

MAPPEMONDE

par A. Vuillemin.

F. GELLÉE

PAR COMBETTE, ÉDITEUR

Rue Paul Ambroise Loti, 12

Paris

Localization done bien

Trainline × CocoaHeads Paris, April 2018

MAPPEMONDE

par A. Vuillemin.

F. GELLÉE

PAR COMBETTE, ÉDITEUR

Rue Paul Ambroise Loti, 12

Paris

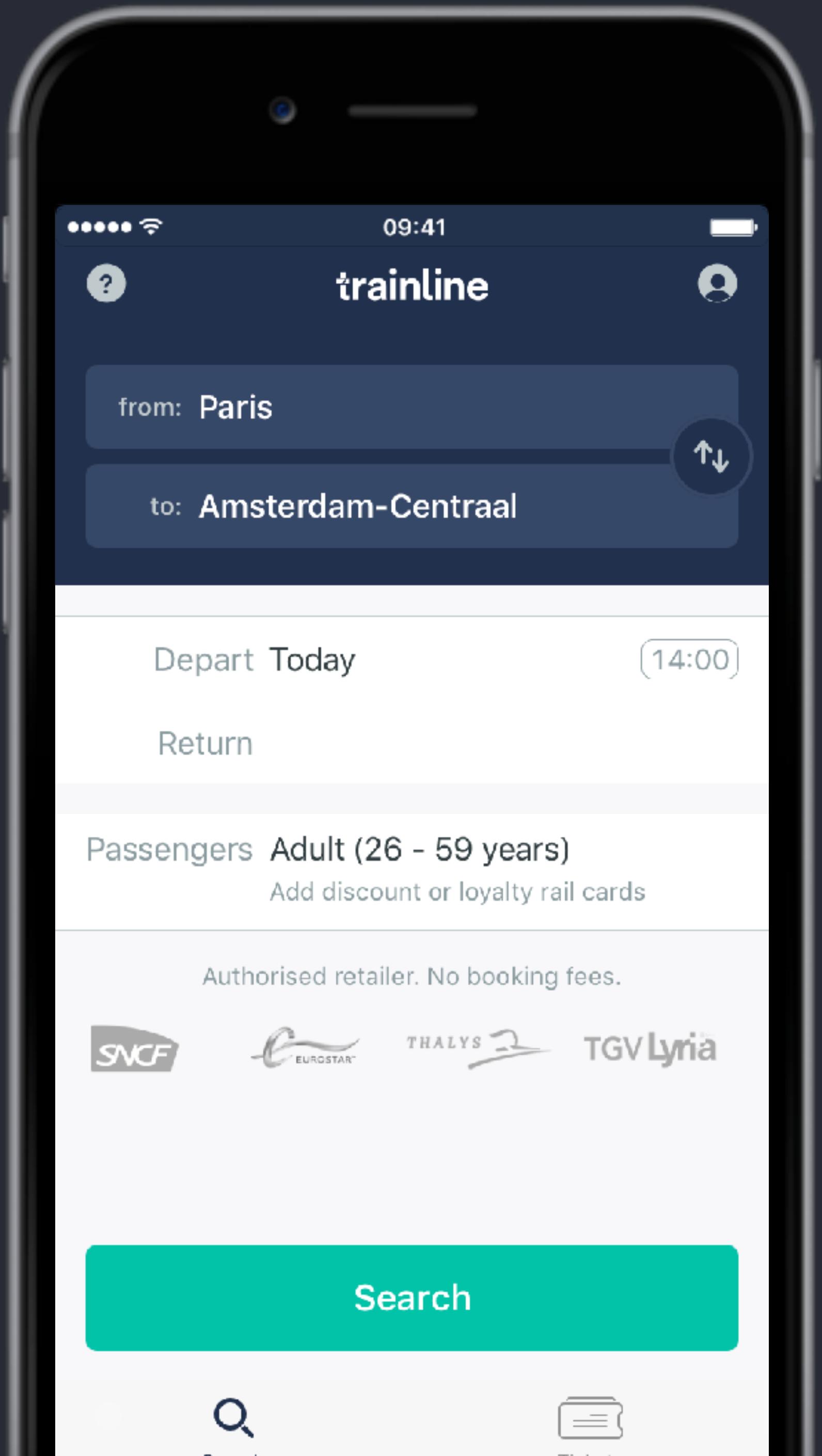
Localisation done bien

Trainline x CocoaHeads Paris, April 2018



DISCLAIMER

This talk will be given in fr_FR and en_MX





14 languages



14 locales

[correct] localisation?

- Travellers won't buy domestic tickets if they don't feel the app is domestic
- Conversion rate is higher if localised in customer's language
- Bad localisation is worse than no localisation

Localisation

Words → Translator



Internationalisation

Code → Developer



Make your app feel so native in any language
that people think it has been written in their own language.

It is difficult

What is a locale?

