ANGULAR

Installation de l'environnement

Prérequis

JDK 8

JAVA_HOME

- https://nodejs.org/en/
- Installation: typescript, angular -cli, gulp

```
npm install -g @angular/cli gulp typescript
```

- Désinstallation d'une ancienne version de angular-cli (ligne de commande en mode administrateur)

```
npm uninstall -g @angular/cli
```

```
npm cache clear
```

Typescript

- Langage de programmation open source développé par Microsoft en 2012 qui génère du Javascript
- Types de données suppotés: number, string, boolean
- Décorateurs: depuis ES2017

Javascript

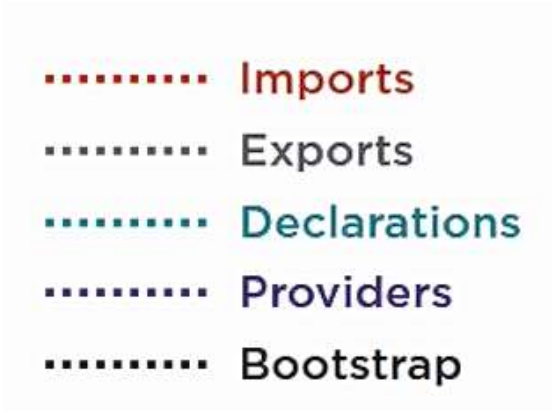
```
var num = 455;
var nom = "Javascript";
var test = true;
var data = "155abc";
var liste = [1, 3, 5];
function somme(a, b) {
    return a + b;
}
```

Typescript

```
var num: number = 455;
var nom: string = "Javascript";
var test: boolean = true;
var data: any = "155abc";
var liste: Array<number> = [1, 3, 5];
function somme(a:number, b:number) :number{
    return a + b;
}
```

Angular

- Le design pattern component
- MVC
- Type Script
- Le standard web component (webcomponents.org)
  - Éléments personnalisés
  - Html imports
  - Templates: blocs de code html qui peuvent être insérés dynamiquement par code javascript
  - Shadow DOM: une zone masquée dans une page qui peut contenir des scripts, css, html non accessible dans la page mais utilisable par les composants
  - Une application angular est composée de modules, chaque module peut contenir un ou plusieurs composants
  - Une application angular doit posséder au moins un module racine.
  - NgModule permet de définir un module



Angular 1 vs Angular 2

- Angular 1
  - Framework MVC
  - Templates côté client

- Angular 2
  - UI basée sur les composants
  - Plus de modularité
  - Supporte le code Angular 1
  - Typescript
  - Plus rapide.

Navigation

- Le module RouterModule (@angular/router) est un mécanisme de navigation entre vues.
- Configuration:
  - Routes
- Directives:
  - RouterLink
  - RouterOutlet
  - RouterLinkActive

Configuration:

- Dans app.module.ts:importer RouterModule

```
import {RouterModule } from '@angular/router’;
```

- Dans la section imports:

```
RouterModule.forRoot([
{
path: ‘user',
component: UserComponent
},
{
path: ‘ressources',
component: RessourceComponent
}
])
],
```

Directives structurelles

***ngFor**

```
<ul>
<li *ngFor="let user of users">

{{user.nom}}
</li>
</ul>
```

***ngSwitch**

- [ngSwitch]="expression«
- *ngSwitchCase=« val_expression«
- *ngSwitchDefault

***ngIf**

```
<p *ngIf=« users.length < 1">La liste est vide!</p>
```

***ngIf/Else**

```
<div *ngIf="condition;then contenu else elseContenu">Cette partie est ignorée</div>
<ng-template #contenu>contenu </ng-template>
<ng-template #elseContenu>else contenu</ng-template>
```

Liaison aux données

```
<h2>Liaison aux données</h2>
<h3>Interpolations</h3>
<h4>   {{nom}} </h4>
<input type="text" name="t1" [value]="nom"/>   Ou bien <input type="text" name="t1" [ngModel]="t1">
<p>{{num}}</p>
<p>{{num|json}}</p>
<h3>Event bindings</h3>
<button (click)="f1()">Cliquez ici</button>
<h3>Liaisons bidirectionnelles</h3>
<!--pour ngModel FormsModule doit être importé-->
val:<input [(ngModel)]="val"/>
<p>Valeur de val: <span>{{val}}</span></p>
```

Liaison dans une seule direction

Liaison dans les deux direction:
[(ngModel)="t1" est équivalent à
[(ngModel)="t1"
ngModelChange="t1=$event"

