

Les éléments à télécharger sont disponibles à l'adresse suivante :
<http://www.editions-eni.fr>
Saisissez la référence ENI de l'ouvrage **RITYP** 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. Un peu d'histoire. | 9 |
| 2. ECMAScript | 13 |
| 3. Pourquoi TypeScript? | 17 |
| 3.1 Transpilation | 18 |
| 3.2 Typage statique | 21 |
| 4. Les dessous du typage. | 23 |
| 4.1 Les fichiers de définition | 23 |
| 4.2 Typage structurel | 24 |
| 5. L'architecture de TypeScript | 27 |
| 5.1 Core TypeScript Compiler. | 28 |
| 5.2 Standalone TS Compiler | 28 |
| 5.3 Language Service. | 28 |
| 5.4 Options de compilation | 29 |
| 6. Environnement de développement | 30 |
| 6.1 Node.js. | 30 |
| 6.2 Visual Studio Code. | 32 |
| 6.3 Installer TypeScript | 38 |

Chapitre 2

Types et instructions basiques

| | |
|---|----|
| 1. Variable et portée | 41 |
| 2. Types basiques | 45 |
| 2.1 Introduction | 45 |
| 2.2 Types primitifs basiques | 45 |
| 2.3 Types primitifs : undefined et null | 49 |
| 2.4 Types par référence : array et tuples | 52 |
| 2.5 Types par référence : object | 54 |
| 2.6 Types spéciaux : any et void | 56 |
| 2.7 Affirmation de type | 58 |
| 2.8 Type spécial : unknown | 59 |
| 3. Décomposition | 62 |
| 4. Énumération | 65 |
| 5. Conditionnelle | 72 |
| 6. Opérateurs | 74 |
| 6.1 Opérateurs arithmétiques | 74 |
| 6.2 Opérateurs unaires | 75 |
| 6.3 Opérateurs de comparaison | 76 |
| 6.4 Opérateurs logiques combinatoires | 76 |
| 7. Zoom sur les opérateurs | 77 |
| 7.1 Opérateurs unaires + et - | 77 |
| 7.2 Opérateur unaire ! | 78 |
| 7.3 Opérateurs == et === | 78 |
| 7.4 Opérateurs et && | 79 |
| 7.5 Opérateur de décomposition | 81 |
| 8. Boucles | 82 |
| 8.1 Boucles for...in | 83 |
| 8.2 Boucle for...of | 84 |
| 9. Symbol | 85 |

| | |
|---|-----|
| 10. Itération et collections | 90 |
| 10.1 Iterator | 90 |
| 10.2 Générateurs | 92 |
| 10.3 Map et Set | 92 |
| 10.3.1Map | 92 |
| 10.3.2Set | 93 |
| 10.3.3WeakMap et WeakSet | 94 |
| 11. Fonctions | 95 |
| 11.1 Les bases | 95 |
| 11.2 Paramètre rest | 97 |
| 11.3 Les closures | 98 |
| 11.4 This et les fonctions fléchées | 100 |
| 11.5 Type fonction | 104 |
| 12. Prototype | 105 |
| 13. Gestion des exceptions | 106 |

Chapitre 3

La programmation orientée objet

| | |
|--|-----|
| 1. Introduction | 109 |
| 2. Les classes | 110 |
| 3. Les propriétés | 113 |
| 4. Les méthodes | 116 |
| 5. Les constructeurs | 122 |
| 6. Le statisme | 124 |
| 7. L'accessibilité des membres | 126 |
| 8. L'encapsulation | 132 |
| 9. L'héritage | 135 |
| 10. L'abstraction | 142 |
| 11. Les interfaces | 145 |

| | |
|--|-----|
| 12. Le polymorphisme | 156 |
| 13. Les principes SOLID | 158 |
| 13.1 Introduction aux principes SOLID | 158 |
| 13.2 Le principe de responsabilité unique | 158 |
| 13.3 Le principe d'ouverture à l'extension, fermeture à la modification | 161 |
| 13.4 Le principe de substitution de Liskov | 164 |
| 13.5 Le principe de ségrégation des interfaces | 169 |
| 13.6 Le principe d'inversion des dépendances | 171 |

