

Les éléments à télécharger sont disponibles à l'adresse suivante :  
**<http://www.editions-eni.fr>**  
Saisissez la référence de l'ouvrage **EI2JAV** dans la zone de recherche  
et validez. Cliquez sur le titre du livre puis sur le bouton de téléchargement.

## Avant-propos

### Chapitre 1

#### Bien démarrer vos projets

1. Choisir votre environnement de développement . . . . .	13
1.1 Introduction . . . . .	13
1.2 NetBeans . . . . .	16
1.3 Visual Studio Express . . . . .	17
1.4 Eclipse . . . . .	19
1.5 Aptana . . . . .	20
1.6 WebStorm . . . . .	22
1.7 Bilan . . . . .	23
2. Organiser votre code . . . . .	24
2.1 Séparer le code de la présentation . . . . .	24
2.2 Chargement en pied de page . . . . .	28
2.3 Chargement de script par le cache du navigateur . . . . .	29
3. Portée des variables et fonctions . . . . .	33
3.1 Portée des variables . . . . .	33
3.2 Portée des variables dans une fonction . . . . .	34
4. Optimisation de la portée . . . . .	36
4.1 Limiter le contexte global . . . . .	36
4.2 Fonction anonyme . . . . .	37
4.3 Fermeture . . . . .	39
5. Simplifier vos expressions . . . . .	40
5.1 L'opérateur    . . . . .	40
5.2 L'opérateur && . . . . .	42

5.3	Comparaison	43
5.4	Paramètres variants	45
5.5	Extension de type	46
5.5.1	prototype	46
5.5.2	Array	47
5.5.3	String	48
5.5.4	Function	49
5.5.5	Object	50
6.	Passage au mode strict	51
6.1	Usage	51
6.2	Quelques cas	54
7.	JSHint	56
7.1	Usage de base	56
7.2	Chargement de code	58
7.3	Autre usage	60
7.3.1	Ligne de commande	60
7.3.2	Dans votre éditeur	62
8.	Documenter votre code avec JSDoc	63
8.1	Le principe	63
8.2	Les marqueurs	64
8.2.1	Déclarations	64
8.2.2	Fonctions	64
8.2.3	Objet	66
8.2.4	Meta	68
8.3	Usage	68
8.3.1	EDI WebStorm	68
8.3.2	Génération d'une documentation	69
9.	Passage d'un environnement de développement à un environnement de production	72
9.1	Tests unitaires	72
9.1.1	Introduction	72
9.1.2	Gestion avec votre propre librairie	72

- 9.1.3 QUnit .....76
- 9.2 Minimisation .....79
  - 9.2.1 Présentation .....79
  - 9.2.2 YUI Compressor.....80
  - 9.2.3 Closure Compiler .....81

**Chapitre 2**  
**Développer efficacement en objet**

- 1. Première approche .....83
  - 1.1 Rappels .....83
    - 1.1.1 Classe.....83
    - 1.1.2 Instance.....84
    - 1.1.3 Encapsulation .....85
    - 1.1.4 Héritage.....85
    - 1.1.5 Abstraction .....86
    - 1.1.6 Surcharge.....86
    - 1.1.7 L'interface .....87
    - 1.1.8 Polymorphisme.....87
  - 1.2 Construction d'un objet.....88
    - 1.2.1 Préambule .....88
    - 1.2.2 Instance directe.....88
    - 1.2.3 Tableau associatif.....91
    - 1.2.4 Parcours des propriétés.....92
  - 1.3 Construction d'une classe .....94
    - 1.3.1 Le constructeur.....94
    - 1.3.2 L'opérateur new .....94
    - 1.3.3 L'opérateur this.....95
    - 1.3.4 Le mot-clé with.....96
    - 1.3.5 Le mot-clé instanceof.....97
- 2. Contexte d'exécution .....97
  - 2.1 this.....97
  - 2.2 that .....100

