

Les éléments à télécharger sont disponibles à l'adresse suivante :

<http://www.editions-eni.fr>

Saisissez la référence ENI de l'ouvrage **RI2TYP** 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	17
3.2	Typage statique	20
4.	Les dessous du typage	22
4.1	Les fichiers de définition	22
4.2	Typage structurel	24
5.	L'architecture de TypeScript	27
5.1	Core TypeScript Compiler	27
5.2	Standalone TS Compiler	28
5.3	Language Service	28
5.4	Options de compilation	28
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	55
2.6	Types spéciaux : any et void	58
2.7	Affirmation de type	60
2.8	Type spécial : unknown	61
3.	Décomposition	64
4.	Énumération	68
5.	Conditionnelle If...Else	75
6.	Optional chaining	77
7.	Opérateurs	78
7.1	Opérateurs arithmétiques	79
7.2	Opérateurs unaires	79
7.3	Opérateurs de comparaison	80
7.4	Opérateurs logiques combinatoires	81
8.	Zoom sur les opérateurs	81
8.1	Opérateurs unaires + et -	82
8.2	Opérateur unaire !	82
8.3	Opérateurs == et ===	83
8.4	Opérateurs et &&	84
8.5	Opérateur de coalescence null (??)	86
8.6	Opérateur de décomposition	86
9.	Boucles	88
9.1	Boucles for...in	88
9.2	Boucle for...of	90

10. Symbol	91
11. Itération et collections	95
11.1 Iterator	95
11.2 Générateurs	97
11.3 Map et Set	98
11.3.1 Map	98
11.3.2 Set	99
11.3.3 WeakMap et WeakSet	100
12. Fonctions	100
12.1 Les bases	100
12.2 Paramètre rest	102
12.3 Les closures	105
12.4 This et les fonctions fléchées	106
12.5 Type fonction	111
13. Prototype	112
14. Gestion des exceptions	113

Chapitre 3

La programmation orientée objet

1. Introduction	117
2. Les classes	118
3. Les propriétés	121
4. Les méthodes	124
5. Les constructeurs	130
6. Le statisme	132
7. L'accessibilité des membres	135
8. L'encapsulation	143
9. L'héritage	146
10. L'abstraction	153
11. Les interfaces	157

12. Le polymorphisme	168
13. Les principes SOLID	170
13.1 Introduction aux principes SOLID	170
13.2 Le principe de responsabilité unique	170
13.3 Le principe d'ouverture à l'extension, fermeture à la modification	173
13.4 Le principe de substitution de Liskov	176
13.5 Le principe de ségrégation des interfaces	181
13.6 Le principe d'inversion des dépendances	183

Chapitre 4 Les modules

1. Introduction	187
2. Historique	188
2.1 Pattern module	188
2.2 AMD et CommonJs	190
2.3 Standardisation	195
3. La norme ECMAScript 2015	195
3.1 Export	199
3.2 Import	203
3.3 Import assert	206
4. Gestion des types	206
5. Chargement et résolutions	208
5.1 Résolutions	208
5.2 Chargement	211

Chapitre 5 La généricité

1. Introduction	213
2. Déclaration de base	213
3. Classes et interfaces	216

4. Contraintes	219
5. Générique et constructeur	225

Chapitre 6

Les décorateurs

1. Introduction	227
2. Décorateurs expérimentaux	229
2.1 Décorateurs expérimentaux de classe	230
2.2 Décorateurs expérimentaux de méthode	232
2.3 Décorateurs expérimentaux de propriété	235
2.4 Décorateurs expérimentaux de paramètre	239
2.5 Fabrique de décorateurs	242
2.6 Métadonnées	244
3. Décorateurs ECMAScript	253
3.1 Décorateurs ECMAScript de classe	255
3.2 Décorateurs ECMAScript de méthode	257
3.3 Décorateurs ECMAScript de propriété	259
3.4 Décorateurs ECMAScript d'accesseurs	262
3.4.1 Décorateur "getter"	262
3.4.2 Décorateur "setter"	264
3.4.3 Décorateur d'accesseurs automatiques	265

Chapitre 7

Asynchronisme

1. Introduction	271
2. Callback hell	272
3. Promesse	274
3.1 Historique	274
3.2 Promesses ECMAScript 2015	275
3.2.1 Résolution des promesses	275
3.2.2 Then, catch et finally	277
3.2.3 Méthodes statiques	282

