Why use TypeScript, good and bad reasons

Charly POLY

Paris TypeScript #13

Why use TypeScript, good and bad reasons

> *“ Should I use TypeScript? Do I need it?*

⚠ *Theses slides only reflects my opinion, lessons I learned and quoted opinions of others developers.*

Bad reasons

Good reasons

Conclusion

Why use TypeScript, good and bad reasons

With TypeScript,
no need of writing tests

Types are written by the developers and not checked at runtime, so the “strength” of your program depends on your usage of TypeScript.

With TypeScript,
no need of writing tests

> *“ Is a 15% reduction in bugs making it all the way through your development pipeline worth it to you?*

With TypeScript,
I can trust my code at runtime

TypeScript is a static type-checking tool, it does not guarantee that your code will run as expected at runtime.

Why use TypeScript, good and bad reasons

TypeScript looks more like C#,
I don’t need to understand JavaScript

TypeScript shouldn't be used to put some non-JavaScript developers on the front-end as fast as possible.

Why use TypeScript, good and bad reasons

Bad reasons

Good reasons

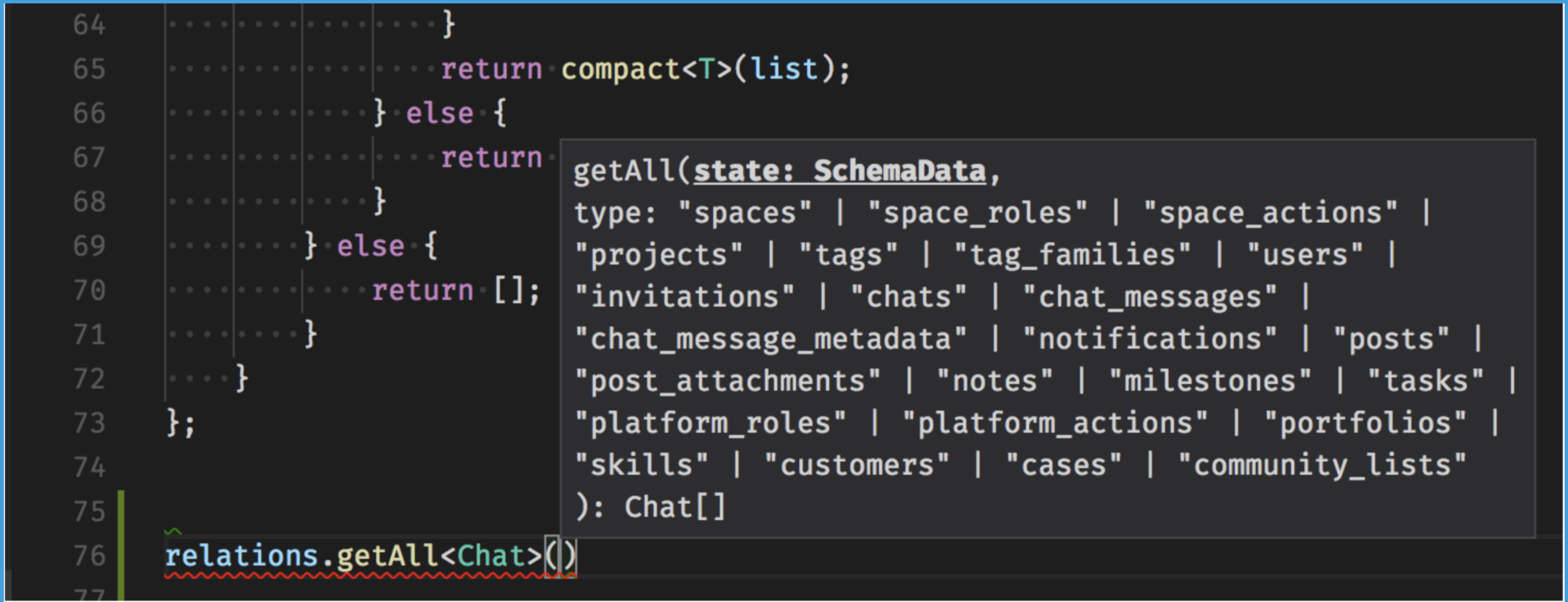
Conclusion

Why use TypeScript, good and bad reasons

TypeScript will help your developers work together on a growing codebase

TypeScript allow developers to focus on exposed API rather than having to know all the code by heart.

TypeScript will help your developers work together on a growing codebase



```
                }
                return compact<T>(list);
            } else {
                return
            }
        } else {
            return [];
        }
    }
};

relations.getAll<Chat>()
```

```
getAll(state: SchemaData,
type: "spaces" | "space_roles" | "space_actions" |
"projects" | "tags" | "tag_families" | "users" |
"invitations" | "chats" | "chat_messages" |
"chat_message_metadata" | "notifications" | "posts" |
"post_attachments" | "notes" | "milestones" | "tasks" |
"platform_roles" | "platform_actions" | "portfolios" |
"skills" | "customers" | "cases" | "community_lists"
): Chat[]
```

TypeScript tools will save your developers time

> *“ This allow me to focus on real problems and let TypeScript cover my back on the things I already know but that I forget sometimes (typos, standard APIs, etc ...)*

Why use TypeScript, good and bad reasons

The Microsoft community-effort success story

- JS typing tool battle winner
- OSS libraries typing coverage
- Libraries integrations : React, ApolloData

Why use TypeScript, good and bad reasons

TypeScript gives you a taste of future JavaScript, with typings.