[*language*] + [*formatting conventions*] ON A [*territory*] USING A [*codeset*]

FR ES
EN NL

Date Time Numbers
Currencies Durations

France
Great Britain

UTF-8
UTF-16

Netherlands

Mexico

It's not a matter of languages...



en_US

Luggage

Target tourists
that probably have
never traveled by train



en_GB

Baggage

Target commuters
going to London

...or of countries alone



nl_NL



nl_BE



fr_BE



fr_FR



fr_CA



en_CA



en_US

Grammar

ru_RU

**has 4 different
forms of plural**

is_IS

**has 15 to 20
indefinite articles**



ahh, formatting too



fr_FR

jeudi 12 avril
19h15

204 €



en_GB

Thursday 12 April
19:15

£178



es_MX

jueves, 12 de abril
7:15 PM

\$ 4602 MXN

and let's not forget about typography

¿Cómo podemos ayudarte?



Comment pouvons-nous vous aider ?



and let's not forget about typography



« Je recommande l'usage de Trainline, surtout pour les entreprises ayant plusieurs centaines de voyages tous les mois. »

Tarif :
PROMOTION STANDARD



"I definitely recommend using Trainline for Business for all companies that book hundreds of trips per month."

Refund and exchange:
PROMOTION STANDARD



train



(Localeapp.com)

Translations ▾

daySummary

ui ios daySummary manypassengers

fr à %{departuretime}, %{passengers} prennent le train de %{departurestation} à %{arrivalstation}.
en at %{departuretime}, %{passengers} are taking the train from %{departurestation} to %{arrivalstation}.

onDay

ui ios daySummary onDay

fr one Le %{day}
 other Le %{day}:
en one On %{day}
 other On %{day}:

onepassenger

ui ios daySummary onepassenger

fr à %{departuretime}, %{passenger} part de %{departurestation} à destination de %{arrivalstation}.
en at %{departuretime}, %{passenger} leaves from %{departurestation} to %{arrivalstation}.

selfpassenger

ui ios daySummary selfpassenger

fr à %{departuretime}, vous partez de %{departurestation} pour %{arrivalstation}.
en at %{departuretime}, you are taking the train from %{departurestation} to %{arrivalstation}.

phraseapp.com

PhraseApp

Trainline Europe adhumi

Trainline Europe > ios

en (en) Search translations Advanced Search Best match All filters

ui.ios.daySummary.manypassengers
at %{departuretime}, %{passengers} are taking the train from %{departurestation} to %{arrivalstation}.

ui.ios.daySummary.onepassenger
at %{departuretime}, %{passenger} leaves from %{departurestation} to %{arrivalstation}.

ui.ios.daySummary.selfpassenger
at %{departuretime}, you are taking the train from %{departurestation} to %{arrivalstation}.

ui.ios.daySummary.onDay
On %{day}
On %{day}:
On %{day}
On %{day}:

en (en) Discard Exclude Placeholders: %{departuretime} %{passengers} %{departurestation} %{arrivalstation}

at %{departuretime}, %{passengers} are taking the train from %{departurestation} to %{arrivalstation}.

Verify Save

en (en)

Context Comments (0) Suggestions (8) More

Key ui.ios.daySummary.manypassengers

Description No description available.

Screenshot No screenshot attached.

Context View Context View not available.

Tags No tags assigned.

Jobs

Jobs Organise all your translation tasks by assigning team members, providing briefings and setting due dates to the locales of your choice.

⚙️ Verifying configuration...



🛠️ Running with required locales ["zh-Hans", "cs", "es", "nb", "de", "en", "it", "pl", "pt-PT", "nl", "sv", "fr", "pt-BR", "da"]

🔎 Finding string keys required by scanning app source code and xibs (shelling out)...



🔄 Fetching latest strings from remote service...



💻 Updating local strings tables...



👌 No translation issues detected.

🎉 Complete 🎉

```
__attribute__((annotate("returns_localized_nsstring")))
NSString *CATLocalized(NSString *key);
NSString *CATLocalized(NSString *key, NSString *defaultText);
NSString *CATLocalized(NSString *key, NSInteger count);
NSString *CATLocalized(NSString *key,
                      NSDictionary <NSString *, id> *substitutions);

NSString *CATUnlocalized(NSString *);
```

```
__attribute__((annotate("returns_localized_nsstring")))
NSString *CATLocalized(NSString *key);

CATLocalized(@"ui.ios.session.disconnected")
```

```
__attribute__((annotate("returns_localized_nsstring")))
NSString *CATLocalized(NSString *key, NSString *defaultText);

- (NSString *)nameForCarrier:(NSString *)carrier {
    NSString *key =[@"data.carrier." stringByAppendingString:self];
    return CATLocalized(key.lowercaseString, carrier);
}
```

```
__attribute__((annotate("returns_localized_nsstring")))
NSString *CATLocalized(NSString *key, NSInteger count);
```

```
CATLocalized(@"ui.ios.exchange.passengerWarning", self.passengers.count)
```

"You must select the passenger"

"You must select the %{count} passengers"