3.2.4 Conclusion	283
4. Async/await	284

Chapitre 8

Système de types avancés

1. Introduction	289
2. Alias de type	289
3. Type never	293
4. Type union et intersection	295
4.1 Union	295
4.2 Intersection	297
5. Type guards	300
5.1 Introduction	300
5.2 Opérateur typeof	301
5.3 Opérateur instanceof	302
5.4 Opérateur in	304
5.5 Union discriminante	307
5.6 Type guards définis par l'utilisateur	309
5.7 Gestion de null et undefined	312
5.8 Cas impossible	315
5.9 Fonction d'affirmation	316
6. Types littéraux	318
7. Modèle de type littéral	320
8. Type index	324
9. Mapped type	329
9.1 Avant TypeScript 2.1	329
9.2 Les opérateurs	332
9.3 Remappage des clés	333
9.4 Mapped type natif	334
10. Affirmation constante	337
11. Tuples variadiques	339

12. Type conditionnel	341
12.1 Les bases	341
12.2 Distributivité	343
12.3 Opérateur infer	346
12.4 Type conditionnel natif	347
13. Opérateur satisfies	351

Chapitre 9**TypeScript et la programmation fonctionnelle**

1. Introduction	355
2. Fonction pure et impure.	357
3. Immutabilité.	361
4. Itération.	368
5. Conditions.	372
6. Fonction partielle	373
7. Currying	374
8. Pattern Matching	377
9. Composition de fonctions	380
10. Pour aller plus loin...	384

Chapitre 10**Un premier projet avec Node.js**

1. Introduction	385
2. Mise en place du projet	386
2.1 Création du fichier package.json	387
2.2 Mise en place de TypeScript	388
2.2.1 Installation	388
2.2.2 Configuration de TypeScript	389
2.3 Une première application.	396

3.	Création du Framework MVC	399
3.1	Registre des routes	400
3.2	Fabrique de contrôleurs	403
3.3	Le décorateur Controller	405
3.4	La fabrique de décorateurs Action	406
3.5	Création du contrôleur EmployeeController	408
3.6	Initialisation de l'application	410
4.	Validation et OpenApi	415
4.1	Typage des schémas	416
4.2	Le décorateur de propriété SchemaProperty	421
4.3	Liaison entre une action et un modèle	422
4.4	Le décorateur de méthode Model	425
4.5	Gestion des schémas dans la classe Server	428
4.6	L'explorateur d'API	429
4.7	L'action post du contrôleur EmployeeController	430
5.	Accès et persistance des données	435
5.1	Typage des entités métier	436
5.2	Persistance des données	437
5.3	Repository	439
6.	Inversion des dépendances	444
6.1	Le conteneur de dépendances	445
6.2	Le décorateur Inject	448
6.3	Injection des dépendances	449
6.4	Stockage des données dans un fichier	451
7.	Allez plus loin	453
	Index	455

Les éléments à télécharger sont disponibles à l'adresse suivante :

<http://www.editions-eni.fr>

Saisissez la référence de l'ouvrage **EIREACT** dans la zone de recherche et validez. Cliquez sur le titre du livre puis sur le bouton de téléchargement.

Chapitre 1

Avant-propos

1. Introduction	17
1.1 À qui s'adresse ce livre	17
1.2 Pourquoi choisir React	18
1.2.1 Réactivité	18
1.2.2 Composabilité	18
1.2.3 Communauté active	18
1.2.4 React Native	19
1.3 Comment utiliser ce livre	19
1.4 À propos des exemples de code	20
2. Qu'est-ce que React ?	20
2.1 Introduction à React	20
2.1.1 Origine de React	20
2.1.2 Objectif de React	20
2.1.3 Pourquoi React est-il si populaire ?	20
2.1.4 Principaux concepts de React	21
2.2 Philosophie de React	22
2.2.1 Composants : l'unité de base	22
2.2.2 Réactivité : mise à jour efficace de l'interface utilisateur	23
2.2.3 Composition : assemblage de composants réutilisables	23
2.2.4 Une approche déclarative	23
2.3 La communauté React	24
2.3.1 Projets open source	24
2.3.2 Contributeurs engagés	25
2.3.3 Événements et conférences	25

