any
```

value : la valeur en entrée de filtre

spread operator args : les paramètres éventuels

Une formation
Alphorm.com



Une formation
Alphorm.com



Merci



Une formation
Alphorm.com

Les Child Routes et les routes secondaires



Frédéric GAURAT



Une formation
Alphorm.com

Plan

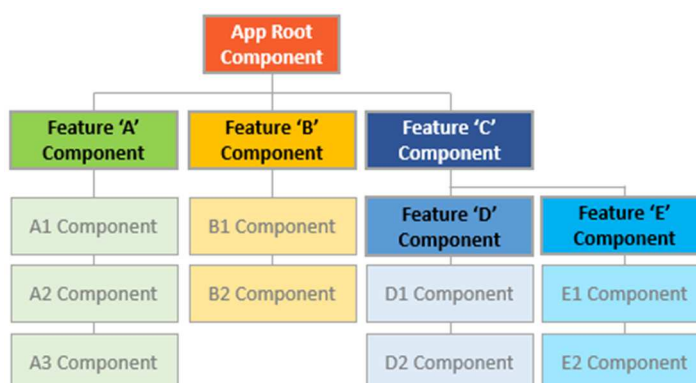
Organiser ses modules
Configuration
Chemins relatifs



Une formation
Alphorm.com

Organiser ses modules

Un module permet d'organiser son application en bloc de fonctionnalités



Une formation
Alphorm.com

Configuration

Les routes sont configurées dans un tableau d'objets de type **Routes**

Ces objets ont une propriété children : un tableau de type Routes



Une formation
Alphorm.com



Une formation
Alphorm.com

Chemins relatifs

Configuration par défaut basée sur
l'adressage absolue : à partir de la racine ('/')
Si le chemin du composant parent change, il
faut changer la totalité des chemins
Pour résoudre ce problème on va définir des
chemins relatifs à l'URL courante



Une formation
Alphorm.com



Une formation
Alphorm.com

Les routes secondaires

Dans le template d'un component, il est possible d'utiliser plusieurs `<router-outlet></router-outlet>`

Ils se distinguent par un attribut name :

```
<router-outlet name='contact'></router-outlet>
```

Ils sont la cible des **routes secondaires**



Une formation
Alphorm.com



Merci



Une formation
Alphorm.com

Route 'guards'



Frédéric GAURAT



Une formation
Alphorm.com

Plan

Surveillance des routes
Les 'guards'



Surveillance des routes

Vérifier l'accès à une route

Autorisation d'accès

Login nécessaire

Récupération de data avant la navigation

Sauvegarde et validation par l'utilisateur

Une formation
Alphorm.com



Les 'guards'

CanActivate : gère la navigation vers une route

CanActivateChild : gère la navigation vers une route enfant

Resolve : permet la récupération de donnée avant l'activation de la route

CanLoad : gère la navigation vers un module chargé de façon asynchrone

Une formation
Alphorm.com



Une formation
Alphorm.com



Merci



Une formation
Alphorm.com

Routing asynchrone



Frédéric GAURAT



Une formation
Alphorm.com

Plan

Le routing asynchrone
Exemple



Le routing asynchrone

Une application volumineuse doit être chargée dans le navigateur pour fonctionner
Pour éviter un chargement long, il est possible de demander certains modules « au bon moment » (lazy loading)

Une formation
Alphorm.com



Une formation
Alphorm.com

Merci



Projet TodoList Avancé



Frédéric GAURAT

Une formation
Alphorm.com



Plan

Le projet Todo List
Deployd et Docker

Une formation
Alphorm.com



Deployd

deployd

docs

community

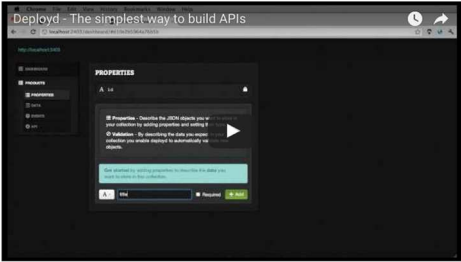
chat

Twitter

THE SIMPLEST WAY TO BUILD AN API.

An open source platform

Design, build, and scale APIs for web and mobile apps in minutes instead of days.



Install deployd

npm v0.8.9

Follow @deploydapp

Star 4,558

Fork 643

Une formation
Alphorm.com



Docker



Une formation
Alphorm.com



Une formation
Alphorm.com

Merci



Une formation
Alphorm
com

Conclusion



Frédéric GAURAT



Bilan

Le fonctionnement d'Angular

Les reactive forms

Les services

Les pipes

Les routes (avancé)

Une formation
Alphorm.com

Merci