

# TYPESCRIPT LIBRARY COOKBOOK

Jakub Chodorowicz  
[github.com/chodorowicz](https://github.com/chodorowicz)  
twitter @chodorowicz

# WHERE DID I USE IT

- <https://github.com/chodorowicz/ts-debounce>

```
import { debounce } from 'ts-debounce';

const debouncedFunction = debounce(originalFunction, waitMilliseconds, options);
```

# FIRST THINGS FIRST

```
{  
  "name": "your-package-name",  
  "version": "0.1.0",  
  "description": "🚀狎&emsp;🍄",  
  "keywords": [],  
  "homepage": "https://github.com/chodorowicz/ts-debounce",  
  "bugs": {  
    "url" : "https://github.com/chodorowicz/ts-debounce/issues"  
  }  
  "files": ["array", "of", "file", "patterns"], // defaults to ["*"]  
  "license": "MIT",  
  "main": "dist/src/index.js",  
  "module": "dist/src/index.m.js",  
  "browser": "dist/src/index.umd.js",  
  "types": "// can be left out if index.d.ts in root",  
  "author": "Jakub Chodorowicz <chodorowicz@gmail.com> (https://jakub.chodorowicz.pl/)"  
}
```

# WHAT DO YOU NEED

microbundle (with TypeScript) + NP

```
npm i --D microbundle np
```

```
mkdir src && cd src && touch index.ts
```

# CONFIGURE TYPESCRIPT

- not much here to do 
- microbundle/rollup-plugin-typescript2 enforces most of options

```
{  
  "compilerOptions": {  
    "strict": true,  
    "noUnusedLocals": true,  
    "noUnusedParameters": true,  
    "esModuleInterop": true, // if needed  
    "declarationMap": true  
    // other important declaration can be left out when using microbundle  
  }  
}
```

# MICROBUNDLE

- one dependency to bundle your library using only a package.json
- supports multiple entry points
- multiple outputs for each entry - CJS, UMD & ESM
- tiny, optimized
  - terser compression
  - gzipped
  - cssnano

# ROLLUP AND MICROBUNDLE ENFORCED CONFIG

```
noEmitHelpers: false
importHelpers: true // required tslib, provided by rollup-plugin-typescript2
noResolve: false
noEmit: false
inlineSourceMap: false
outDir: taken from package.json `main`
target: 'esnext' // because it's transpiled by babel and Bublé
declaration: true // generate d.ts files
sourceMap: true
```

<https://github.com/developit/microbundle/blob/master/src/index.js#L454>

```
useTypescript &&
  typescript({
    typescript: require('typescript'),
    cacheRoot: `./.rts2_cache_${format}`,
    tsconfigDefaults: {
      compilerOptions: {
        sourceMap: options.sourcemap,
        declaration: true,
        jsx: options.jsx,
      },
    },
    tsconfigOverride: {
      compilerOptions: {
        target: 'esnext',
      },
    },
  }),
}
```

# TARGET 🎯 - WHAT IS DOWNLEVELING

- ES3/ES5 downleveling

The screenshot shows a browser-based TypeScript transpiler interface. On the left, the TypeScript code is displayed:

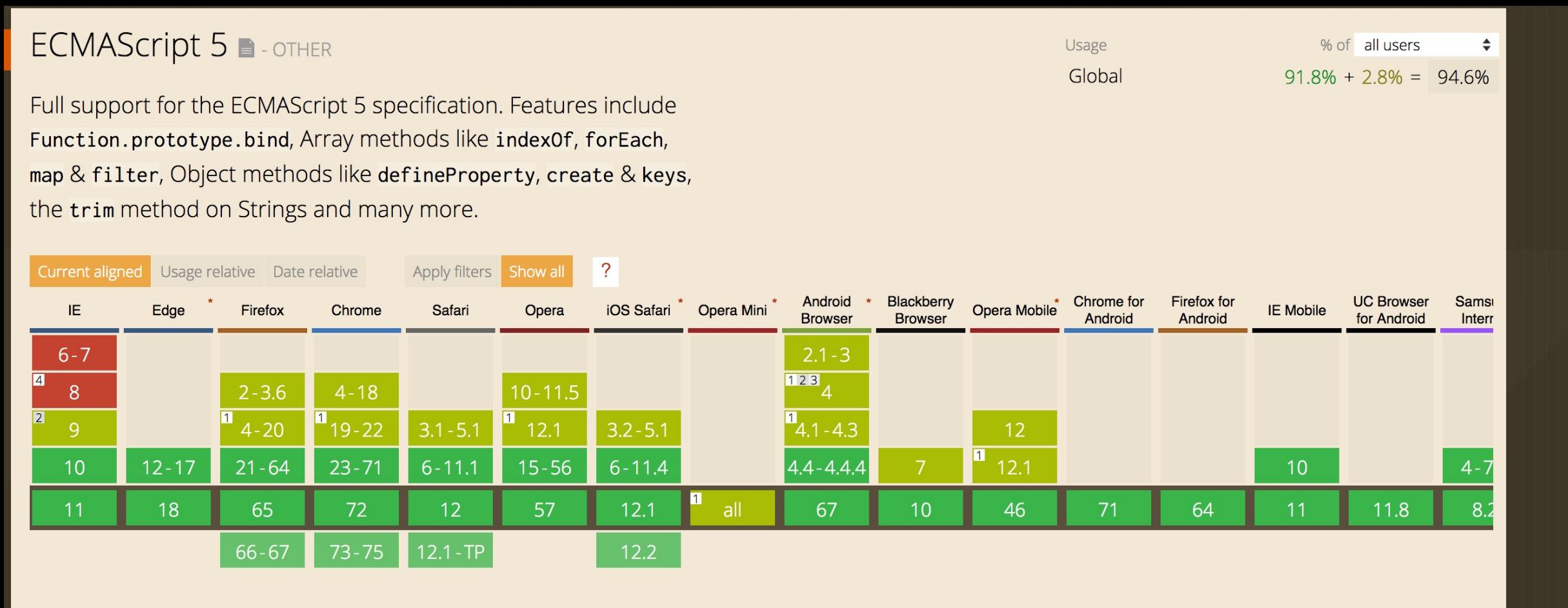
```
1 function delay(ms: number) {
2   return new Promise<void>(function(resolve) {
3     setTimeout(resolve, ms);
4   });
5 }
6
7
8 async function asyncAwait() {
9   await delay(1000);
10}
11
12
```

On the right, the generated JavaScript code is shown, which implements the `delay` and `asyncAwait` functions using Promises and a generator-based loop:

```
1 var __awaiter = (this && this.__awaiter) || function (thisArg, _arguments, P, generator)
2   return new (P || (P = Promise))(function (resolve, reject) {
3     function fulfilled(value) { try { step(generator.next(value)); } catch (e) { reject(e); } }
4     function rejected(value) { try { step(generator["throw"](value)); } catch (e) { reject(e); } }
5     function step(result) { result.done ? resolve(result.value) : new P(function (resolve) {
6       step((generator = generator.apply(thisArg, _arguments || [])).next());
7     });
8   };
9   var __generator = (this && this.__generator) || function (thisArg, body) {
10    var _ = { label: 0, sent: function() { if (t[0] & 1) throw t[1]; return t[1]; }, try: function() { if (t[0] & 2) throw t[1]; return t[1]; }, error: function() { if (t[0] & 4) throw t[1]; return t[1]; }, done: function(v) { if (t[0] & 8) return step([v]); else return step([v]); } };
11    return g = { next: verb(0), "throw": verb(1), "return": verb(2) }, typeof Symbol === "function" ? (g[Symbol.iterator] = function() { return this; }), __generator : g;
12    function verb(n) { return function (v) { return step([n, v]); }; }
13    function step(op) {
14      if (f) throw new TypeError("Generator is already executing.");
15      while (_) try {
16        if (f = 1, y && (t = op[0] & 2 ? y["return"] : op[0] ? y["throw"] || ((t = y).value = op[1], y = void 0, t.value) : op[1]), t)
17          if (y = 0, t) op = [op[0] & 2, t.value];
18          switch (op[0]) {
19            case 0: case 1: t = op; break;
20            case 4: _.label++; return { value: op[1], done: false };
21            case 5: _.label++; y = op[1]; op = [0]; continue;
22            case 7: op = _.ops.pop(); _.trys.pop(); continue;
23            default:
24              if (!(t = _.trys, t = t.length > 0 && t[t.length - 1]) && (op[0] === 6 || op[0] === 0))
25                if (op[0] === 3 && (!t || (op[1] > t[0] && op[1] < t[3]))) { _.label = op[1]; continue; }
26                if (op[0] === 6 && _.label < t[1]) { _.label = t[1]; t = op; break; }
27                if (t && _.label < t[2]) { _.label = t[2]; _.ops.push(op); break; }
28                if (t[2]) _.ops.pop();
29                _.trys.pop(); continue;
30          }
31          op = body.call(thisArg, _);
32        } catch (e) { op = [6, e]; y = 0; } finally { f = t = 0; }
33        if (op[0] & 5) throw op[1]; return { value: op[0] ? op[1] : void 0, done: true };
34    }
35  };
36  function delay(ms) {
37    return new Promise(function (resolve) {
38      setTimeout(resolve, ms);
39    });
40 }
```