2.3	Fonction anonyme	100
2.4	Binding	101
3.	Classes prédéfinies	104
3.1	Object	104
3.2	String	106
3.3	RegExp	109
3.4	Array	112
3.5	Date	115
3.6	Math	117
4.	Notions avancées	118
4.1	Prototypage	118
4.1.1	Simple	118
4.1.2	Optimisée	121
4.2	Héritage	122
4.2.1	Principe	122
4.2.2	Surcharge	124
4.2.3	polymorphisme	126
4.2.4	Résolution des conflits	127
4.3	Gestion mémoire	129
4.3.1	Garbage collector	129
4.3.2	null	129
4.3.3	delete	129
4.3.4	var	130
5.	Framework pour le développement objet	130
5.1	Prototype	130
5.1.1	Création d'une classe	130
5.1.2	Héritage	131
5.1.3	Composition	132
5.2	MooTools	133
5.2.1	Création d'une classe	134
5.2.2	Héritage	134
5.2.3	Implémentation	135

- 5.3 jQuery ..... 136
  - 5.3.1 Introduction ..... 136
  - 5.3.2 Merge simple ..... 136
  - 5.3.3 Mélange récursif ..... 137
- 5.4 Dojo ..... 138
  - 5.4.1 Introduction ..... 138
  - 5.4.2 Création d'une classe ..... 138
  - 5.4.3 Héritage ..... 139

**Chapitre 3**  
**Adopter les bonnes pratiques**

- 1. Espace de noms ..... 141
  - 1.1 Principe ..... 141
  - 1.2 Fonction ..... 142
    - 1.2.1 Fonction interne ..... 142
    - 1.2.2 Fonction anonyme ..... 144
    - 1.2.3 Fonction anonyme avec paramètres ..... 145
  - 1.3 Fermeture ..... 147
  - 1.4 Classe ..... 148
- 2. Module ..... 151
  - 2.1 Présentation ..... 151
  - 2.2 Composition ..... 154
- 3. CommonJS ..... 156
  - 3.1 Présentation ..... 156
  - 3.2 define ..... 156
- 4. AMD ..... 159
  - 4.1 Introduction ..... 159
    - 4.1.1 Présentation ..... 159
    - 4.1.2 Quelques conseils ..... 160
  - 4.2 RequireJS ..... 161
  - 4.3 Dojo ..... 162

4.4	curl	164
4.4.1	Principe	164
4.4.2	Accès relatif	165
4.4.3	Gestion des erreurs de chargement	166
4.5	Écriture de votre gestionnaire de modules	166
5.	Déboguer votre code	170
5.1	Fonction alert	170
5.1.1	Usage	170
5.1.2	Simplification	171
5.1.3	Module	172
5.2	Activation/désactivation	174
5.3	Fenêtre indépendante	174
5.4	Console	177
5.4.1	Accès	177
5.4.2	log	178
5.4.3	Groupe	182
5.4.4	Niveaux de trace	182
5.4.5	Mesure	183
5.4.6	Pile d'appels	184
5.5	Débogueur intégré	185
5.5.1	Internet Explorer	185
5.5.2	Chrome	187
5.5.3	Firefox/Firebug	189
5.5.4	Conclusion	191

## Chapitre 4

### Améliorer vos compétences Web

1.	Page HTML	193
1.1	L'indispensable	193
1.1.1	Balises	193
1.1.2	Balises de structure	195
1.1.3	Lien	196

1.1.4	Tableau	197
1.1.5	Formulaire	198
1.1.6	Générique	200
1.2	Le CSS	200
1.2.1	Déclaration	200
1.2.2	Sélecteurs	202
1.2.3	Propriétés	203
2.	Librairies JavaScript	206
2.1	L'objet window	206
2.1.1	Rôle	206
2.1.2	open	207
2.1.3	setInterval, setTimeout	208
2.1.4	location	210
2.1.5	navigator	211
3.	DOM	214
3.1	Document	214
3.1.1	Propriétés et méthodes	214
3.1.2	Parcours	217
3.2	Modification	219
3.2.1	Construction	222
3.2.2	Événements	223
4.	Formulaire	229
4.1	Validation simple	229
4.2	Gestion des champs	232
4.3	Contraintes supplémentaires	233
4.4	Conception d'un module de validation	234
5.	Étude de cas	236
5.1	Gestion de notes dans une page Web, architecture MVC	236
5.1.1	Première étape	236
5.1.2	Modèle de données	237
5.1.3	La vue	238
5.1.4	Finalisation	238

5.2	Gestion de notes, version 2	240
5.2.1	Multivue	240
5.2.2	Trier les notes	242
5.3	Gestion d'un QCM dans une page web	244
5.3.1	Première étape	244
5.3.2	Réalisation	248