2.4	Versions de React et cycle de mise à jour	25
2.4.1	Les différentes versions de React	25
2.4.2	Changements majeurs et améliorations	26
2.4.3	Suivre les nouvelles versions	26
3.	Structure du livre	27
3.1	Découverte de React	27
3.2	Gestion de l'état dans React	27
3.3	Développement mobile avec React Native	27
3.4	Aller plus loin avec React	28

Chapitre 2

Bien démarrer avec React

1.	Notions essentielles de JavaScript	29
1.1	Le DOM (Document Object Model)	30
1.2	Définir une variable en JavaScript	31
1.3	Les types de valeurs	32
1.4	Chaînes de caractères	34
1.5	Tableaux	35
1.6	Objets	36
1.7	Classes	38
1.8	Fonctions fléchées	40
1.8.1	Syntaxe des fonctions fléchées	40
1.8.2	Comparaison entre une fonction traditionnelle et une fonction fléchée	41
1.8.3	Utilisation des fonctions fléchées dans React	41
1.9	Asynchronisme	42
1.9.1	Callbacks	42
1.9.2	Promesses (Promises)	43
1.9.3	Async/Await	44

2. Prérequis, concepts et outils pour développer	44
2.1 Installation des outils nécessaires	44
2.1.1 NodeJS	44
2.1.2 Éditeur	45
2.1.3 Navigateur web et outils de développement	46
3. Une application basique	47
3.1 Le minimum de code pour une application React	47
3.2 Les composants React	48
4. Environnement de développement	50
4.1 Introduction au JSX	50
4.2 Initialisation d'un projet avec Vite	51
4.3 Mise en place de Prettier dans VSCode	51
4.4 Configuration d'ESLint	54
4.5 Conclusion	55

Chapitre 3

Découvrir le JSX

1. Introduction au JSX	57
2. Syntaxe et éléments JSX	60
2.1 Éléments JSX	60
2.2 Attributs JSX	61
2.2.1 Attributs booléens	61
2.2.2 Attributs personnalisés	62
2.3 Éléments React et fragments	63
2.4 Commentaires JSX	64
2.5 Utilisation de JavaScript dans le JSX	65
3. Construire une interface avec des composants	65
3.1 Composition	66
3.2 Isolation des fonctionnalités	67
3.3 Profondeur de la composition	70

3.4 Bonnes pratiques	71
3.4.1 Découpage en composants réutilisables	72
3.4.2 Profondeur de composition raisonnable	72
3.4.3 Respect du principe de responsabilité unique.	72
4. Les fragments JSX.	72
4.1 Utilisation des fragments JSX	73
4.2 Fragment explicite	74
5. Expressions conditionnelles	74
5.1 L'opérateur ternaire	75
5.2 L'instruction if	75
6. Utiliser les listes et les clés	76
6.1 Les clés des éléments de liste	76
6.1.1 Pourquoi les clés sont-elles importantes ?	76
6.1.2 Comment attribuer des clés ?	77
6.2 Utiliser une liste	77
6.3 Les clés et les performances	79

Chapitre 4

Les fondamentaux de React

1. Les composants.	81
1.1 Introduction aux composants	81
1.1.1 Types de composants.	81
1.1.2 Créer un composant.	82
1.1.3 Importer un composant.	83
1.2 Utilisation des props	84
1.2.1 Passer des props à un composant	84
1.2.2 Utilisation des props dans le composant enfant	85
1.3 Différences entre composants fonctionnels et de classe	86
1.4 Cycle de vie des composants de classe	87

1.5	Composants contrôlés et non contrôlés	88
1.5.1	Composants contrôlés	88
1.5.2	Composants non contrôlés	89
1.5.3	Quand utiliser chaque approche ?	90
1.6	Composants à état (stateful) et sans état (stateless)	90
1.6.1	Composants à état	91
1.6.2	Composants sans état	91
1.6.3	Quand utiliser chaque type de composant ?	92
1.7	Composition et hiérarchie des composants	92
2.	Les props et l'état local	93
2.1	Introduction aux props et à l'état local	93
2.1.1	Props	94
2.1.2	Le state	95
2.2	Utilisation des props	97
2.2.1	Passer des données de parent à enfant	97
2.2.2	Utiliser des props par défaut	98
2.3	L'état local	99
2.3.1	Utilisation du hook useState pour gérer l'état local	99
2.3.2	Modifier l'état local avec setState (pour les composants de classe)	100
2.3.3	Comparaison entre props et état local	102
3.	Les hooks	103
3.1	Introduction aux hooks	103
3.2	Les problèmes avec les composants de classe	103
3.3	Les principaux hooks intégrés	104
3.3.1	useState	104
3.3.2	useEffect	105
3.3.3	useContext	107
3.3.4	useReducer	109
3.3.5	useCallback	110
3.3.6	useMemo	112
3.3.7	useRef	114
3.3.8	useImperativeHandle	115