Chapitre 4

Les modules

| | |
|------------------------------------|-----|
| 1. Introduction | 175 |
| 2. Historique | 176 |
| 2.1 Pattern module | 176 |
| 2.2 AMD et CommonJs | 178 |
| 2.3 Standardisation | 183 |
| 3. La norme ECMAScript 2015 | 183 |
| 3.1 Export | 187 |
| 3.2 Import | 191 |
| 4. Chargement et résolutions | 194 |
| 4.1 Résolutions | 194 |
| 4.2 Chargement | 197 |

Chapitre 5

La généricité

| | |
|--------------------------------|-----|
| 1. Introduction | 199 |
| 2. Déclaration de base | 199 |
| 3. Classes et interfaces | 202 |
| 4. Contraintes | 205 |

5. Générique et constructeur 211

Chapitre 6
Les décorateurs

1. Introduction 213
2. Décorateur de classe 215
3. Décorateur de méthode 217
4. Décorateur de propriété 220
5. Décorateur de paramètre 225
6. Fabrique de décorateurs 227
7. Métadonnées..... 229

Chapitre 7
Asynchronisme

1. Introduction 239
2. Callback hell 240
3. Promesse 242
 3.1 Historique 242
 3.2 Promesses ECMAScript 2015..... 243
 3.2.1 Résolution des promesses..... 243
 3.2.2 Then, catch et finally 245
 3.2.3 Promise.all et Promise.race..... 248
 3.2.4 Conclusion 249
4. Async/await 250

Chapitre 8

Système de types avancés

| | |
|---|-----|
| 1. Introduction | 255 |
| 2. Alias de type | 255 |
| 3. Type never | 259 |
| 4. Type union et intersection. | 261 |
| 4.1 Union. | 261 |
| 4.2 Intersection. | 263 |
| 5. Type guards | 266 |
| 5.1 Introduction | 266 |
| 5.2 Opérateur typeof | 266 |
| 5.3 Opérateur instanceof | 268 |
| 5.4 Opérateur in | 270 |
| 5.5 Union discriminante | 271 |
| 5.6 Type guards définis par l'utilisateur | 273 |
| 5.7 Gestion de null et undefined | 276 |
| 5.8 Cas impossible | 279 |
| 6. Types littéraux | 280 |
| 7. Type index | 282 |
| 8. Mapped type. | 285 |
| 8.1 Avant TypeScript 2.1 | 285 |
| 8.2 Les opérateurs | 287 |
| 8.3 Mapped type natif | 289 |
| 9. Affirmation constante | 292 |
| 10. Type conditionnel | 294 |
| 10.1 Les bases | 294 |
| 10.2 Distributivité | 296 |
| 10.3 Opérateur infer | 298 |
| 10.4 Type conditionnel natif | 299 |

Chapitre 9
TypeScript et la programmation fonctionnelle

- 1. Introduction 303
- 2. Fonction pure et impure. 306
- 3. Immutabilité. 309
- 4. Itération. 316
- 5. Conditions. 320
- 6. Fonction partielle 321
- 7. Currying 322
- 8. Pattern Matching 324
- 9. Composition de fonctions 328
- 10. Pour aller plus loin... 333

Chapitre 10
Un premier projet avec Node.js

- 1. Introduction 335
- 2. Préparation de la base de données 338
- 3. Mise en place du projet 342
 - 3.1 Création du fichier package.json 343
 - 3.2 Mise en place de TypeScript 344
 - 3.2.1 Installation 344
 - 3.2.2 Configuration de TypeScript 345
 - 3.3 Amélioration de l'environnement de développement 349
 - 3.3.1 Exécution de l'application 349
 - 3.3.2 Débogage 351
 - 3.4 Une première application. 353
 - 3.4.1 Mise en place de la bibliothèque Express.js. 353
 - 3.4.2 Gestion des variables d'environnement. 356
 - 3.4.3 Une première requête SQL. 357