## Chapitre 5

### Développer aisément en client/serveur

1.	AJAX	251
1.1	Requête simple	251
1.2	Réponse XML	254
1.3	Paramètres GET	255
1.4	Paramètres POST	256
1.5	Requêtes entre domaines différents	258
1.6	Module	259
1.6.1	Version monorequête	259
1.6.2	Version multirequête	261
1.7	Format d'échange	264
1.7.1	Texte	264
1.7.2	XML	267
1.7.3	XSLT	270
1.7.4	JSON	279
1.8	Objets en client/serveur	284
1.8.1	Implémentation Java intrusive	284
1.8.2	Implémentation Java non intrusive	289
2.	Chargement dynamique de script	292
2.1	Première implémentation	292
2.2	Module AMD	294



**Chapitre 6**  
**Maîtriser les frameworks Web**

- 1. Bien démarrer avec jQuery . . . . . 299
  - 1.1 Concepts . . . . . 299
    - 1.1.1 Présentation . . . . . 299
    - 1.1.2 Sélecteur . . . . . 301
    - 1.1.3 DOM . . . . . 309
    - 1.1.4 Utilitaires . . . . . 312
    - 1.1.5 Effets de transition . . . . . 314
  - 1.2 Événements . . . . . 315
    - 1.2.1 Interactions utilisateur . . . . . 315
    - 1.2.2 AJAX . . . . . 317
    - 1.2.3 Conclusion . . . . . 321
  - 1.3 Plug-ins jQuery . . . . . 322
    - 1.3.1 Construction . . . . . 322
    - 1.3.2 Usage . . . . . 324
- 2. Bien démarrer avec Dojo . . . . . 327
  - 2.1 Concepts . . . . . 327
    - 2.1.1 Présentation . . . . . 327
    - 2.1.2 Configuration . . . . . 328
    - 2.1.3 Vos modules AMD . . . . . 331
    - 2.1.4 DOM . . . . . 333
  - 2.2 Événements . . . . . 338
    - 2.2.1 Interactions utilisateur . . . . . 338
    - 2.2.2 AJAX . . . . . 341
    - 2.2.3 Conclusion . . . . . 343
- 3. Construire votre framework web . . . . . 343
  - 3.1 Première étape . . . . . 343
    - 3.1.1 Présentation . . . . . 343
    - 3.1.2 Organisation . . . . . 344

3.2	Deuxième étape	346
3.2.1	Usage de templates de composant	346
3.2.2	Réalisation d'un bouton	350
3.3	Troisième étape	352
3.3.1	Création d'une calculatrice simple	352
3.3.2	L'avenir	356

## Chapitre 7

### TypeScript

1.	Introduction	359
1.1	Objectifs	359
1.2	Hello world	360
1.3	Environnement de développement	361
1.3.1	Visual Studio Community 2017	361
1.3.2	WebStorm 2017	362
2.	Variable et constante	362
2.1	Variable	362
2.2	Constante	364
3.	Typage	365
3.1	Déclaration	365
3.1.1	Variable	365
3.1.2	Fonctions	365
3.2	Chaînes	366
3.3	Nombre	366
3.4	Booléens	367
3.5	Énumération	367
3.6	Any, Null et Undefined	367
3.6.1	Any	367
3.6.2	Null et Undefined	368
3.7	Tableaux	369
3.8	Assertions	370
3.9	Itérations	370

4. Classes	371
4.1 Déclaration et usage	371
4.2 Héritage	372
4.3 Visibilité des champs	373
4.4 Propriétés en lecture seule	375
4.5 Accesseurs	376
4.6 Propriétés statiques	377
5. Interfaces	378
5.1 Déclaration	378
5.2 Propriétés optionnelles et en lecture seule	379
5.3 Types de fonctions	381
5.4 Types indexables	381
5.5 Implémentation	383
6. Génériques	385
6.1 Déclaration et usage	385
6.2 Types génériques	386
6.3 Classes génériques	388
6.4 Contraintes	388
7. Modules	390
7.1 Déclaration et usage	390
7.2 Import	392
8. Espace de noms	393
8.1 Déclaration et usage	393
8.2 Espaces de noms multiples	395
8.2.1 Construction d'un fichier unique	395
8.2.2 Instruction de compilation	396
8.3 Espaces de noms imbriqués	397
8.4 Alias	398