3.3.9 useLayoutEffect	117
3.3.10useDebugValue.....	118
3.4 Création de vos propres hooks.....	120
3.5 Bonnes pratiques pour l'utilisation des hooks	121
3.5.1 Respecter l'ordre des hooks et ne pas les appeler conditionnellement	123
3.5.2 Diviser les responsabilités des hooks pour éviter la complexité	124
3.6 Limitations des hooks	126
4. Le style	126
4.1 Introduction au style en React	126
4.1.1 Utilisation des styles en ligne (Inline Styles)	126
4.1.2 Utilisation de classes CSS	127
4.1.3 Utilisation de modules CSS	128
4.1.4 Utilisation de bibliothèques de styles (ex. Bootstrap, Material-UI)	130
4.2 Tailwind	131
4.3 La gestion des classes conditionnelles.....	132
4.4 Les animations et transitions.....	136
5. Le routage	138
5.1 Introduction au routage en React	138
5.1.1 Comprendre le concept de routage dans une application web	138
5.1.2 React Router.....	139
5.2 Utilisation des paramètres de l'URL (URL Parameters).....	140
6. Les formulaires	141
6.1 Utilisation des formulaires contrôlés	141
6.1.1 Gestion de l'état du formulaire avec useState.....	142
6.1.2 Validation des champs du formulaire	143
6.1.3 Soumission du formulaire	144

6.2 Utilisation des formulaires non contrôlés	146
6.2.1 Définition des formulaires non contrôlés	146
6.2.2 Gestion des données du formulaire avec les références (refs)	147
6.2.3 Accès aux valeurs du formulaire soumises	148
6.3 Utilisation des composants de formulaire	149
6.3.1 Présentation de Formik	149
6.3.2 Installation et configuration de Formik	150
6.3.3 Création d'un formulaire avec Formik	150
6.3.4 Validation des champs avec Formik	151
6.4 Les champs de formulaire avancés	153
6.4.1 Champ de sélection (select)	153
6.4.2 Cases à cocher (checkbox) et boutons radio (radio button)	154
6.4.3 Champ de fichier (file input)	156
6.4.4 Champs de date (date picker)	157
6.5 La validation des formulaires	159
6.6 React-hook-form, l'alternative à Formik	161
6.7 La gestion des formulaires complexes	162
6.7.1 Formulaires avec champs dépendants	162
6.7.2 Formulaires avec étapes	163
6.7.3 Formulaires avec listes dynamiques (ajout/suppression de champs)	166
7. Optimiser les performances d'une application	169
7.1 Comprendre les enjeux de performance	169
7.1.1 Réactivité de l'interface utilisateur	170
7.1.2 Vitesse de chargement initial	170
7.1.3 Gestion de l'état	170
7.1.4 Rendu non nécessaire	170
7.1.5 Gestion des requêtes et des données	170
7.1.6 Rendu conditionnel	171
7.2 Mise en cache et mémoïsation	171
7.3 Lazy loading et Suspense	171

7.4 Virtualisation	172
7.5 Code splitting	173
7.6 Analyse des performances	173
7.7 Test de performance	174

Chapitre 5

Gestion du state

1. Introduction	175
2. Les différentes solutions	175
2.1 L'état serveur (server state) et l'état client (client state)	176
2.2 Le Context API	176
2.3 Redux	176
2.4 MobX	177
2.5 Recoil	177
2.6 Zustand	177
3. L'association du contexte et des hooks	178
3.1 Crédation d'un contexte	178
3.2 Utilisation de Context avec le hook useContext	179
3.3 Utilisation de useReducer pour des états plus complexes	180
4. Redux	182
4.1 Les éléments-clés de Redux	182
4.1.1 Actions	182
4.1.2 Réducteur (reducer)	183
4.1.3 Store	183
4.1.4 Redux Toolkit	184
4.2 Crédation d'un store	184
4.3 Utiliser createSlice	185
4.4 Asynchronisme avec createAsyncThunk	186
4.5 Testabilité	187