Even if “TypeScript” contains “type”,
the language do not force us to use types.

Bad reasons

Good reasons

Conclusion

Why use TypeScript, good and bad reasons

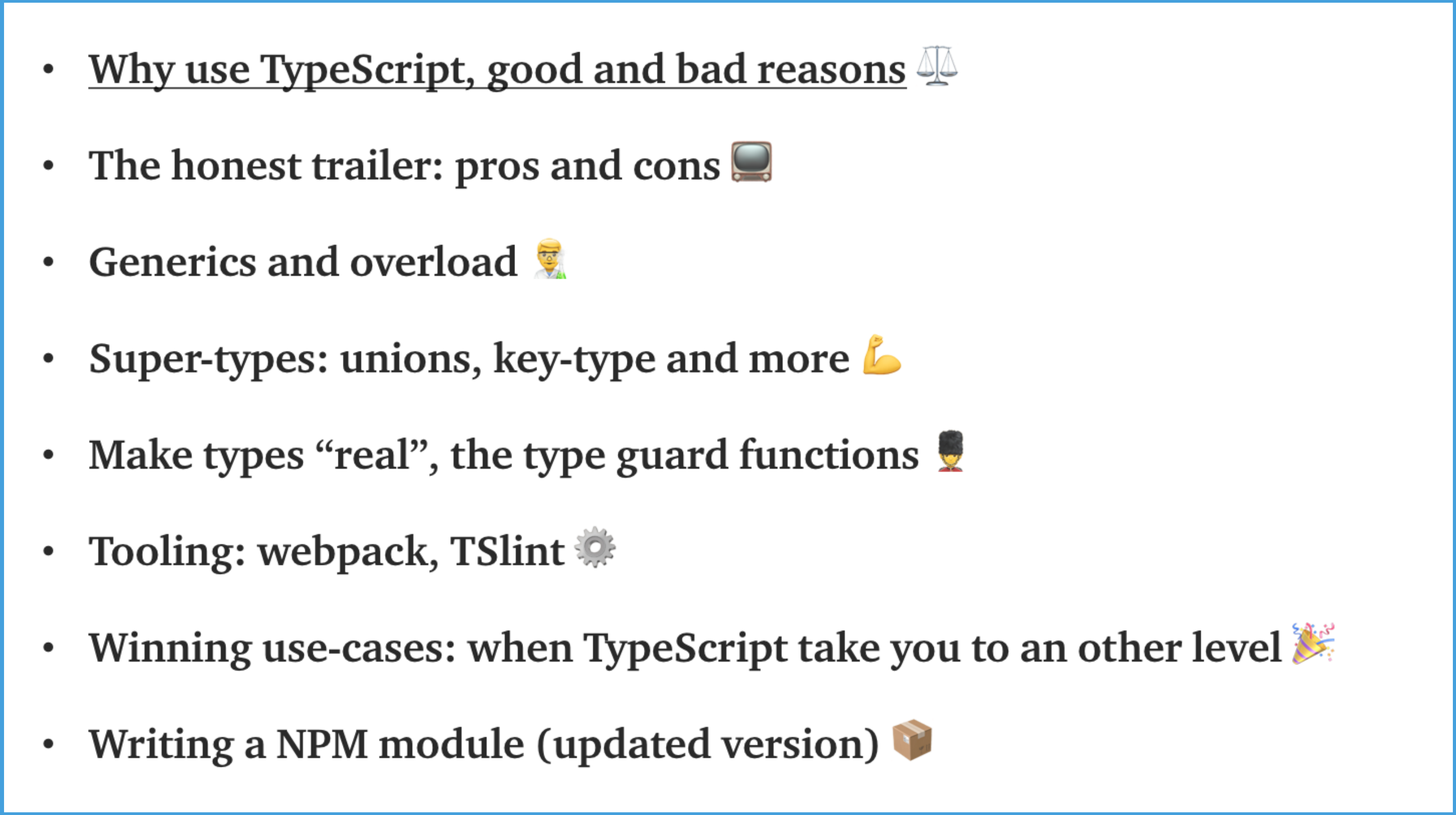
Bad

- Test replacement,
  TypeScript ain't a test framework
- Runtime safety,
  TypeScript ain't a runtime type checker
- I can write object-oriented C# a-like code,
  TypeScript is typed JavaScript

Good

- Team and code scalability with "Interface oriented development"
  TypeScript will help you dealing with growing teams and large codebase
- Tooling and Community
  Microsoft made an awesome community effort work.
- ES-next compliance
  TypeScript give you a taste of future JavaScript, with typing.

TypeScript Essentials



- Why use TypeScript, good and bad reasons ⚖️
- The honest trailer: pros and cons 📺
- Generics and overload 👨‍🔬
- Super-types: unions, key-type and more 💪
- Make types “real”, the type guard functions 💂
- Tooling: webpack, TSlint ⚙️
- Winning use-cases: when TypeScript take you to an other level 🎉
- Writing a NPM module (updated version) 📦

Sources

- https://medium.com/javascript-scene/the-shocking-secret-about-static-types-514d39bf30a3
- https://blog.acolyer.org/2017/09/19/to-type-or-not-to-type-quantifying-detectable-bugs-in-javascript/
- https://github.com/Microsoft/TypeScript/wiki/Standalone-Server-%28tsserver%29
- https://microsoft.github.io/language-server-protocol/