**Annexes****Liste des URL**

1. Éditeurs JavaScript .....	399
2. Frameworks .....	400
3. Frameworks objets .....	400
4. Librairies .....	400
5. Gestionnaires AMD .....	400
6. Outils .....	401
7. Plugins jQuery .....	402
8. Références .....	402

Index .....	403
-------------	-----

Les éléments à télécharger sont disponibles à l'adresse suivante :  
**<http://www.editions-eni.fr>**  
Saisissez la référence ENI de l'ouvrage **EI2ANGNOD** dans la zone de recherche  
et validez. Cliquez sur le titre du livre puis sur le bouton de téléchargement.

## Avant-propos

### Chapitre 1 Introduction

1. Introduction . . . . .	13
2. L'architecture MEAN pour une application web . . . . .	15
2.1 Le principe des applications monopages (single page applications) . . . . .	16
2.2 Le paradigme de conception modèle-vue-contrôleur . . . . .	16
3. Angular au centre de l'architecture MEAN . . . . .	19
4. Présentation du fil rouge : une application d'e-commerce . . . . .	21

### Chapitre 2 Le langage JavaScript

1. Introduction à JavaScript . . . . .	25
1.1 Bref historique . . . . .	25
1.2 Panorama de l'utilisation de JavaScript . . . . .	26
1.3 Les bibliothèques et les frameworks applicatifs JavaScript . . . . .	27
2. Où coder du code JavaScript ? . . . . .	28
3. Les outils du navigateur et le débogage . . . . .	29
4. Les éléments de programmation de base . . . . .	30
4.1 Les variables . . . . .	30
4.1.1 Les types internes . . . . .	30
4.1.2 Le transtypage . . . . .	31

# 2 **Angular et Node.js**

Optimisez le développement de vos applications web

4.2	Les structures de données usuelles	32
4.3	Application d'expressions régulières	34
4.4	Les blocs d'instructions	35
4.5	Les structures conditionnelles	35
4.5.1	La structure if ... else	35
4.5.2	La structure switch d'aiguillage multiple	36
4.6	Les structures itératives	36
4.6.1	Les structures itératives avec indices de boucle	36
4.6.2	Les structures itératives sans indices de boucle	37
5.	La programmation fonctionnelle en JavaScript	39
5.1	Une fonction passée en paramètre (fonction de callback)	39
5.1.1	Exemple avec la méthode forEach()	39
5.1.2	Exemple avec la méthode map()	40
5.2	Une fonction retourne une fonction (factory)	41
6.	La programmation objet avec JavaScript	42
6.1	Les principes de la programmation objet avec JavaScript	42
6.2	Les objets littéraux	43
6.3	L'héritage par chaînage de prototypes	44
6.3.1	La propriété __proto__ de l'objet héritant	44
6.3.2	La propriété prototype	45
6.4	La création d'un objet par l'appel d'une fonction constructrice	46
6.5	Exemples d'implémentations d'une méthode	47
6.6	La problématique de l'objet courant (this)	48
6.7	L'héritage	51
6.8	Le chaînage de méthodes	51
7.	Les principaux apports de la norme ECMAScript 6	52
7.1	La norme ECMAScript	52
7.2	Le mot réservé let	52
7.3	L'interpolation de variables dans les chaînes	53
7.4	Les paramètres par défaut	53
7.5	Une manipulation plus confortable des listes	53
7.5.1	La structure for (... of ...) ...	53
7.5.2	La méthode includes()	54

- 7.6 L'opérateur « fat arrow » (= >). . . . . 54
- 7.7 Les classes . . . . . 55
- 8. La programmation réactive, les observables et les promises . . . . . 55
  - 8.1 Premier exemple : un observable sur un bouton. . . . . 57
  - 8.2 Deuxième exemple : un observable sur un entier. . . . . 58
  - 8.3 Troisième exemple : un observable sur un timer . . . . . 59
  - 8.4 Les promises . . . . . 59