| | |
|---|-----|
| 4. Création du Framework MVC | 359 |
| 4.1 Registre des routes | 360 |
| 4.2 Décorateurs de routes. | 366 |
| 4.2.1 Le décorateur Controller | 366 |
| 4.2.2 Le décorateur Route | 368 |
| 4.3 Mise en œuvre du Framework MVC | 370 |
| 4.3.1 Création du contrôleur. | 370 |
| 4.3.2 Création des routes. | 372 |
| 4.4 Middleware Express.js | 374 |
| 5. Accès aux données | 377 |
| 5.1 Entité métier | 378 |
| 5.2 Dépôt de données abstrait | 379 |
| 5.2.1 Dépôt. | 379 |
| 5.2.2 Type d'opération. | 380 |
| 5.2.3 Dépôt de données abstrait | 380 |
| 5.3 Définition des modèles métiers | 388 |
| 5.4 Un premier dépôt de données : EmployeeRepository | 389 |
| 5.5 Utilisation du dépôt de données EmployeeRepository | 391 |
| 6. Injection des dépendances | 393 |
| 6.1 Service de gestion des dépendances | 393 |
| 6.2 Registre de clé | 396 |
| 6.3 Décorateurs de gestion des dépendances | 397 |
| 6.4 Résolution des dépendances lors de la création des contrôleurs | 401 |
| 6.5 Enregistrement d'une dépendance. | 403 |
| 7. Finalisation de l'API | 405 |
| 7.1 Ajout d'un employé | 405 |
| 7.2 Gestion des équipes | 414 |
| 7.3 Changer l'équipe d'un employé | 420 |
| 8. Conclusion de l'exercice | 426 |
| Index | 427 |

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

Avant-propos

- 1. React 9
- 2. À qui s'adresse ce livre ? 10
- 3. Que trouverez-vous dans ce livre ? 10
- 4. À propos des exemples 11
- 5. Rester informé et en savoir plus 12

Chapitre 1 Découverte de React

- 1. Installation des outils requis 13
 - 1.1 NodeJS 13
 - 1.2 Un éditeur de texte 14
 - 1.3 Et ensuite ? 14
- 2. Création du premier projet 14
 - 2.1 Explication du code 16
 - 2.2 Composants et propriétés 18
- 3. Le langage JSX 19
- 4. Un composant par fichier 23
- 5. Ajouter du style 26
- 6. Prochaines étapes 29

Chapitre 2

Ajouter du comportement aux composants

| | |
|---|----|
| 1. Conserver un état local dans le composant | 31 |
| 2. Réagir aux actions et entrées de l'utilisateur | 36 |
| 3. Requêtes Ajax et cycle de vie des composants React | 45 |
| 4. Simplifier les composants grâce aux hooks | 49 |
| 4.1 Gérer le state local avec useState | 50 |
| 4.2 Optimiser le rendu avec useCallback | 51 |
| 4.3 Requête Ajax au démarrage avec useEffect | 53 |
| 4.4 Un point d'attention important sur les hooks | 55 |
| 5. Déclarer et typer les propriétés des composants | 56 |
| 6. En conclusion | 59 |

Chapitre 3

Concevoir une application avec Redux

| | |
|---|----|
| 1. Introduction | 61 |
| 2. Découverte de Redux | 62 |
| 2.1 Concepts de base | 62 |
| 2.1.1 Le state | 62 |
| 2.1.2 Les actions | 62 |
| 2.1.3 Le reducer | 63 |
| 2.1.4 Le store | 63 |
| 2.2 Premier exemple | 64 |
| 3. Utilisation avec React | 69 |
| 3.1 Découverte de React-Redux | 69 |
| 3.2 Connecter un composant à Redux à l'aide des hooks | 74 |
| 4. Actions complexes et asynchrones | 76 |
| 5. Un exemple complet | 84 |
| 5.1 Les services | 84 |
| 5.2 Les composants | 90 |

6. Conclusion 93

Chapitre 4
Gérer les effets de bord avec Redux-Saga

1. Introduction 95
2. Les générateurs 96
3. Les effets de Redux-Saga 102
4. Un exemple plus complet. 107
5. Conclusion 116

Chapitre 5
Développer pour le mobile avec React Native

1. Présentation de React Native. 117
 1.1 Un peu d’historique 117
 1.2 Outils utilisés 119
2. Une première application. 120
 2.1 Génération de l’application et premier lancement. 120
 2.2 Premiers composants 124
 2.3 Gérer des entrées de l’utilisateur 128
 2.3.1 Gestion de la navigation. 133
3. Affichage de listes. 140

Chapitre 6
Fonctionnalités avancées avec React Native

1. Utiliser une fonctionnalité native : l’appareil photo 145
2. Ajouter Redux à l’application 149

| | |
|--|-----|
| 3. Plus loin avec la navigation | 152 |
| 3.1 Une modale pour l'édition d'un contact | 153 |
| 3.2 Intégration de la prise de photo | 157 |
| 4. Conclusion | 159 |