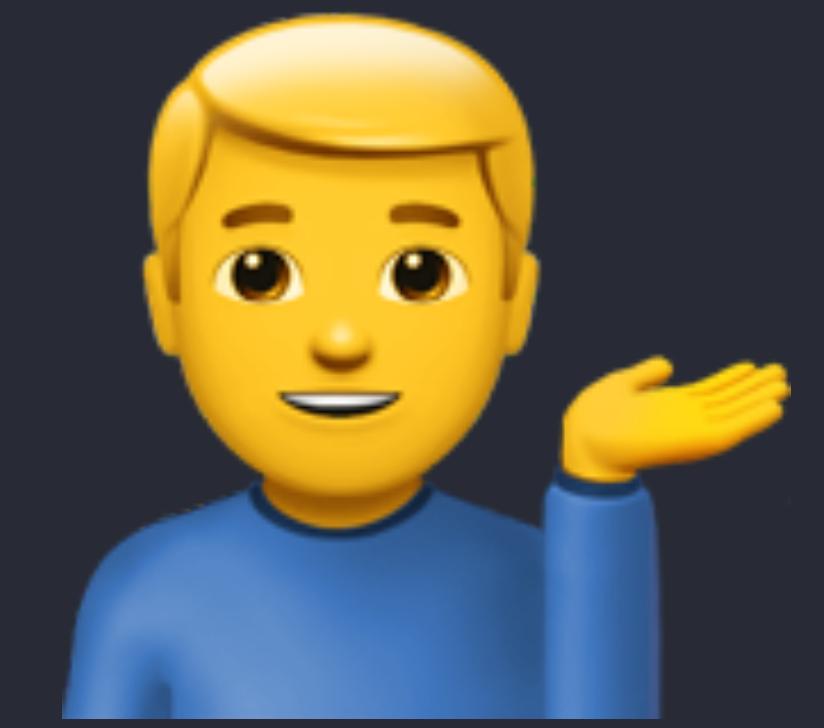
```
__attribute__((annotate("returns_localized_nsstring")))
NSString *CATLocalized(NSString *key,
                      NSDictionary <NSString *, id> *substitutions);

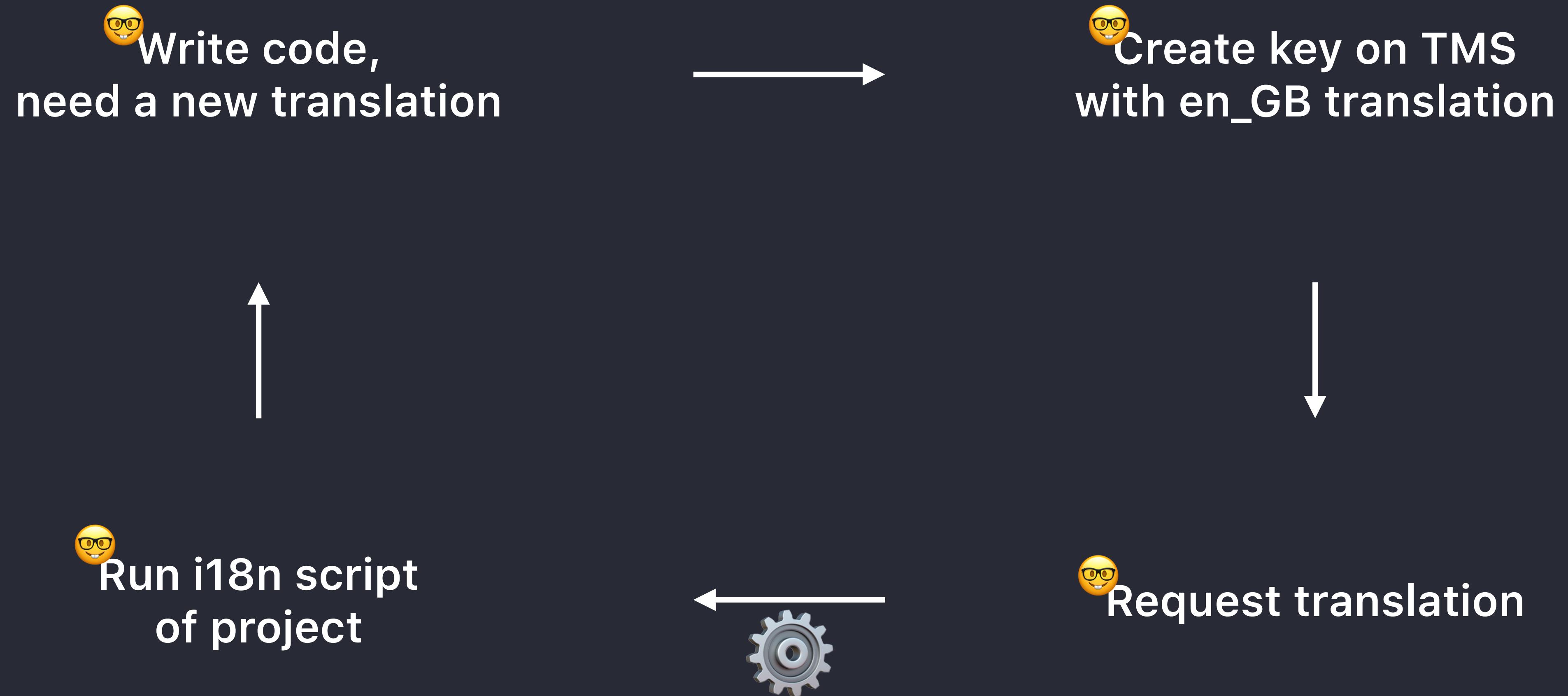
CATLocalized(@"ui.ios.profiles.addYouthAge",
             @{@"age":self.currentProfile.ageDescription})
```

```
NSString *CATUnlocalized(NSString *);
```

```
CATUnlocalized(@"https://192.168.0.1:3000")
```