**Chapitre 3**

**Extensions JavaScript pour les classes**

- 1. Présentation des extensions à JavaScript . . . . . 61
- 2. Le langage TypeScript . . . . . 62
  - 2.1 Le transpiler tsc. . . . . 62
  - 2.2 Typage statique des variables . . . . . 63
    - 2.2.1 Les types de base . . . . . 63
    - 2.2.2 Typage de variables non scalaires. . . . . 63
    - 2.2.3 Le type enum . . . . . 64
    - 2.2.4 Le type générique any . . . . . 64
    - 2.2.5 Typage des fonctions. . . . . 64
  - 2.3 Les classes . . . . . 65
    - 2.3.1 L'héritage . . . . . 66
    - 2.3.2 Les interfaces . . . . . 67
    - 2.3.3 La généricité . . . . . 67
- 3. Le langage Dart . . . . . 68
  - 3.1 Installation et test de Dart. . . . . 68
  - 3.2 Génération du code JavaScript avec Dart . . . . . 69
  - 3.3 Les classes . . . . . 69
    - 3.3.1 L'héritage . . . . . 70
    - 3.3.2 Les interfaces . . . . . 71

# 4 **Angular et Node.js**

Optimisez le développement de vos applications web

## **Chapitre 4**

### **La plateforme Node.js**

1. Présentation de Node.js . . . . .	73
2. Installation et test de Node.js . . . . .	74
2.1 Création du fichier de test . . . . .	74
2.2 Installation et test de Node.js sous Ubuntu . . . . .	75
2.3 Installation et test de Node.js sous Windows . . . . .	75
2.4 Installation et test de Node.js sous Mac OS . . . . .	76
3. La modularité de Node.js . . . . .	76
3.1 Les modules et les packages . . . . .	76
3.1.1 Le gestionnaire de modules de Node.js : npm . . . . .	77
3.1.2 Spécification des dépendances : le fichier package.json . . . . .	77
3.2 Création d'un premier serveur Node.js de test . . . . .	78
3.3 Création et réutilisation d'un module . . . . .	80
3.4 Création d'un serveur renvoyant des données . . . . .	82
3.5 Le module express . . . . .	83
3.5.1 Gestion de routes REST avec le module express . . . . .	83
3.5.2 Gestion des templates avec le module express . . . . .	86
3.5.3 Spécification des dépendances dans un fichier package.json . . . . .	88
3.5.4 Installation du module express . . . . .	88
3.6 Le module fs (FileSystem) . . . . .	88
3.7 Test d'un serveur Node.js . . . . .	90
3.7.1 Création d'un fichier de données JSON . . . . .	90
3.7.2 La problématique du contrôle d'accès HTTP . . . . .	91
3.7.3 Renvoi au client d'un fichier JSON . . . . .	92
3.7.4 Paramétrage des routes . . . . .	93
3.7.5 Gestion des paramètres . . . . .	98
4. Sécurisation d'un serveur Node.js (protocole HTTPS) . . . . .	99
4.1 Création de l'autorité de certification . . . . .	100
4.2 Création du certificat . . . . .	100
4.3 Création du serveur . . . . .	101
5. Bilan des acquis de ce chapitre . . . . .	102



**Chapitre 5****Le SGBD NoSQL MongoDB**

1. Introduction . . . . .	103
2. Pourquoi utiliser une base de données NoSQL ? . . . . .	104
3. Présentation de MongoDB . . . . .	105
3.1 Les collections et les documents . . . . .	105
3.2 Les index . . . . .	106
4. Mise en œuvre de MongoDB . . . . .	106
4.1 Installation de MongoDB . . . . .	106
4.1.1 Installation de MongoDB sous Linux . . . . .	106
4.1.2 Installation de MongoDB sous Windows ou sous Mac OS . . . . .	107
4.1.3 Utilisation de MongoDB en lignes de commande . . . . .	107
4.2 Affichage de la liste des bases de données . . . . .	108
4.3 Création d'une base de données . . . . .	108
4.4 Affichage de la liste des collections . . . . .	108
4.5 Création d'une collection . . . . .	108
4.5.1 Insertion des documents dans une collection . . . . .	109
4.5.2 Importation de documents à partir d'un fichier . . . . .	109
4.5.3 Exportation des documents d'une collection dans un fichier JSON . . . . .	110
4.6 Interrogation d'une collection . . . . .	111
4.6.1 Interrogation via un objet « filtre » . . . . .	111
4.6.2 Les opérateurs de comparaison, les opérateurs ensemblistes et logiques . . . . .	112
4.6.3 L'opérateur \$exists . . . . .	113
4.6.4 L'opérateur \$in . . . . .	113
4.6.5 L'opérateur \$nin . . . . .	113
4.6.6 L'opérateur \$or . . . . .	114
4.6.7 L'opérateur \$not . . . . .	114
4.6.8 L'opérateur \$nor . . . . .	114
4.7 Application d'expressions régulières . . . . .	115