# WHAT SHOULD BE THE TARGET

- ES5 should be good from most cases
  - <https://kangax.github.io/compat-table/es5/>
  - <https://caniuse.com/#search=es5>



## CONTROLL WHAT YOU'RE PUBLISHING

- using `.gitignore` and `.npmignore` (blacklisting) considered dangerous
- better use `files` in package.json
- use `npm-packlist` to verify your files programtically
- use `npm pack`

```
~/projects/talk-ts-library master*
> tar tvf $(npm pack)
npm notice
npm notice 📦 talk-ts-library@1.0.0
npm notice === Tarball Contents ===
npm notice 579B package.json
npm notice 9B README.md
npm notice 271B dist/index.d.ts
npm notice 205B dist/index.d.ts.map
npm notice 395B dist/index.js
npm notice 706B dist/index.js.map
npm notice 408B dist/index.mjs
npm notice 772B dist/index.mjs.map
npm notice 562B dist/index.umd.js
npm notice 711B dist/index.umd.js.map
npm notice === Tarball Details ===
npm notice name:          talk-ts-library
npm notice version:       1.0.0
npm notice filename:       talk-ts-library-1.0.0.tgz
npm notice package size:  1.6 kB
npm notice unpacked size: 4.6 kB
npm notice shasum:        a536c2d73024686099813fa77fa17a946e43fc21
npm notice integrity:     sha512-ebhZnmyBrspbL[...]2GoD4R41vutag==
npm notice total files:   10
npm notice
-rw-r--r-- 0 0      0      579 26 Oct 1985 package/package.json
-rw-r--r-- 0 0      0      9 26 Oct 1985 package/README.md
-rw-r--r-- 0 0      0     271 26 Oct 1985 package/dist/index.d.ts
-rw-r--r-- 0 0      0     205 26 Oct 1985 package/dist/index.d.ts.map
-rw-r--r-- 0 0      0     395 26 Oct 1985 package/dist/index.js
-rw-r--r-- 0 0      0     706 26 Oct 1985 package/dist/index.js.map
```

**EVERYBODY MAKES MISTAKES** 😔

- <https://unpkg.com/ts-debounce@1.0.0/>

# PUBLISHING STEPS

- prerequisite: npm account
- Ensure you are publishing from the master branch
- Ensure the working directory is clean and that there are no unpulled changes
- Reinstall dependencies to ensure your project works with the latest dependency tree
- Run the tests
- Bump the version in package.json and npm-shrinkwrap.json (if present) and creates a git tag
- Publish the new version to npm
- Push commits and tags to GitHub/GitLab
- release notes on GitGub

# NP

- better `npm publish`

```
✓ Cleanup
✓ Installing dependencies
∴ Running tests
Bumping version
Publishing package
Pushing tags
```

# INTERACTIVE NP



np

```
~/dev/cat-pad master
> np
```

Publish a new version of **cat-pad** (2.0.0)

? **Select semver increment or specify new version**

- patch (2.0.1)
- > minor (2.1.0)
- major (3.0.0)
- prepatch (2.0.1-0)
- preminor (2.1.0-0)
- premajor (3.0.0-0)
- prerelease (2.0.1-0)

---

Other (specify)

# ADD IT TO DEPENDENCIES AND SCRIPTS

```
{  
  "scripts": {  
    "build": "microbundle",  
    "dev": "microbundle watch",  
    "release": "np"  
  }  
}
```

# BACKWARD TYPES COMPABILITY

- use **typesVersions** (available since TS 3.1)

```
{  
  "name": "package-name",  
  "version": "1.0",  
  "types": "./index.d.ts",  
  "typesVersions": {  
    ">=3.2": { "*": ["ts3.2/*"] },  
    ">=3.1": { "*": ["ts3.1/*"] }  
  }  
}
```

# BACKWARD TYPES COMPABILITY

## TypeScript compatibility notes

---

- v1 - minimum TS v2.7.2
- v2 - minimum TS v2.8.1 (rewritten to conditional types)
- v3 - minimum TS v3.1.0

## BACKWARD TYPES COMPABILITY - OTHER IDEAS

🤔.cloud peerDependencies...