⚠ Debug code only



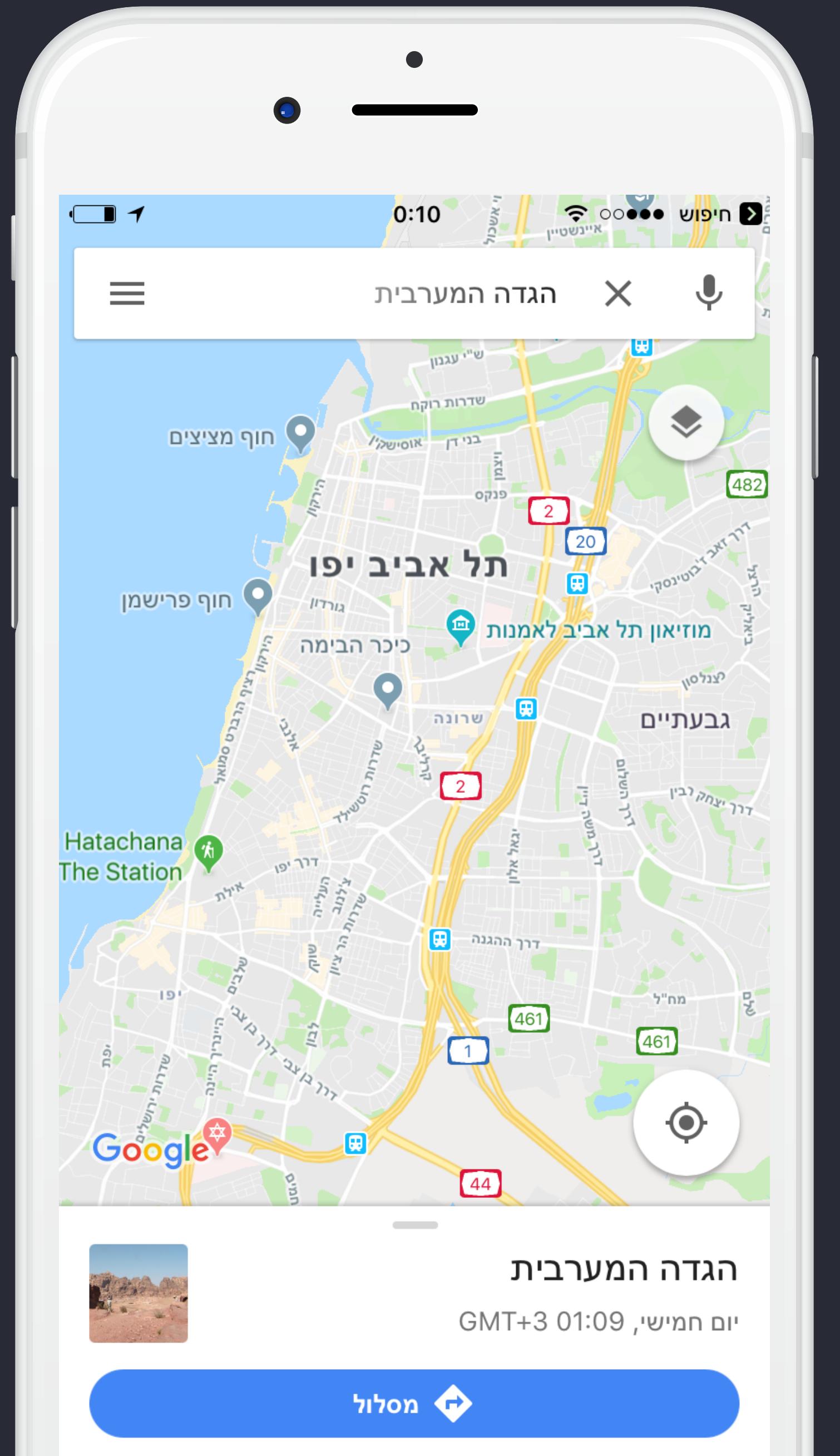


Some stuff isn't localised

and that's okay

What should we take care of?





right-to-left languages

Out of the box
support from iOS 9 up

two rules of thumb:

```
NSLayoutConstraint.activate([
    rightToLeftTextView.leadingAnchor.constraint(equalTo: superview.leadingAnchor),
    rightToLeftTextView.trailingAnchor.constraint(equalTo: superview.trailingAnchor),
])

rightToLeftTextView.textAlignment = .Natural

let attribute = rightToLeftTextView.semanticContentAttribute
let layoutDirection = UIView.userInterfaceLayoutDirectionForSemanticContentAttribute(attribute)
if layoutDirection == .RightToLeft {
    rightToLeftTextView.textAlignment = .Left
}
```

first rule:

```
NSLayoutConstraint.activate([
    rightToLeftTextView.leadingAnchor.constraint(equalTo: superview.leadingAnchor),
    rightToLeftTextView.trailingAnchor.constraint(equalTo: superview.trailingAnchor),
])
```

```
rightToLeftTextView = UITextView()
let attribute = NSMutableAttributedString(string: "Right-to-left text")
let layoutDirection = UILayoutDirection(rawValue: "rtl")
if layoutDirection == UILayoutDirection(rawValue: "rtl") {
    rightToLeftTextView.textAlignment = .Left
}
```

Use ***leadingAnchor*** and ***trailingAnchor*** instead
of ***leftAnchor*** and ***rightAnchor***

second rule:

