Effective TypeScript

62 Specific Ways to Improve Your TypeScript

Dan Vanderkam

Beijing · Boston · Farnham · Sebastopol · Tokyo O'REILLY®

Effective TypeScript
by Dan Vanderkam

Copyright © 2020 Dan Vanderkam. All rights reserved.

Printed in the United States of America.

Published by O’Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472.

O’Reilly books may be purchased for educational, business, or sales promotional use. Online editions are also available for most titles (*http://oreilly.com*). For more information, contact our corporate/institutional sales department: 800-998-9938 or *corporate@oreilly.com*.

Acquisitions Editor: Jennifer Pollock
Development Editor: Angela Rufino
Production Editor: Deborah Baker
Copyeditor: Jasmine Kwityn
Proofreader: Kim Wimpsett

Indexer: Judith McConville
Interior Designer: David Futato
Cover Designer: Karen Montgomery
Illustrator: Rebecca Demarest

November 2019: First Edition

Revision History for the First Edition
2019-10-16: First Release

See *http://oreilly.com/catalog/errata.csp?isbn=9781492053743* for release details.

The O’Reilly logo is a registered trademark of O’Reilly Media, Inc. *Effective TypeScript*, the cover image, and related trade dress are trademarks of O’Reilly Media, Inc.

The views expressed in this work are those of the author, and do not represent the publisher’s views. While the publisher and the author have used good faith efforts to ensure that the information and instructions contained in this work are accurate, the publisher and the author disclaim all responsibility for errors or omissions, including without limitation responsibility for damages resulting from the use of or reliance on this work. Use of the information and instructions contained in this work is at your own risk. If any code samples or other technology this work contains or describes is subject to open source licenses or the intellectual property rights of others, it is your responsibility to ensure that your use thereof complies with such licenses and/or rights.

978-1-492-05374-3

[LSI]

For Alex.
You’re just my type.

This page intentionally left blank

Table of Contents

Preface. xi

1. Getting to Know TypeScript. 1
Item 1: Understand the Relationship Between TypeScript and JavaScript 1
Item 2: Know Which TypeScript Options You're Using 7
Item 3: Understand That Code Generation Is Independent of Types 10
Item 4: Get Comfortable with Structural Typing 16
Item 5: Limit Use of the any Type 20

2. TypeScript's Type System. 25
Item 6: Use Your Editor to Interrogate and Explore the Type System 25
Item 7: Think of Types as Sets of Values 29
Item 8: Know How to Tell Whether a Symbol Is in the Type Space or Value Space 35
Item 9: Prefer Type Declarations to Type Assertions 40
Item 10: Avoid Object Wrapper Types (String, Number, Boolean, Symbol, BigInt) 43
Item 11: Recognize the Limits of Excess Property Checking 46
Item 12: Apply Types to Entire Function Expressions When Possible 49
Item 13: Know the Differences Between type and interface 52
Item 14: Use Type Operations and Generics to Avoid Repeating Yourself 56
Item 15: Use Index Signatures for Dynamic Data 64
Item 16: Prefer Arrays, Tuples, and ArrayLike to number Index Signatures 67
Item 17: Use readonly to Avoid Errors Associated with Mutation 71
Item 18: Use Mapped Types to Keep Values in Sync 77

3. Type Inference. 81
Item 19: Avoid Cluttering Your Code with Inferable Types 81

Item 20: Use Different Variables for Different Types 87
Item 21: Understand Type Widening 90
Item 22: Understand Type Narrowing 93
Item 23: Create Objects All at Once 96
Item 24: Be Consistent in Your Use of Aliases 99
Item 25: Use async Functions Instead of Callbacks for Asynchronous Code 102
Item 26: Understand How Context Is Used in Type Inference 107
Item 27: Use Functional Constructs and Libraries to Help Types Flow 111

4. Type Design. 117
Item 28: Prefer Types That Always Represent Valid States 117
Item 29: Be Liberal in What You Accept and Strict in What You Produce 122
Item 30: Don’t Repeat Type Information in Documentation 125
Item 31: Push Null Values to the Perimeter of Your Types 127
Item 32: Prefer Unions of Interfaces to Interfaces of Unions 131
Item 33: Prefer More Precise Alternatives to String Types 134
Item 34: Prefer Incomplete Types to Inaccurate Types 138
Item 35: Generate Types from APIs and Specs, Not Data 142
Item 36: Name Types Using the Language of Your Problem Domain 147
Item 37: Consider “Brands” for Nominal Typing 149

5. Working with any. 153
Item 38: Use the Narrowest Possible Scope for any Types 153
Item 39: Prefer More Precise Variants of any to Plain any 155
Item 40: Hide Unsafe Type Assertions in Well-Typed Functions 157
Item 41: Understand Evolving any 159
Item 42: Use unknown Instead of any for Values with an Unknown Type 162
Item 43: Prefer Type-Safe Approaches to Monkey Patching 166
Item 44: Track Your Type Coverage to Prevent Regressions in Type Safety 168

6. Types Declarations and @types. 171
Item 45: Put TypeScript and @types in devDependencies 171
Item 46: Understand the Three Versions Involved in Type Declarations 173
Item 47: Export All Types That Appear in Public APIs 177
Item 48: Use TSDoc for API Comments 178
Item 49: Provide a Type for this in Callbacks 181
Item 50: Prefer Conditional Types to Overloaded Declarations 185
Item 51: Mirror Types to Sever Dependencies 187
Item 52: Be Aware of the Pitfalls of Testing Types 189

7. Writing and Running Your Code. 195
Item 53: Prefer ECMAScript Features to TypeScript Features 195

Item 54: Know How to Iterate Over Objects 200
Item 55: Understand the DOM hierarchy 202
Item 56: Don’t Rely on Private to Hide Information 207
Item 57: Use Source Maps to Debug TypeScript 210

8. Migrating to TypeScript. 215
Item 58: Write Modern JavaScript 216
Item 59: Use @ts-check and JSDoc to Experiment with TypeScript 223
Item 60: Use allowJs to Mix TypeScript and JavaScript 228
Item 61: Convert Module by Module Up Your Dependency Graph 229
Item 62: Don’t Consider Migration Complete Until You Enable noImplicitAny 234

Index. 237