Liaison aux données

1- le modèle: Dans le dossier app créer models/res.models.ts

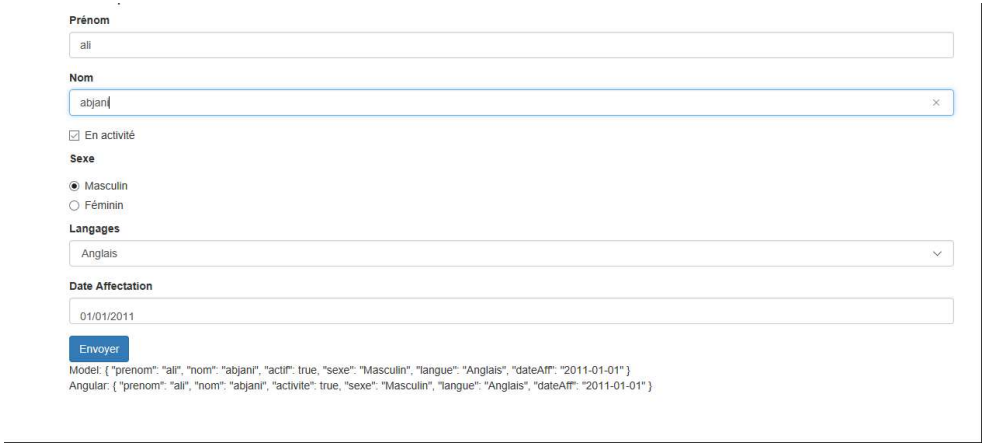
```
export class Ressource {
    constructor(
        public prenom: string,
        public nom: string,
        public actif: boolean,
        public sexe: string,
        public langue: string,
        public dateAff: string
    ) {

    }

}
```

2- dans le composant Angular contenant le formulaire, importer le modèle Ressource

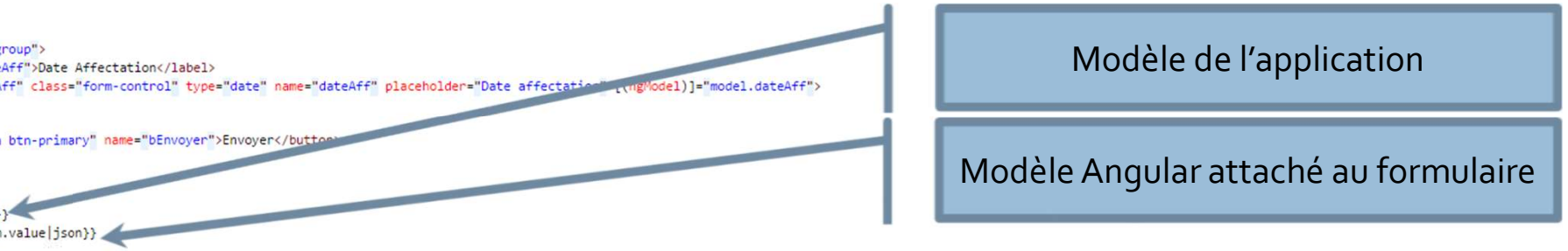
```
import { Ressource } from "./models/res.model";
```

3- dans le code Javascript du composant, définissez votre modèle

```
model:Ressource = new Ressource("ali", "abjani", true,  "Masculin","Anglais", "2011-01-01");
```



```
<div class="container">
  <h1>
    {{title}}
  </h1>
  <h3>Exemple 1</h3>
  <form #form="ngForm" novalidate>
    <div class="form-group">
      <label for="prenom">Prénom</label>
      <input id="prenom" class="form-control" type="text" required name="prenom" placeholder="Prénom" [(ngModel)]="model.prenom">
    </div>
    <div class="form-group">
      <label for="nom">Nom</label>
      <input id="nom" class="form-control" type="text" required name="nom" placeholder="Nom" [(ngModel)]="model.nom" (ngModelChange)="nomToMaj($event)"
      />
    </div>

    <div class="checkbox">
      <label>
     <input type="checkbox" name="activite" [(ngModel)]="model.actif">En activité</label>
    </div>

    <label>Sexe</label>
    <div class="radio">
      <label>
      <input type="radio" name="sexe" value="Masculin" checked="checked" [(ngModel)]="model.sexe"/>Masculin</label>
    </div>

    <div class="radio">
      <label>
      <input type="radio" name="sexe" value="Féminin" [(ngModel)]="model.sexe"/>Féminin</label>
    </div>


    <div class="form-group">
      <label for="langue">Langages</label>
      <select class="form-control" id="langue" name="langue" [(ngModel)]="model.langue">
<option *ngFor="let l of langue" [value]="l">{{l}}</option>
    </select>
    </div>


    <div class="form-group">
      <label for="dateAff">Date Affectation</label>
      <input id="dateAff" class="form-control" type="date" name="dateAff" placeholder="Date affectation" [(ngModel)]="model.dateAff">
    </div>

    <button class="btn btn-primary" name="bEnvoyer">Envoyer</button>

  </form>

  Model: {{model|json}}
  <br> Angular: {{form.value|json}}
</div>
```

