



Accueil > Cours > Débutez avec Angular > Quiz : Améliorer la structure de votre application avec les services et le routing

# Débutez avec Angular



Mis à jour le 13/06/2022



# Améliorer la structure de votre application avec les services et le routing

Vous n'avez pas validé ce quiz.

Vous n'avez pas atteint le seuil de validation de cet exercice, c'est-à-dire 70%. Ce n'est pas très grave car vous pourrez refaire le quiz dans 24h.

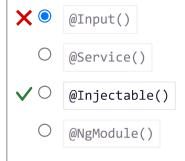
## Compétences évaluées



Améliorer la structure de votre application avec les services et le routing

#### **Question 1**

Quel décorateur permet de transformer facilement une classe TypeScript en service?



Le décorateur @Injectable()

transforme une classe en service.

#### **Question 2**

#### Quelle option ci-dessous enregistre FolderService à la racine de mon application?

```
XO
                                  typescript
          1 @Injectable('root')
          2 export class FolderService {}
V O
                                  typescript
          1 @Injectable({
                providedIn: 'root'
          3 })
          4 export class FolderService {}
   \bigcirc
                                  typescript
          1 @Injectable({
          2 injectIn: 'root'
          4 export class FolderService {}
   \bigcirc
                                        typescript
          1 @Injectable()
          2 export class FolderService<root> {}
En passant un objet de configuration { providedIn: 'root' } au décorateur @Injectable()
, on s'assure d'enregistrer le service à la racine de l'application.
```

#### **Question 3**

Je veux injecter un service MarbleService dans GamesComponent. Quelle est la bonne syntaxe ?

```
1 @Component({
2   selector: 'app-games',
3   templateUrl: './games.component.html',
4   styleUrls: ['./games.component.scss']
5 })
6 export class GamesComponent implements OnInit {
7  constructor() {
```



typescript

```
1 @Component({
       selector: 'app-games',
 3
       templateUrl: './games.component.html',
       styleUrls: ['./games.component.scss']
 4
 5 })
  export class GamesComponent implements OnInit {
       constructor(private marbleService = MarbleService) {
 8
 9
10
       ngOnInit() {
11
12
13
14 }
```

 $\bigcirc$ typescript 1 @Component({ selector: 'app-games', 3 templateUrl: './games.component.html', styleUrls: ['./games.component.scss'] export class GamesComponent implements OnInit { 7 constructor() { 8 9 10 11 private marbleService = MarbleService; 12 13 14 }



typescript

```
1 @Component({
2    selector: 'app-games',
3    templateUrl: './games.component.html',
4    styleUrls: ['./games.component.scss']
5 })
6 export class GamesComponent implements OnInit {
7    constructor(private marbleService: MarbleService) {
8     9    }
10     11    ngOnInit() {
```

```
12
13 }
14 }
```

Pour injecter un service dans un component, on passe un argument du type du service au constructor avec un modificateur d'accès ; ex. :

typescript

```
1 constructor(private myService: MyService) {}
```

#### **Question 4**

Mon service LightsService contient une méthode getAllLights() qui retourne un tableau de type Light[] .

Dans RedLightsComponent, je veux initialiser ma propriété lights à partir de la méthode du service.

#### Comment je fais?

```
typescript

//...

lights: Light[];

constructor(private lightsService: LightsService) {

function of the service of the
```

```
1 //...
2 Lights: Light[];
3
4 constructor(private LightsService: LightsService) {
5     this.lights = this.lightsService.getAllLights();
6 }
7
8 ngOnInit() {
9
10 }
11 //...
```

O



typescript

```
1 //...
2 Lights: Light[];
3
4 constructor() {
5
6 }
7
8 ngOnInit(private LightsService: LightsService) {
9    this.lights = this.lightsService.getAllLights();
10 }
11 //...
```

Dans le constructor, on injecte le service.

Dans | ngOnInit() | , on initialise la propriété en appelant la méthode.

