flowup

Typed translations

Vojtech Mašek
Head dev @ FlowUp

/ vmasek
/ VojtechMasek
/ @vmasek

Common solutions

ngx-translate	Angular i18n
3rd party solution	Integrated solution
Widely used	Not so widely used
Can be lazy loaded	App is build for each locale
Usage is (im-pure) pipe and translation “keys”	Usage is template attribute and translation id

What can be improved

- ➔ Simple way to check if **all translation** files are **correct**
- ➔ A way to check that **keys** used to access translation are **valid**

TypeScript knows all we need and dependency injection will provide the way, with just a little bit of router’s elegant lazy-loading

flow^up

Translation object and its type

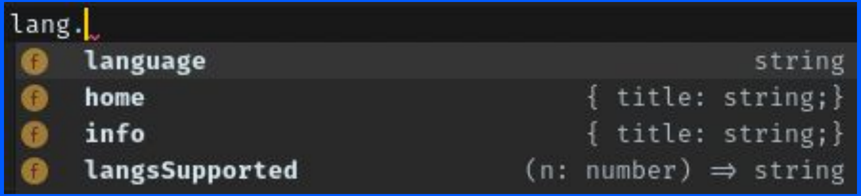
```typescript
export const englishTranslation = {
  language: 'English',
  home: { // object containing multiple translations
    title: 'Home works!', // plain string translation
    desc: 'This is description',
  },

  // simple function using string literal and interpolation
  // see the cs.translation.ts for another example
  langsSupported: (n: number) =>
    `This demo supports ${n} language${n === 1 ? '' : 's'}.`,
};

export type Translation = typeof englishTranslation;
```

Translated component

```
@Component(
  selector: 'app-example',
  template: `<h2>{{lang.home.title}}</h2>`,
)
export class ExampleComponent {
  constructor(@Inject(TRANSLATION) public lang: Translation) {
    console.log('current language is', lang.language);
  }
}
```

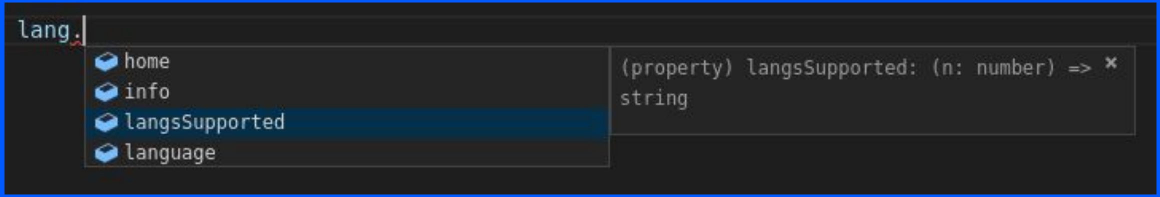
flowup

Auto-completion IDE support



```
lang.
  language                              string
  home                      { title: string;}
  info                      { title: string;}
  langsSupported         (n: number) ⇒ string
```

Intellisense for translations (WebStorm)

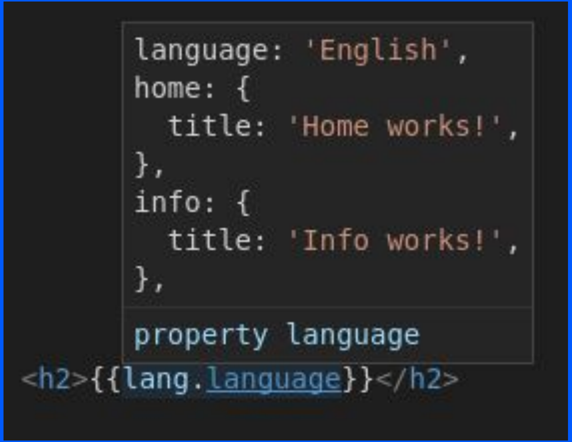


```
lang.
  home               (property) langsSupported: (n: number) =>
  info               string
  langsSupported
  language
```