The locale comes with a defined
natural order (RTL or LTR)

```
NSLayoutConstraint.activate([
    rightToLeftTextView.leadingAnchor.constraint(equalTo: leftView.trailingAnchor),
    rightToLeftTextView.trailingAnchor.constraint(equalTo: rightView.leadingAnchor),
])
```

```
rightToLeftTextView.textAlignment = .Natural
```

```
let attribute = rightToLeftTextView.semanticContentAttribute
let layoutDirection = UIView.userInterfaceLayoutDirectionForSemanticContentAttribute(attribute)
if layoutDirection == .RightToLeft {
    rightToLeftTextView.textAlignment = .Left
}
```

and an exception...

```
NSLayoutConstraint.activate([  
    rightToLeftTextView.centerXAnchor.constraint(equalTo: view.centerXAnchor),  
    rightToLeftTextView.centerYAnchor.constraint(equalTo: view.centerYAnchor),  
])  
  
rightToLeftTextView.semanticContentAttribute = .forceRightToLeft
```

Override the right to left alignment on
any specific text view

```
let attribute = rightToLeftTextView.semanticContentAttribute  
let layoutDirection = UIView.userInterfaceLayoutDirectionForSemanticContentAttribute(attribute)  
if layoutDirection == .RightToLeft {  
    rightToLeftTextView.textAlignment = .Left  
}
```

will you flip... or not?



MOST CONTROLS

Segmented controls
Progress indicators
Outline views

Video/timeline indicators

Images
Clocks
Music notes
Graphs

Use more powerful file
formats

better plurals handling



.stringdicts
to the rescue

better plurals handling

```
<key>$Key$</key>
<dict>
    <key>NSStringLocalizedFormatKey</key>
    <string>%#@variable@</string>
    <key>variable</key>
    <dict>
        <key>NSStringFormatSpecTypeKey</key>
        <string>NSStringPluralRuleType</string>
        <key>NSStringFormatValueTypeKey</key>
        <string>d</string>
        <key>zero</key>
        <string><#Optional string for zero#></string>
        <key>one</key>
        <string><#String with %d for one#></string>
        <key>other</key>
        <string><#String with %d for other#></string>
    </dict>
</dict>
```

better plurals handling

```
<key>$Key$</key>
```

```
<dict>
```

The key to grab the translation
on our code

```
<string>NSStringPluralRuleType</string>
<key>NSStringFormatValueTypeKey</key>
<string>d</string>
<key>zero</key>
<string><#Optional string for zero#></string>
<key>one</key>
<string><#String with %d for one#></string>
<key>other</key>
<string><#String with %d for other#></string>
</dict>
</dict>
```

better plurals handling

```
<key>$Key$</key>
<dict>
    let format = NSLocalizedString("$Key$", comment: "A useful comment")
    return String(format: format, locale: Locale.current, arguments: 2)
</dict>
```

```
<key>NSStringFormatSpecTypeKey</key>
<string>NSStringPluralRuleType</string>
<key>NSStringFormatValueTypeKey</key>
<string>d</string>
<key>zero</key>
<string><#Optional string for zero#></string>
<key>one</key>
<string><#String with %d for one#></string>
<key>other</key>
<string><#String with %d for other#></string>
</dict>
</dict>
```

better plurals handling

```
<key>$Key$</key>
<dict>
    <key>NSStringLocalizedFormatKey</key>
    <string>%#@variable@</string>
    <key>variable</key>
    <dict>
        <key>placeholder</key>
        <string>%#g</string>
        <key>NSStringFormatValueTypeKey</key>
        <string>d</string>
        <key>zero</key>
        <string><#Optional string for zero#></string>
        <key>one</key>
        <string><#String with %d for one#></string>
        <key>other</key>
        <string><#String with %d for other#></string>
    </dict>
</dict>
```

The value for the key
and placeholder

better plurals handling

```
<key>$Key$</key>
<dict>
    <key>NSStringLocalizedForString</key>
    <string>%#@variable@</string>
    <key>variable</key>
    <dict>
        <key>NSStringFormatSpecTypeKey</key>
        <string>NSStringPluralRuleType</string>
        <key>NSStringFormatValueTypeKey</key>
        <string>d</string>
        <key>zero</key>
        <string><#Optional string for zero#></string>
        <key>one</key>
        <string><#String with %d for one#></string>
        <key>other</key>
        <string><#String with %d for other#></string>
    </dict>
</dict>
```

The plural rules applying for the first declared variable

?

?

?

?

?

?

?



?

?

?

?