# 6 **Angular et Node.js**

Optimisez le développement de vos applications web

4.8	Les projections et la méthode <code>distinct()</code> . . . . .	115
4.8.1	Les projections . . . . .	115
4.8.2	La méthode <code>distinct()</code> . . . . .	116
4.9	Référencement des documents et jointures . . . . .	116
4.9.1	Les objets imbriqués (nested objects) . . . . .	117
4.9.2	Les objets référencés . . . . .	118
4.9.3	Les jointures . . . . .	119
4.10	Mise à jour et suppression d'un document . . . . .	123
4.10.1	Mise à jour d'un document . . . . .	123
4.10.2	Suppression d'un document . . . . .	124
4.11	Suppression d'une collection . . . . .	124
5.	Utilisation de MongoDB via Node.js . . . . .	124
5.1	Installation du module MongoDB pour Node.js . . . . .	124
5.2	Connexion au serveur MongoDB . . . . .	126
5.3	Insertion de données à partir d'un serveur Node.js . . . . .	127
5.4	Interrogation de données à partir d'un serveur Node.js . . . . .	127
5.4.1	Exploitation du résultat de la méthode <code>find()</code> . . . . .	128
5.4.2	Utilisation de la méthode <code>toArray()</code> . . . . .	129
5.5	Synchronisation des requêtes . . . . .	131
5.5.1	Utilisation des fonctions de callback . . . . .	132
5.5.2	Utilisation du module <code>async</code> . . . . .	134
5.5.3	La méthode <code>async.series()</code> . . . . .	135
5.5.4	La méthode <code>async.waterfall()</code> . . . . .	136
6.	Interrogation de MongoDB via les routes gérées par express . . . . .	138
6.1	La structure d'un serveur Node.js interrogeant MongoDB . . . . .	138
6.2	La problématique du cross-origin resource sharing . . . . .	139
6.3	Exemples de gestion de routes . . . . .	139
6.3.1	Gestion d'une route pour lister les marques . . . . .	140
6.3.2	Gestion d'une route pour filtrer les produits . . . . .	141
6.3.3	Recherche d'un produit à partir de son identifiant interne . . . . .	142

- 7. Le fil rouge du côté serveur ..... 144
  - 7.1 Création de la collection. .... 144
  - 7.2 Mise en place de deux recherches sur les produits ..... 146
    - 7.2.1 La superstructure du serveur. .... 146
    - 7.2.2 Gestion de la route qui filtre les documents sur différents critères. .... 148
    - 7.2.3 Gestion de la route qui renvoie un document via son identifiant interne. .... 149
    - 7.2.4 Exemples de requête sur le serveur. .... 149
- 8. Bilan des acquis de ce chapitre. .... 150

**Chapitre 6**

**Introduction au framework applicatif Angular**

- 1. Présentation d'Angular ..... 151
  - 1.1 Une évolution radicale d'AngularJS. .... 151
  - 1.2 La modularité de l'application : les modules et les composants ..... 152
    - 1.2.1 Les modules ..... 153
    - 1.2.2 Les composants et les services ..... 154
  - 1.3 Manipulation des composants comme des balises. .... 155
  - 1.4 Utilisation d'une extension à JavaScript (TypeScript ou Dart) ..... 156
- 2. Angular par rapport au framework MVC (voire MVVM). .... 156
- 3. Mise en place d'une application Angular ..... 158
  - 3.1 Présentation d'Angular CLI ..... 159
  - 3.2 Installation d'Angular CLI ..... 160
  - 3.3 Création d'un projet Angular avec Angular CLI. .... 160
  - 3.4 Structure des dossiers d'un projet Angular CLI ..... 162
  - 3.5 Un premier composant créé par Angular CLI. .... 163
  - 3.6 Le root module créé par Angular CLI ..... 166
  - 3.7 Mise à jour d'Angular via Angular CLI ..... 167
- 4. Génération du code JavaScript à partir de TypeScript. .... 168