5.	MobX	188
5.1	Principes de base de MobX	188
5.1.1	Observables	188
5.1.2	Actions	189
5.1.3	Réactions	189
5.1.4	Computed values	189
5.2	MobX avec React	190
5.3	Conclusion	191

Chapitre 6

Gestion du state serveur avec React Query

1.	Introduction	193
2.	Installation	193
2.1	Installer les dépendances	193
2.2	Mise en place du provider	194
3.	Requêtes (queries)	195
3.1	La fonction de fetch	195
3.2	Utilisation de useQuery	196
4.	Mutations	198
5.	Invalidation	200
5.1	Différents moyens d'invalider une requête	200
5.2	Autres moyens d'invalidation	201
5.3	Invalidations conditionnelles	201
6.	Utiliser les outils de développement dédiés	202

Chapitre 7**Développer pour le mobile avec React Native**

1.	Introduction	205
2.	Présentation de React Native.....	205
2.1	Composants natifs	205
2.2	Réutilisation de la logique	206
2.3	Accès aux fonctionnalités du périphérique.....	206
3.	Créer une application avec Expo	206
3.1	L'outil create-expo-app.....	207
3.2	L'application Expo Go	207
4.	Mettre en place la navigation	209
4.1	Installer les dépendances et configurer le projet.....	209
4.2	Le répertoire app.....	211
4.3	Naviguer entre les écrans	212
5.	Ajouter du style	213
5.1	Utilisation de styles en ligne	213
5.2	Utilisation de styles externes.....	214
5.3	Styles conditionnels	215
5.4	Utiliser Tailwind	216

Chapitre 8**Fonctionnalités avancées avec React Native**

1.	Introduction	219
2.	Utiliser la géolocalisation et accéder aux contacts	220
2.1	Utiliser la géolocalisation	220
2.1.1	Installer le module de géolocalisation	220
2.1.2	Utilisation de la géolocalisation.....	220
2.2	Accéder aux contacts	222
2.2.1	Installer le module de contacts	222
2.2.2	Utilisation de l'accès aux contacts.....	223

3. Exemple : utiliser un thème clair ou sombre avec l'API Context	225
4. Conclusion	228

Chapitre 9

Next.js, le framework React par Vercel

1. Introduction	229
1.1 Pourquoi Next.js ?	229
1.1.1 Rendu côté serveur (SSR) et génération de site statique (SSG)	229
1.1.2 Routage simple et intuitif	230
1.1.3 Préchargement des pages	230
1.1.4 Optimisation des images	230
1.1.5 Support CSS-in-JS et SCSS	230
2. Création d'un site avec Next.js	231
2.1 Crédit d'un projet Next.js	231
2.2 Crédit de pages	232
2.3 Lancement de l'application	233
3. Naviguer entre les pages.....	233
3.1 Utilisation du composant Link	233
3.2 Route dynamique	234
3.3 Naviguer avec useRouter	235
3.4 Navigation programmatique	235
4. Gestion des assets.....	236
4.1 Dossiers publics	236
4.2 Chargement des images	237
5. Utiliser SWR, l'équivalent de React Query.....	238
5.1 Installation de SWR	238
5.2 Utilisation de SWR.....	239
5.3 Options de configuration avancées	240

6.	Rendu côté serveur et génération de site statique	241
6.1	Rendu côté serveur (SSR)	241
6.2	Génération de site statique (SSG)	242
6.3	Rafraîchissement automatique (ISR)	243
7.	Présentation des React Server Components (RSC)	243

Chapitre 10

Introduction à Storybook

1.	Découverte et installation	247
1.1	Qu'est-ce que Storybook ?	247
1.2	Installation de Storybook	248
1.3	Création de stories	249
1.4	Lancement de Storybook	249
1.5	Ce qu'offre Storybook	250
2.	Création d'un composant	250
2.1	Création d'un composant	250
2.2	Visualisation et test du composant	252
2.3	L'utilité de la création de composants dans Storybook	252
3.	Utiliser des stories	253
3.1	Visualisation et test de la story	253
3.2	Avantages de la création de stories	253
4.	Utiliser des add-ons	254
4.1	Installation d'add-ons	254
4.2	Utilisation d'add-ons	255
4.3	Ce qu'offre l'utilisation d'add-ons	255
5.	Conclusion	256