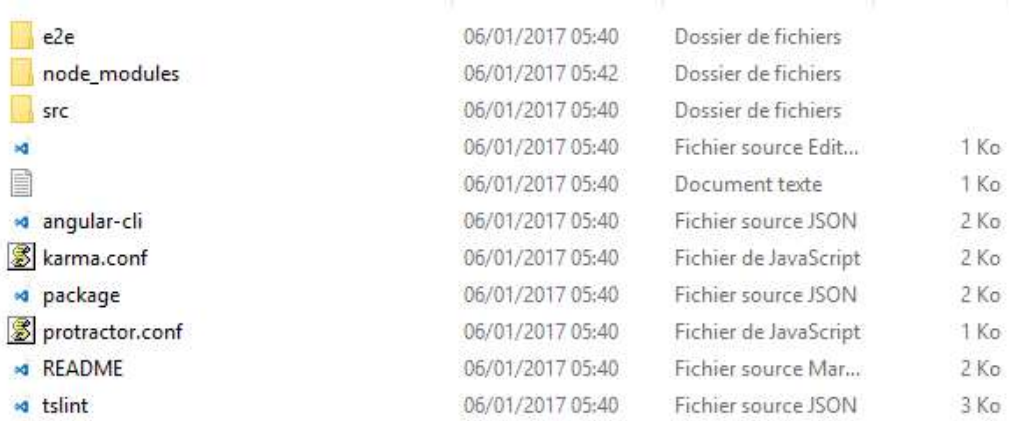


- Les directives ngFor et ngIf sont définies dans BrowserModule importé par AppModule

```
<h4 [class.red]="toggle">Mes tâches</h4>
<h4 [ngClass]="{red:toggle,blue:!toggle}">Mes tâches</h4>
<h4 [ngClass]="{red:toggle}">Mes tâches</h4>
<h4 *ngIf="toggle">Je suis visible</h4>
<ul>
    <li *ngFor="let task of tasks">
        {{ task }}
    </li>
</ul>
```

Atelier 1: une première application

- exécuter la commande `ng new app1`
- Angular-cli créer un dossier nommé app1 contenant la structure suivante:



Validation des champs d'un formulaires

- Classes css
  - ng-untouched ng-pristine ng-valid
  - ng-touched (lostfocus) ng-dirty ng-invalid
- Propriétés ngModel attachées
  - untouched pristine valid
  - touched dirty invalid
- Attributs html5: required, minlength, maxlength, pattern

Les services

```
h2>Les services</h2>
span>{{taskService.tasks | json}}</span>
ul>
    <li *ngFor="let t of taskService.tasks">
        {{t}}
    </li>

/ul>
h2>Le Routage</h2>
```

- Le premier composant
  - UI dans une vue
  - Le comportement dans une classe

Routage et navigation

1. Importer RouterModule et Routes (app/app.module.ts)

```
import { RouterModule, Routes } from '@angular/router';
```

Une application angular possède une seule instance singleton Router,lorsque l'url dans le navigateur change, cette instance cherche la route correspondante à partir de laquelle elle cherche le composant à afficher

2 Ajout des liens

```
export const appRoutes:Routes=[  {path:'tasks',component:TasksComponent}];
```

3 Configuration des routes

```
@NgModule({
 imports:   [ BrowserModule,FormsModule, RouterModule.forRoot (appRoutes)]
```

4- dans le template

```
<a routerLink='/tasks' routerLinkActive='active'>Tasks</a>
<router-outlet></router-outlet>
```

Observable et données serveur.

- Installation du module rxjs et import

```
import { HttpModule,Http ,Response} from '@angular/http';
import {Observable} from 'rxjs/Observable';
```

- Données: taches.json

```
{
 "data":   [{"titre":"Tache 1","termine":true,"created_at":"","updated_at":""},
{"titre":"Tache 2","termine":true,"created_at":"","updated_at":""},
 {"titre":"Tache 3","termine":true,"created_at":"","updated_at":""}]
}
```

Promises vs Observable

Promise	Observable
Retourne une valeur	Retourne plusieurs valeurs
Ne peut pas être annulé	Peut être annulé
Supporté nativement par les navigateurs	Basé sur des librairies comme RxJS
	Supporte les fonctions standards sur les tableaux (map, reduce, filter)