# 8 **Angular et Node.js**

Optimisez le développement de vos applications web

5. Les décorateurs . . . . .	170
6. Création d'un nouveau composant qui affiche un message . . . . .	171
6.1 Création du composant . . . . .	171
6.1.1 Le template HTML . . . . .	173
6.1.2 La feuille de style . . . . .	173
6.2 Interfaçage du composant dans le composant racine . . . . .	173
6.3 Spécification des composants dans le module . . . . .	173
6.4 Activation du module. . . . .	174
6.5 La page web frontale . . . . .	175
7. Le cycle de vie d'un composant . . . . .	176
7.1 Le hook ngOnChanges() . . . . .	177
7.2 Le hook ngOnInit() . . . . .	178
7.3 Le hook ngDoCheck() . . . . .	178
7.4 Le hook ngOnDestroy() . . . . .	179
8. Bilan des acquis de ce chapitre. . . . .	180

## **Chapitre 7**

### **Angular : les templates, bindings et directives**

1. Les templates . . . . .	183
1.1 Imbrication des templates . . . . .	185
1.2 Les templates insérés (embedded templates) . . . . .	191
1.3 Les templates externalisés . . . . .	192
2. Data bindings entre le composant et le template. . . . .	192
2.1 Accès aux éléments du DOM . . . . .	193
2.2 Interpolation d'une variable dans un template. . . . .	193
2.3 Property binding . . . . .	195
2.4 Event binding . . . . .	197
2.5 Two-way data binding. . . . .	199
3. Les directives. . . . .	201
3.1 Les directives structurales . . . . .	202
3.1.1 La directive *ngFor . . . . .	203

- 3.1.2 La directive \*ngIf ..... 205
- 3.2 Les directives attributs ..... 206
- 3.3 Émission d'information d'un composant  
vers son père (@Output()) ..... 211
- 4. Les pipes ..... 214
- 5. Exemple de synthèse : un formulaire d'authentification ..... 218
- 6. Le fil rouge : création d'un composant qui affiche les produits ... 221
  - 6.1 Le composant ..... 222
    - 6.1.1 La classe implémentant le composant ..... 223
    - 6.1.2 Le template HTML ..... 224
    - 6.1.3 La feuille de style ..... 226
  - 6.2 Le module spécifiant le composant ..... 226
  - 6.3 Activation du module. .... 227
  - 6.4 La page web frontale ..... 227
  - 6.5 Lancement de l'application ..... 228
- 7. Bilan des acquis de ce chapitre. .... 228

**Chapitre 8**

**Angular et la connexion à Node.js : les services**

- 1. Introduction ..... 231
- 2. Injection de dépendances ..... 232
- 3. Utilisation des services pour le transfert de données ..... 233
  - 3.1 Récupération de données formatées en JSON ..... 233
  - 3.2 Envoi de données JSON au serveur ..... 234
  - 3.3 Envoi de données via la querystring ..... 236
- 4. Mise en œuvre des services dans le fil rouge ..... 237
  - 4.1 Déclaration des routes du côté serveur ..... 237
  - 4.2 Gestion des produits ..... 239
    - 4.2.1 Affichage des sélecteurs. .... 239
    - 4.2.2 Affichage des produits suivant  
des critères de recherche ..... 243

# 10 **Angular et Node.js**

Optimisez le développement de vos applications web

4.2.3	Affichage des produits associés à des mots-clés. . . . .	246
4.2.4	Accès à un produit par son identifiant. . . . .	249
4.3	Gestion du panier . . . . .	250
4.3.1	Affichage des identifiants des produits du panier . . . . .	250
4.3.2	Affichage de tous les produits du panier . . . . .	251
4.3.3	Ajout d'un produit au panier . . . . .	253
4.3.4	Suppression d'un produit du panier . . . . .	256
4.3.5	Réinitialisation du panier . . . . .	258
5.	La bibliothèque NgRx et les stores . . . . .	258
6.	Bilan des acquis de ce chapitre. . . . .	259