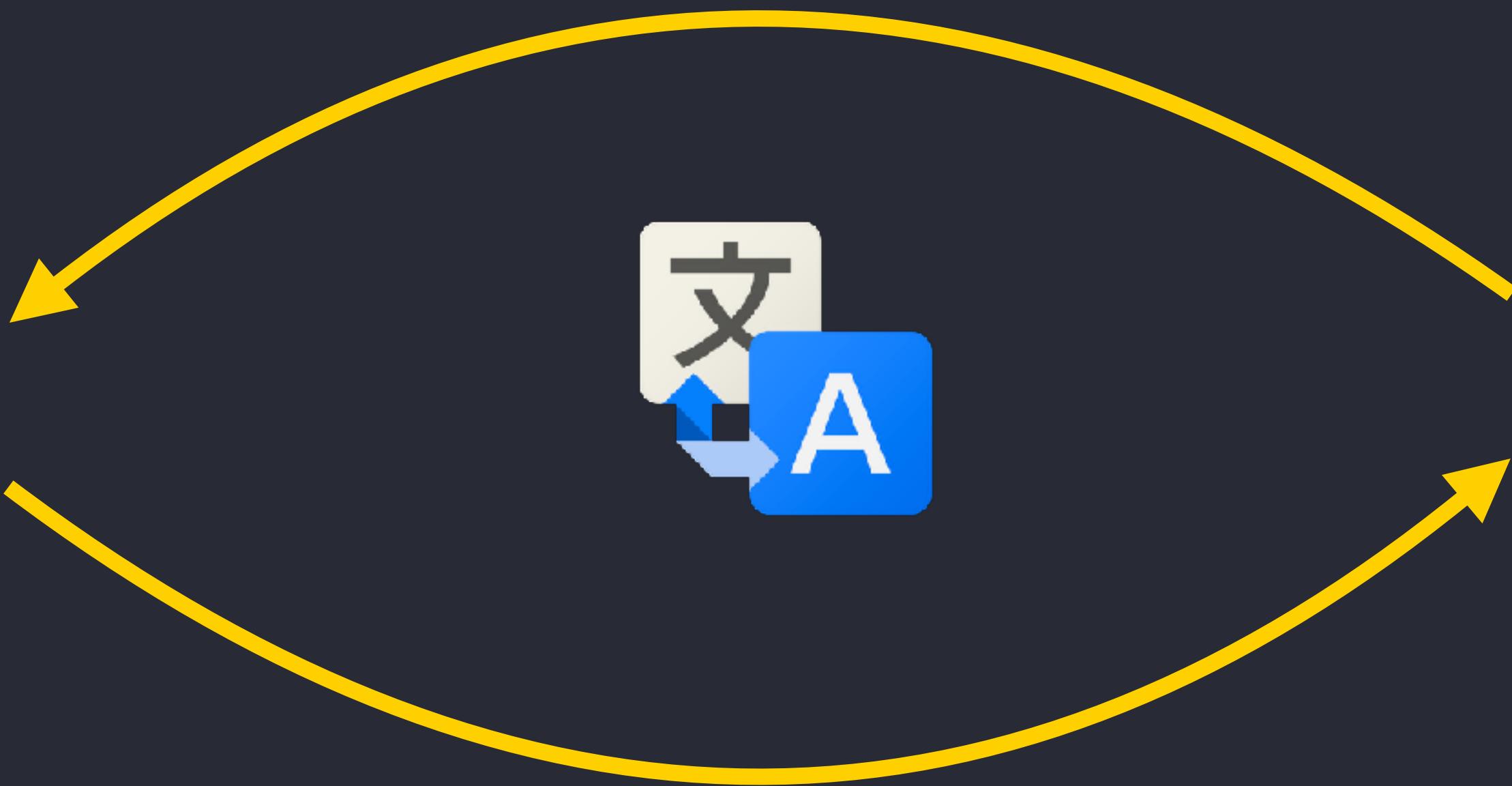
?

?

?

?

xliff anyone?



xliff anyone?