# **Question 5**

#### Voici la déclaration de la méthode addCowbell() de CowbellService :

typescript

```
1 addCowbell(howMuch: 'some' | 'more', when: number): Cowbell[] {
2   //...
3 }
```

#### Quelle expression ci-dessous est valable?

```
O typescript

1 const cowbell: Cowbell = this.cowbellService.addCowbell('more', 5);
```

```
O typescript

1 const cowbells: Cowbell[] = this.cowbellService.addCowbell('some', '5');
```



typescript

```
1 const cowbells: Cowbell[] = this.cowbellService.addCowbell('more', 5);
```

```
\bigcirc
                                                                             typescript
       1 const cowbells: Cowbell[] = this.cowbellService.addCowbell('all', 5);
```

Faites bien attention aux types des arguments et du retour!

# **Question 6**

Quelle déclaration de route associe correctement le component CoffeeComponent à la route 'coffee'?

```
XO
                                                  typescript
          1 { path: coffee, component: CoffeeComponent }
V O
                                                    typescript
          1 { path: 'coffee', component: CoffeeComponent }
   \bigcirc
                                                    typescript
          1 { path: coffee, component: 'CoffeeComponent' }
   \bigcirc
                                                     typescript
          1 { route: 'coffee', component: CoffeeComponent }
La déclaration de route spécifiée a besoin d'un path de type
                                                                                       component
                                                                   string
est le type du component correspondant.
```

# **Question 7**

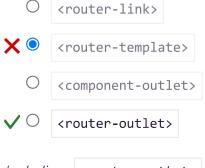
Quel lien se verra attribuer la classe 'active-link' lorsque la route qui y correspond sera la route active?

```
V O
                                                                              html
           1 <a routerLink="/stuff" routerLinkActive="active-link">Stuff</a>
   \bigcirc
                                                          html
             <a routerLinkActive="active-link">Places</a>
                                                                         html
```



#### **Question 8**

# Quelle balise permet de spécifier l'endroit dans le DOM où le component de la route active sera inséré ?



La balise | <router-outlet> | spécifie l'emplacement du component à afficher.

### **Question 9**

Une fois le Router injecté dans GadgetComponent, comment je l'utilise pour naviguer vers la route new-gadgets ?

```
typescript
```

La bonne méthode est

navigateByUrL()

#### **Question 10**

La route où se trouve mon component contient un paramètre dynamique oldAmount . Je veux récupérer ce paramètre et l'utiliser pour appeler cette méthode de service :

```
1 calculateNewAmount(oldAmount: number): number {
2    //...
3 }
```

#### Quelle est la bonne solution?



typescript

```
1 //...
 2 export class AccountantComponent implements OnInit {
       oldAmount: number;
4
       newAmount: number;
 5
       constructor(private route: ActivatedRoute,
 7
                    private amountService: AmountService) {}
 8
9
       ngOnInit() {
10
           this.oldAmount = this.route.params['oldAmount'];
           this.newAmount = this.amountService.calculateNewAmount(this.oldAmount);
11
12
13 }
```

 $\bigcirc$ typescript 2 export class AccountantComponent implements OnInit { 3 oLdAmount: number; newAmount: number; 4 5 constructor(private route: ActivatedRoute, 7 private amountService: AmountService) {} 8 9 ngOnInit() { 10 this.oldAmount = this.route.params['oldAmount']; this.newAmount = +this.amountService.calculateNewAmount(this.oldAmount); 11 12 13 }

```
~ O
```

typescript

```
1 //...
2 export class AccountantComponent implements OnInit {
3    oLdAmount: number;
4    newAmount: number;
5
6    constructor(private route: ActivatedRoute,
```

 $\bigcirc$ typescript 1 //... 2 export class AccountantComponent implements OnInit { 3 oldAmount: number; 4 newAmount: number; constructor(private route: ActivatedRoute, private amountService: AmountService) {} 8 9 ngOnInit() { this.oldAmount = this.route.params['oldAmount']; 10 this.newAmount = this.amountService.calculateNewAmount(this.+oldAmount); 11 12 13 }

Le paramètre se récupère via la clé params de l'objet snapshot de ActivatedRoute. Il faut le cast en number avant d'appeler la méthode, car les params de route sont de type string

**←** Activez les routes avec ActivatedRoute

Conclusion

# Le professeur

AIDE

#### Will Alexander

Scottish developer, teacher and musician based in Paris.

OPENCLASSROOMS

OPPORTUNITÉS

V

