TypeScript migration

Szymon Kłos

Software engineer

szymon.klos@collabora.com

Collabora Online

LibreOffice Technology

Technical Day

COOL days

Technical Day
COOL days

Upgrade in the toolkit

- Newer ECMAScript standard: **ES5 → ES2017**
- Updated **eslint** and it's rules (linter, syntax checker)
- Introduced coding style check – **prettier**

```
make prettier-write
```

Collabora Online

Technical Day COOL days

Newer ECMAScript standard

- **Arrow functions** (“lambda”):

```
(event) => { doSomething(event); }
```

- **Classes** (as a replacement for Leaflet’s OOP concept)

```
class Rectangle          rect = new Rectangle();
```

- Block scope variables: **let** and **const**

```
{
    if (counter > 0) doSomething(); // illegal!!
    let counter = 3;

...
```

Collabora Online

Technical Day COOL days

Newer ECMAScript standard

- **Strings templating**:

```
const label = `Name: ${name}`;
```

- **Async** functions and **await**

Collabora Online

Technical Day
COOL days

TypeScript

- New code should be written in TypeScript.
- It can help to prevent mistakes like unwanted type conversions, etc…
- Fail fast – on transpilation, not runtime
- With defined types it will be easier to work on the unknown code

Collabora Online

Technical Day COOL days

How to convert JS → TS ?

- Change file's extension ***.js → *.ts**
- Adjust Makefile.am entry to use new file
- Add missing types:

```
- generateName(seed) {}
+ generateName(seed: number): string {}
```

- If possible replace old Leaflet "classes" with native classes
- Change **`var`** into **`let`** or **`const`**

Collabora Online

Collabora Online

LibreOffice Technology

Technical Day COOL days

Thank you!

By Szymon Kłos

@CollaboraOffice
hello@collaboraoffice.com
www.collaboraoffice.com