XML Localisation Interchange
File Format



Xcode 9 goodies for localisation

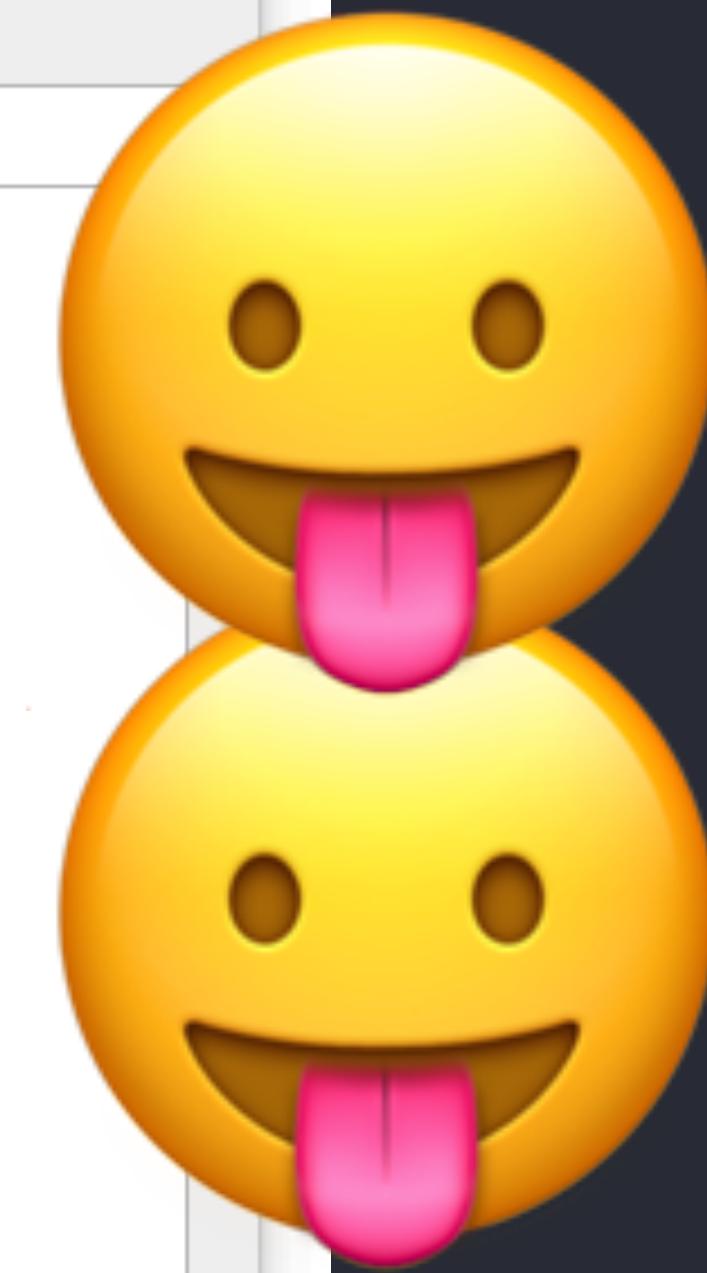
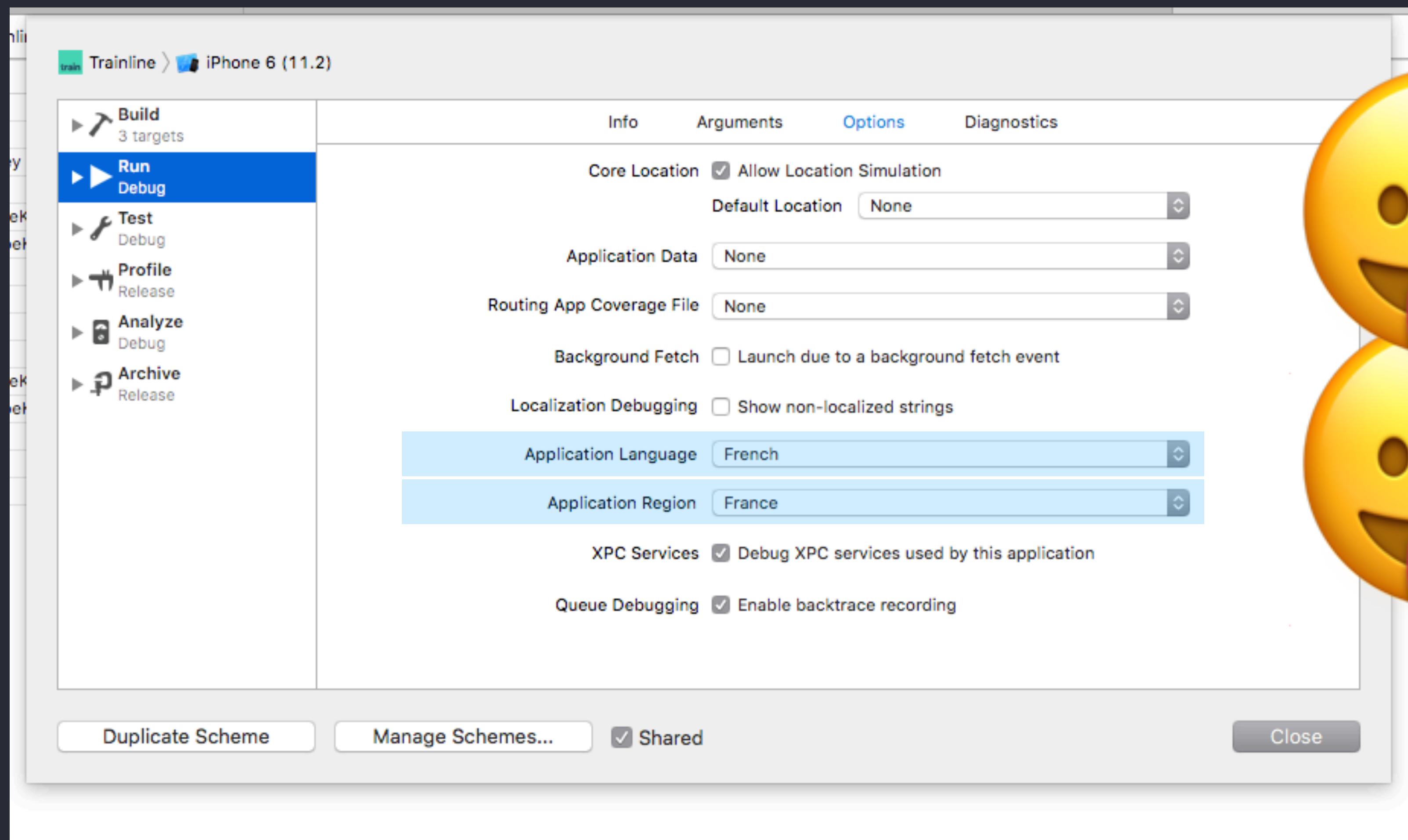


property list view for plurals

Key	Type	Value
▼ Strings Dictionary	Dictionary	(1 item)
▼ trains.and.coaches	Dictionary	(3 items)
NSStringLocalizedFormatKey	String	you have %#@trainvar@ and %#@coachvar@
▼ trainvar	Dictionary	(5 items)
NSStringFormatSpecTypeKey	String	NSStringPluralRuleType
NSStringFormatValueTypeKey	String	d
zero	String	no trains
one	String	%d train
other	String	%d trains
▼ coachvar	Dictionary	(5 items)
NSStringFormatSpecTypeKey	String	NSStringPluralRuleType
NSStringFormatValueTypeKey	String	d
zero	String	no coaches
one	String	%d coach
other	String	%d coaches



