The Road to Intl.MessageFormat

Or, what I’ve been working on for the last few years
Eemeli Aro

- Staff Software Engineer at Mozilla
- Maintainer of yaml, fluent & other npm packages
- OpenJS Foundation CPC voting member
- A Mozilla representative at TC39, ECMA-402 & W3C Intl WG
- A Mozilla+OpenJSF representative at the Unicode Consortium MessageFormat WG
Where are we now?

- Localisation is often an afterthought, encountered when expanding beyond a product’s first market.
- Multiple different solutions that **mostly** work — until they don’t.
- Localisation is focused on translating strings from one locale to another.
What would be better?

- Content on the web should be easily localisable.
- There should be a commonly used message & resource syntax.
- We should have tools and technologies for localising complex structures, **not just simple strings**.
- It should be easy to make the right choices by default.
Brussels is the capital of Belgium.
Brussels is the capital of Belgium.
Brussels is the capital of Belgium.
- What's the format of this `messages.mf` file?
- Is this going to work in all browsers?
- How do I use this from my existing app?
STANDARDISE
ALL THE THINGS
${name} invited you to their party.
match {$host :gender}
when female {{$name} invited you to her party.}
when male {{$name} invited you to his party.}
when * {{$name} invited you to their party.}
let $relDate = {$date : relativetime unit=day}
match {$host : gender}
  when female {{name} invited you to her party {$relDate}.}
  when male {{name} invited you to his party {$relDate}.}
  when * {{name} invited you to their party {$relDate}.}
let $countOther = {$count :number offset=1}
let $relDate = {$date :relativetime unit=day}

match {$countOther} {$host :gender}
when 0 female {{$name} invited you to her party {$relDate}.}
when 0 male {{$name} invited you to his party {$relDate}.}
when 0 * {{$name} invited you to their party {$relDate}.}
when 1 * {{$name} invited you and one other to their party {$relDate}.}
when * * {{$name} invited you and {$countOther} others to their party {$relDate}.}
Adding MF2 support to the browser DOM & JavaScript

- Removing the need to include a message parser in your JS bundle.
- First such addition to JavaScript since `JSON.parse()`.
- Internally, cross-browser support is provided by ICU.
But what about my old app?

- We’ll make it faster.
- We’re not only building a shiny new world, but also the path getting us from here to there.
- Not only defining a universal syntax, but also a universal data model for messages.
- Any existing message syntax can be parsed into the “MF2” data model, allowing for a shared runtime and conversion between all formats.
Is this really happening?

- **Yes.** This is how Firefox is already localised, using Fluent.
- Message syntax is being defined within the Unicode CLDR TC.
- A tech preview of MF2 is included in ICU 72, released in October 2022.
- The Intl.MessageFormat & MessageResource proposals are progressing in TC39, currently at Stage 1.
It’s all public

- MessageFormat 2: unicode-org/message-format-wg
- DOM Localisation: nordzilla/dom-l10n-draft-spec
- Project Fluent: projectfluent.org
- Me:
  - @eemeli@mefi.social on Mastodon
  - @eemeli on GitHub
  - eemeli@mozilla.com
- This talk: eemeli.org/talks/road-to-intl-messageformat