Chapitre 7

Gestion de formulaires et du routage

| | |
|---|-----|
| 1. Introduction | 161 |
| 2. Création de formulaires avec Formik | 163 |
| 2.1 Premier formulaire : inscription d'un utilisateur | 164 |
| 2.2 Deuxième formulaire : création/modification d'une dépense . | 170 |
| 3. Gestion du routage avec React Router | 179 |
| 3.1 Refactoring et définition des routes | 179 |
| 3.2 Ajout du routage avec React Router | 182 |
| 3.3 Persister des données dans le navigateur | 187 |
| 3.4 Ajout d'un nouvel écran | 188 |
| 4. Prochaines étapes | 192 |

Chapitre 8

Sécurité et persistance avec Firebase

| | |
|--|-----|
| 1. Découverte de Firebase | 195 |
| 2. Gestion de l'authentification | 197 |
| 2.1 Inscription d'un utilisateur | 197 |
| 2.2 Connexion d'un utilisateur | 200 |
| 2.3 Gestion de l'authentification dans l'application | 201 |
| 3. Persistance de données avec Firebase | 209 |
| 4. En conclusion | 215 |

Chapitre 9
Connecter React à une API GraphQL

- 1. Présentation de GraphQL et premières requêtes 217
 - 1.1 Qu'est-ce que GraphQL ? 218
 - 1.2 Premières requêtes avec l'API de GitHub 218
 - 1.3 Ajout de données liées à la requête 221
 - 1.4 Écrire des données. 222
 - 1.5 Prochaines étapes 223
- 2. Création d'une API avec Graphcool 224
 - 2.1 Installation et création du projet Graphcool 224
 - 2.2 Ajout de nouveaux modèles et relations. 227
- 3. Appel d'une API avec React et Apollo Client 231
 - 3.1 Lire des données en envoyant des queries. 233
 - 3.2 Gestion de l'authentification. 238
 - 3.3 Écrire des données à l'aide de mutations. 240

Chapitre 10
Écrire des composants réutilisables

- 1. Introduction 245
- 2. Principes pour des composants réutilisables. 246
 - 2.1 Des composants aussi simples que possible 246
 - 2.2 Des composants pour une interface homogène 248
 - 2.3 Sortir la logique des composants 249
 - 2.4 Limiter le state local et les effets de bord 250
 - 2.5 En conclusion 252
- 3. Les high-order components 253
 - 3.1 Exemple : champ de saisie d'un numéro de téléphone. 254
 - 3.1.1 Premier HOC : n'accepter que certains caractères 256
 - 3.1.2 Deuxième HOC : mettre en forme la valeur entrée 257
 - 3.2 En conclusion 260
- 4. Les render props 260

| | |
|--|-----|
| 5. Les hooks | 266 |
| 6. Les contextes et le pattern Provider/Consumer | 268 |
| 7. Conclusion | 274 |

Chapitre 11

Tester une application React

| | |
|---|-----|
| 1. Que tester dans une application React ? | 277 |
| 2. Test unitaire de composants avec Enzyme | 278 |
| 2.1 Initialisation de l'application à tester | 278 |
| 2.2 Jest et Enzyme | 280 |
| 2.3 Écriture des tests du composant | 283 |
| 2.4 Spécificité du shallow rendering d'Enzyme | 288 |
| 3. Tester le store Redux et les sagas avec Redux Saga Test Plan | 290 |
| 4. En conclusion | 297 |

Conclusion

| | |
|---|-----|
| 1. Aller plus loin | 299 |
| 2. Reason : un autre langage pour faire du React | 299 |
| 3. Générer des sites statiques avec Gatsby | 300 |
| 4. Des bibliothèques de composants utiles | 302 |
| 4.1 Material-UI et Paper | 302 |
| 4.2 React-Select et React-Table | 303 |
| 4.3 React-Intl pour internationaliser votre application | 303 |
| 5. Le mot de la fin | 304 |

Table des matières _____ 7

Annexes

| | |
|---|-----|
| 1. Utiliser les React Dev Tools | 305 |
| 2. Construire et déployer une application React | 306 |
| 3. Création d'une application Firebase | 307 |

| | |
|-----------------|-----|
| Index | 313 |
|-----------------|-----|