

Path to strict type checking

Jakub Chodorowicz
github.com/chodorowicz
jakub@chodorowicz.com
@chodorowicz

Make it `--strict`

- `--strictNullChecks`
- `--noImplicitAny`
- `--noImplicitReturns`
- `--noImplicitThis`
- `--alwaysStrict`
- `--strictBindCallApply`
- `--strictFunctionTypes`
- `--strictPropertyInitialization`

TS foo.ts

×

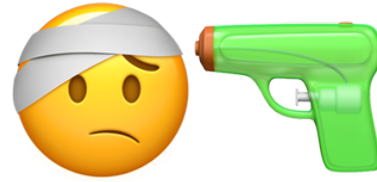
```
1  interface User {
2      name: string;
3      age?: number;
4  }
5  function printUserInfo(user: User) {
6      console.log(`${user.name}, ${user.age.toString()}`);
7      // => error TS2532: Object is possibly 'undefined'.
8      console.log(`${user.name}, ${user.age!.toString()}`);
9      // => OK, you confirm that you're sure user.age is non-null.
10
11     if (user.age !== null) {
12         console.log(`${user.name}, ${user.age.toString()}`);
13     }
14     // => OK, the if-condition checked that user.age is non-null.
15
16     console.log(
17         user.name + ", " + user.age !== null ? user.age.toString() : "age unknown"
18     );
19     // => Unfortunately TypeScript can't infer that age is non-null here.
20 }
```

**Projects often end up
with no strict TS compiler
settings**

Problem in VS Code

- Thousands of errors when turning on `strictNullChecks`
- See more: <https://github.com/Microsoft/vscode/issues/60565>

Problem in our codebase



```
src/main/resources/dgc/Workflows/WorkflowButtonItem.tsx:43:11 - error
Type 'undefined' is not assignable to type 'string'.
```

```
43         id={this.props.workflow.id}
```

~~

```
node_modules/@collibra/uf/components/dom/buttonGroup/ButtonGroupItem.d
```

```
4         id: string;
```

~~

The expected type comes from property 'id' which is declared here on

Found 14624 errors.

Create a second TS config extending original

```
{  
  "extends": "./tsconfig.json",  
  "compilerOptions": {  
    "strict": true,  
    "noUnusedLocals": true,  
    "noUnusedParameters": true  
  },  
}
```

Start adding files one by one

```
{
  "extends": "./tsconfig.json",
  "compilerOptions": {
    "strict": true,
    "noUnusedLocals": true,
    "noUnusedParameters": true
  },
  "files": [
    "src/main/resources/dgc/Types.ts",
  ]
}
```


Hook it up to some automation

- Use Husky and add it as precommit/prepush script
- Add this as additional step to CI (Jenkins/CircleCI/Travis)

```
... "types:check": "tsc -p .",  
... "types:check-strict": "tsc -p ./tsconfig.strict.json",  
"lint": "eslint --fix --ext .ts ./src" }
```

Problem: strict settings are contagious



Some help


- <https://github.com/mjbvz/vscode-strict-null-check-migration-tools>
- `index.js`
the main script prints of list of files that are eligible for strict null checks. This includes all files that only import files there are already strict null checked.
- `autoAdd.js`
Very simple script that tries to auto add any eligible file to the `tsconfig.strictNullChecks.json`. This iteratively compiles the `tsconfig` project with just that file added. If there are no errors, it is added to the `tsconfig`

**More problems and
issues 🤒**

Relay Classic GraphQL API types

```
export interface IRating {  
  __typename?: string;  
  id?: string;  
  system?: boolean;  
  createdAt?: any;  
  createdBy?: IUser;  
  lastModifiedOn?: any;  
  lastModifiedBy?: IUser;  
  rating?: number;  
  review?: string;  
  asset?: IAsset;  
  comments?: ICommentConnection;  
}
```

Barrel files

src ▸ main ▸ resources ▸ dgc ▸ Activities ▸  index.tsx

```
1  import { JobsWatcher } from "../JobsWatcher";
2  import { JobProgressBar } from "../JobProgressBar";
3  import { Statuses } from "../constants/Statuses";
4  import { registerCompletedResult, registerFailedResult }
5  import { ActivitiesTableResultsContext } from "../Activit
6
7  export {
8    · JobsWatcher,
9    · JobProgressBar,
10   · Statuses,
11   · ActivitiesTableResultsContext,
12   · registerCompletedResult,
13   · registerFailedResult
14  };
15
```

The end

Jakub Chodorowicz
github.com/chodorowicz
jakub@chodorowicz.com
@chodorowicz

Charity Hackathon - Kodowanie



Saturday, June 29, 2019

<http://bit.ly/kodawanie>
http://bit.ly/kodawanie_fb