## **Chapitre 9**

### **Angular et la gestion des routes internes**

1.	Principe général du routage . . . . .	261
1.1	Pourquoi mettre en place un routage ? . . . . .	261
1.2	Les routes, le routeur, les tables de routage . . . . .	263
1.3	Les vues activées par les routes . . . . .	265
1.4	Exemple de routage . . . . .	267
1.5	Définition d'un arbre de vues. . . . .	270
1.6	Utilisation des outlets nommées. . . . .	275
2.	La syntaxe des routes . . . . .	280
2.1	Les deux syntaxes d'une route : chaîne ou link parameters array . . . . .	281
2.2	Les routes absolues et les routes relatives . . . . .	282
2.3	Paramétrage des routes. . . . .	282
2.4	Association d'une route à une auxiliary outlet . . . . .	283
3.	Sélection des routes . . . . .	284
3.1	La directive routerLink . . . . .	284
3.2	La méthode navigate() . . . . .	286
3.3	Exemple de route . . . . .	287

- 4. Gestion des routes du contrôleur vers le composant cible . . . . . 287
  - 4.1 Configuration de la table de routage . . . . . 288
  - 4.2 Les propriétés d'une route . . . . . 289
  - 4.3 Prise en charge d'une route  
par plusieurs modules/tables de routage . . . . . 291
  - 4.4 Contrôle des routes : les guards . . . . . 293
  - 4.5 Invocation d'un composant . . . . . 301
  - 4.6 Capture d'une route lors de l'invocation d'un composant . . . 303
- 5. Gestion de routes dans le fil rouge . . . . . 306
  - 5.1 Le module de routage associé au root module . . . . . 307
  - 5.2 Le module de routage associé au feature module research . . . 308
  - 5.3 Le module de routage associé au feature module cart . . . . . 309
- 6. Bilan des acquis de ce chapitre . . . . . 310

**Chapitre 10**  
**Angular et la visualisation d'informations**

- 1. Introduction . . . . . 313
- 2. Création de charts avec D3.js et dc.js . . . . . 314
  - 2.1 Installation de D3.js . . . . . 315
  - 2.2 Le langage SVG . . . . . 315
  - 2.3 Génération d'éléments graphiques  
associés aux objets d'une collection . . . . . 318
  - 2.4 Sélection et modification d'éléments du DOM . . . . . 319
    - 2.4.1 Ajout d'éléments graphiques . . . . . 320
    - 2.4.2 Remplacement d'éléments graphiques . . . . . 321
  - 2.5 Mise en œuvre d'écouteurs d'événements . . . . . 322
  - 2.6 Intégration de D3.js dans Angular . . . . . 323
    - 2.6.1 Un serveur virtuel de données commerciales . . . . . 323
    - 2.6.2 Le service accédant aux données du serveur . . . . . 325
    - 2.6.3 Le template du composant  
qui affiche un histogramme . . . . . 325

# 12 \_\_\_\_\_ Angular et Node.js

Optimisez le développement de vos applications web

2.6.4	Implémentation du composant qui affiche un histogramme . . . . .	326
2.7	Les bibliothèques dc.js et Crossfilter . . . . .	330
2.7.1	Installation de dc.js et de Crossfilter . . . . .	331
2.7.2	Implémentation du composant affichant l'histogramme . . . . .	331
3.	Intégration de cartes Google Map dans un projet Angular . . . . .	333
3.1	Installation des prérequis techniques . . . . .	334
3.1.1	Installation des types TypeScript . . . . .	334
3.1.2	Installation de la bibliothèque PrimeNG . . . . .	335
3.2	Présentation d'une carte Google Map statique . . . . .	335
3.3	Un composant PrimeNG pour gérer une carte Google Map . . . . .	337
3.4	Gestion d'un chart associé à une Google Map . . . . .	341
4.	Mise en œuvre de composants graphiques . . . . .	346
4.1	Autre exemple de composant PrimeNG (Calendar) . . . . .	346
4.2	La bibliothèque de composants Material . . . . .	348
4.3	La bibliothèque ngx-bootstrap . . . . .	350
5.	Bilan des acquis de ce chapitre . . . . .	355

## Chapitre 11 Test et déploiement

1.	Test . . . . .	357
2.	Déploiement . . . . .	361
2.1	Déploiement avec Apache . . . . .	361
2.2	Déploiement avec Node.js . . . . .	362
2.3	Déploiement avec http-server (raccourci sur Node.js) . . . . .	363
3.	Pour aller plus loin . . . . .	364
	Index . . . . .	365