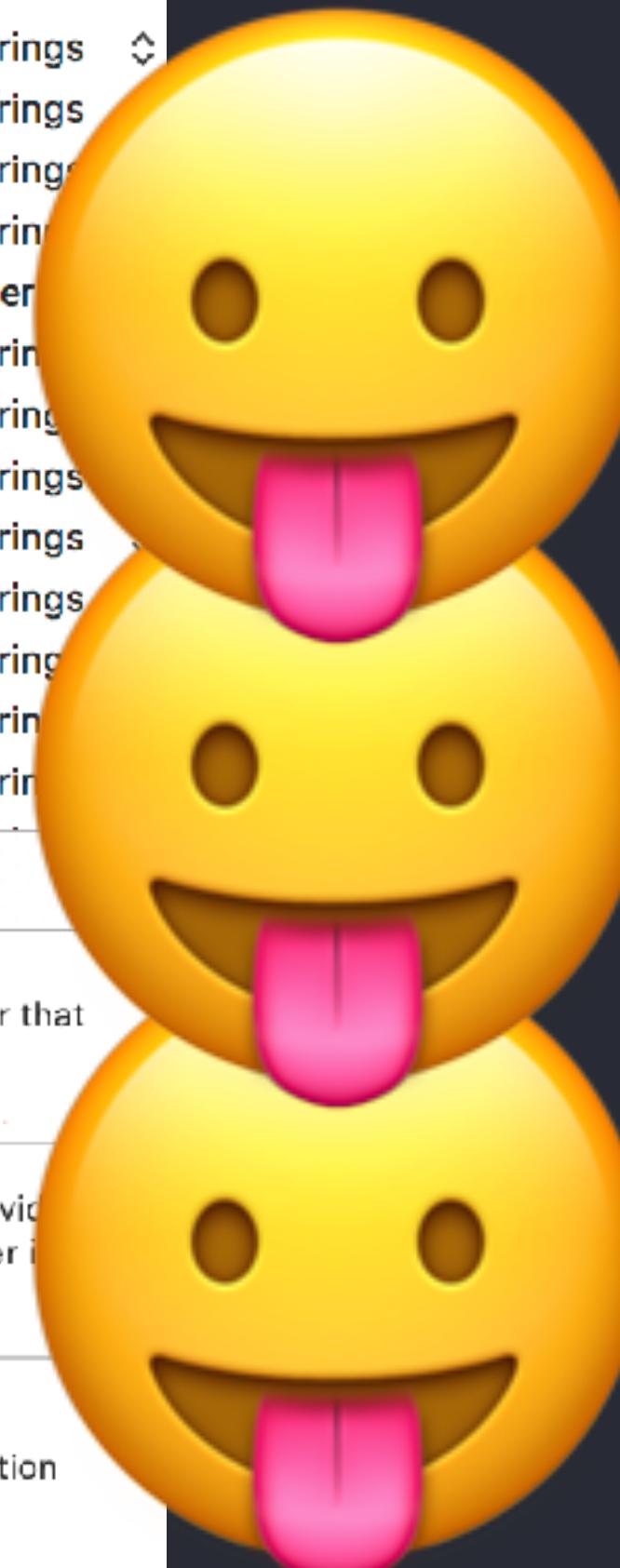
test your localisations with ease



localise images

The screenshot shows the Xcode interface. On the left is a storyboard preview window displaying a white screen with a blue and white graphic element and the text "EUROSTAR LANZAMIENTO 2018". On the right is the "Localization" editor, which lists supported languages: Base, English, French, German, Italian, Spanish, Danish, Norwegian Bok..., Swedish, Portuguese (Br...), Portuguese (P...), Chinese (Simpl...), Dutch, and Czech. Each language has a "Localizable Strings" entry next to it. Below the localization table, there are four items listed:

- View Controller** - A controller that manages a view.
- Storyboard Reference** - Provides a placeholder for a view controller in an external storyboard.
- Navigation Controller** - A controller that manages navigation through a hierarchy of views.
- Table View Controller** - A controller that manages a table view.





Don't us sentences as
localization keys

Don't hardcode string

Don't concatenate strings

“Cancel inward of Adrien”



~~ui.ticket.cancel + ticketPart + ui.of + passengerName~~

"Cancel %{part} of %{passenger}"



« Rückfahrt von Adrien stornieren »

Don't assume punctuations

Don't use the same string
in multiple places



es_ES



en_GB

Cancelar

action

Anular

ticket

Cancel

Cancel

Don't use reflexion



Follow typographic rules

Take care of plurals

Departing on %{date}, one passenger

Departing on %{date}, %{count} passengers

**Use meaningful token names
over position**

The \$1 card ending in \$2 expires on \$3

The %{type} card ending in %{lastDigit} expires on %{date}

Use unicode by default

Localise all the things

including accessibility, server error messages, etc.

Cleanup useless locales



Questions?

@adhum & @vidriloc
trainline

👋 We are hiring an iOS lead