Auto-completion of the available translation values (VS Code)

flow^up

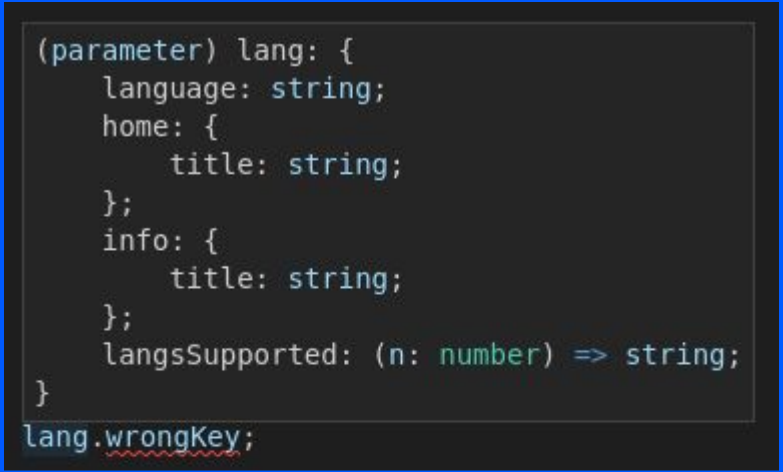
Type IDE support



```
language: 'English',
home: {
  title: 'Home works!',
},
info: {
  title: 'Info works!',
},
property language
<h2>{{lang.language}}</h2>
```

Type is supported even in a template (WebStorm)

Help with inherited type interface and error for an invalid key (VS Code)



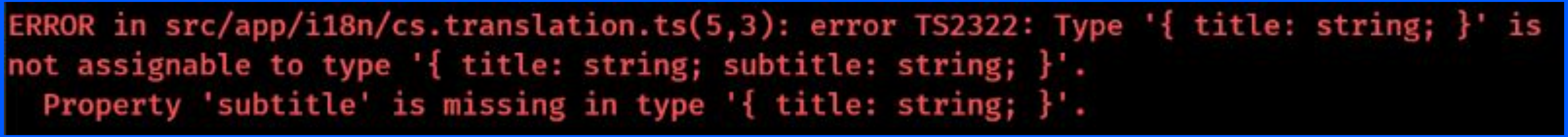
```
(parameter) lang: {
    language: string;
    home: {
        title: string;
    };
    info: {
        title: string;
    };
    langsSupported: (n: number) => string;
}
lang.wrongKey;
```

No more missing translations

Errors are caught at compile time

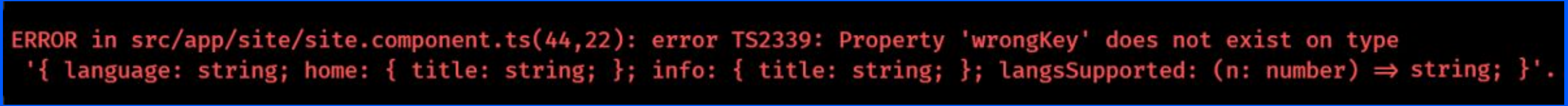
flow^up

Some of the caught errors



```
ERROR in src/app/i18n/cs.translation.ts(5,3): error TS2322: Type '{ title: string; }' is
not assignable to type '{ title: string; subtitle: string; }'.
  Property 'subtitle' is missing in type '{ title: string; }'.
```

Error after adding subtitle to only one of the translations



```
ERROR in src/app/site/site.component.ts(44,22): error TS2339: Property 'wrongKey' does not exist on type
 '{ language: string; home: { title: string; }; info: { title: string; }; langsSupported: (n: number) ⇒ string; }'.
```

Error when trying to access non-existing translation with invalid key

Pros

- ➔ No missing (untranslated) keys and parts of the application
- ➔ A quick way to validate multiple translate files (objects) compatibility
- ➔ Type-safe translation
  - ◆ the compiler will tell if something is not in order
- ➔ Translation data lazy-loaded with route module
- ➔ Instant
  - ◆ nothing needs to be dynamically loaded
- ➔ No need for in-template impure pipes
- ➔ Translation type can be inherited from default translation
- ➔ Routes are prefixed with language/i18n abbreviation
  - ◆ They can be localized entirely
- ➔ Translations can be strings, functions, string literal or any custom format