Chapitre 11

Connecter React à une API GraphQL

1.	Présentation de GraphQL et premières requêtes	257
1.1	Qu'est-ce que GraphQL ?	257
1.2	Premières requêtes GraphQL avec React.....	258
2.	Création d'une API avec Hasura	262
3.	Appel d'une API avec React et Apollo Client	266
3.1	Configuration d'Apollo Client.....	266
3.2	Appel de l'API créée avec Hasura.....	266
4.	Conclusion	267

Chapitre 12

Maîtriser les design patterns React

1.	Introduction	269
2.	Higher-Order Components	270
2.1	Fonctionnement des Higher-Order Components.....	270
2.2	Ce qu'offrent les Higher-Order Components	271
2.3	Cas d'utilisation des Higher-Order Components	271
2.4	Limitations et considérations	272
2.4.1	Les HOC en cascade	272
2.4.2	Conflit de props	273
3.	Render props	274
3.1	Fonctionnement des render props.....	274
3.2	Qualités des render props.....	275
3.3	Cas d'utilisation des render props	275
3.3.1	Contrôle de composants interactifs	275
3.3.2	Gestion des états ou des données	276
3.4	Limitations et considérations	277

4.	Provider	278
4.1	Fonctionnement du Provider Pattern	278
4.2	Avantages du Provider Pattern	278
4.3	Cas d'utilisation du Provider Pattern.	279
4.4	Limitations et considérations	279
5.	Composants container et de présentation	280
5.1	Composants container.	280
5.2	Composants de présentation.	281
5.3	Ce qu'offrent les composants container et de présentation	281
5.4	Cas d'utilisation des composants container et de présentation.	282
5.5	Limitations et considérations	282
6.	Hooks	282
6.1	Les bases du pattern Hooks	283
6.2	Hooks personnalisés.	283
6.3	Extraction de logique	284

Chapitre 13

Tester une application React

1.	Utiliser Jest et React Testing Library	287
1.1	Introduction à Jest	287
1.2	Installation de Jest	288
1.3	Écriture de tests avec Jest et React Testing Library	288
1.4	Exécution des tests	289
1.5	Jest et React Testing Library	289
1.6	Limitations et considérations	289
2.	Écrire son premier test	290
2.1	Configuration de base	290
2.2	Écriture de votre premier test	290
2.3	Explication du test	291
2.4	Exécution du test	291
2.5	Pourquoi ce test simple ?	292

2.6 Limitations et considérations	292
3. Tester les composants avec Cypress	292
3.1 Configuration de base	293
3.2 Installation de Cypress.	293
3.3 Écriture de tests avec Cypress	293
3.4 Exécution des tests	294
3.5 Ce qu'offre Cypress	294
3.6 Limitations et considérations	294
3.7 Playwright, l'alternative à Cypress	295
4. Conclusion	296
4.1 L'importance des tests	296
4.2 Choix de la méthode de test	296
4.3 Bonnes pratiques de test	296
4.4 Intégration et déploiement continus.	297
4.5 Évolution des tests	297

Conclusion

1. Aller plus loin	299
1.1 Explorer des domaines connexes	299
1.2 Contribuer à la communauté.	300
1.3 Créer des projets personnels	300
1.4 Participer à des événements et des conférences	300
1.5 Expérimenter avec de nouvelles technologies.	300
2. Déployer avec Netlify.	301
2.1 Cr éation d'un projet et build	301
2.2 Cr éation d'un compte Netlify	301
2.3 Configuration de déploiement.	301
2.4 Déploiement	302

3.	Utiliser une bibliothèque de composants	303
3.1	Pourquoi utiliser une bibliothèque de composants ?	303
3.2	Exemples de bibliothèques de composants	303
3.3	Limitations	304
4.	Ressources pour aller plus loin	304
4.1	Documentation officielle	304
4.2	Blogs	305
4.3	Veille technologique	305
	Index	307