# TESTING IN TYPESCRIPT OF COURSE

- `npm install --save-dev jest ts-jest @types/jest`

# TYPESCRIPT JEST CONFIG

```
{  
  "scripts": {  
    "test: "jest"  
  },  
  "jest": {  
    "transform": {  
      "^.+\\.tsx?$$": "<rootDir>/node_modules/ts-jest/preprocessor.js"  
    },  
    "testRegex": "(src/__tests__/.*/|(.|/)(test|spec))\\.(ts|js)$",  
    "moduleFileExtensions": [  
      "js",  
      "ts",  
      "tsx"  
    ]  
  }  
}
```

## BONUS EXTRA ROUND - PIKA PACK

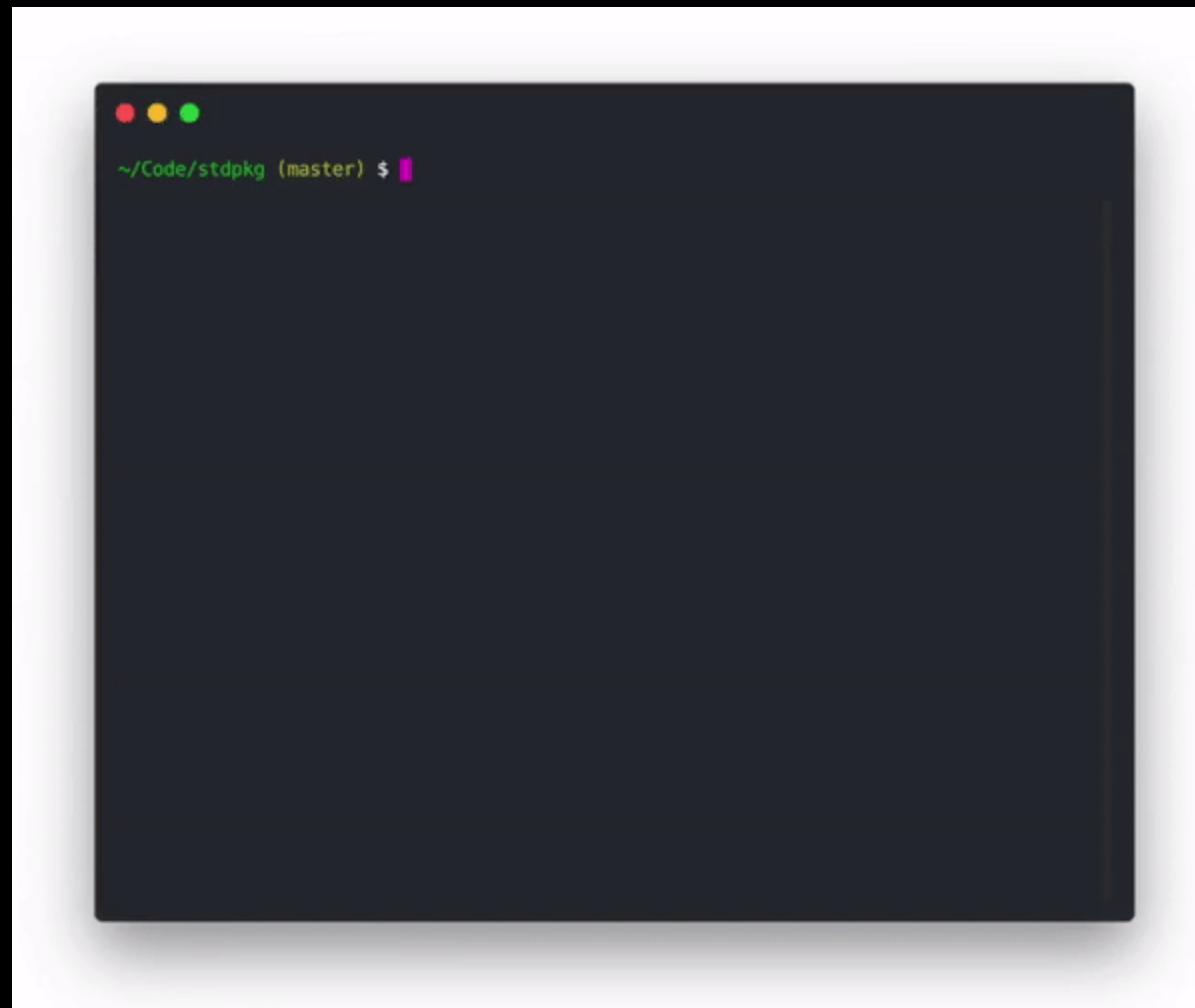
- extensible microbundle + np on one package 

# CONFIG

```
// Before: Your top-level package.json manifest:  
{  
  "name": "simple-package",  
  "version": "1.0.0",  
  "@pika/pack": {  
    "pipeline": [  
      ["@pika/plugin-standard-pkg", {"exclude": ["__tests__/*"]}],  
      ["@pika/plugin-build-node"],  
      ["@pika/plugin-build-web"]  
    ]  
  }  
}
```

**PIKA DEMO**

# PUBLISH WITH PIKA



## WHAT ELSE

- CI (e.g. Travis CI)
- test coverage (e.g.Codecov)
- keeping dependencies up to date (e.g. Greenkeeper)

THE END

**GOOD DAY TO YOU**

**HAVE YOU HEARD ABOUT  
TYPESCRIPT**

[memegenerator.net](http://memegenerator.net)