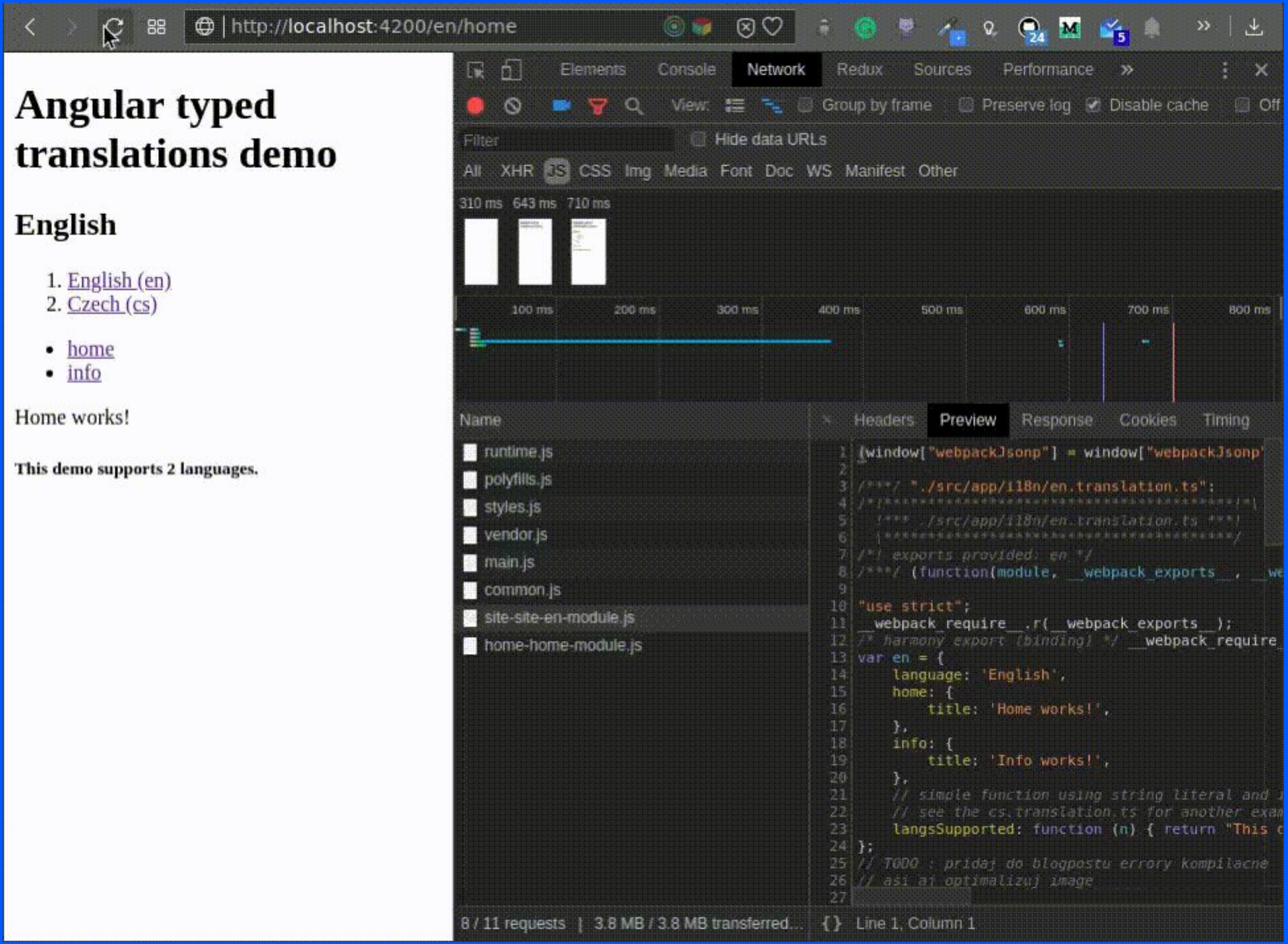
Cons

- ➔ A requirement of lazy-loading route modules

  (if we want the i18n modules and translations lazy-loaded)
- ➔ Loading translation dynamically from another source (server) is a bit tricky and require extra steps

Demo time!

github.com/vmasek/angular-typed-translations-demo



Angular typed translations demo

English

1. English (en)
2. Czech (cs)

- home
- info

Home works!

This demo supports 2 languages.

runtime.js
polyfills.js
styles.js
vendor.js
main.js
common.js
site-site-en-module.js
home-home-module.js

```
(window["webpackJsonp"] = window["webpackJsonp"

/***/ "./src/app/i18n/en.translation.ts":
/*!****************************************!*\
  !*** ./src/app/i18n/en.translation.ts ***!
  \****************************************/
/*! exports provided: en */
/***/ (function(module, __webpack_exports__, __we

"use strict";
__webpack_require__.r(__webpack_exports__);
/* harmony export (binding) */ __webpack_require_
var en = {
    language: 'English',
    home: {
        title: 'Home works!',
    },
    info: {
        title: 'Info works!',
    },
    // simple function using string literal and 
    // see the cs.translation.ts for another exam
    langsSupported: function (n) { return "This c
};
// TODO : pridaj do blogpostu errory kompilacne
// asi aj optimalizuj image

```

8 / 11 requests | 3.8 MB / 3.8 MB transferred...

{} Line 1, Column 1

flow^up

Q&A

Read more at

bit.ly / typed-translations

Vojtech Mašek

Head dev @ FlowUp

/ vmasek

/ VojtechMasek

/ @vmasek