

Paris EmberJS Lab #6

Novembre 2019



Guillaume Gérard (aka GreatWizard)



Projets :

[GreatWizard/ember-circleci](#) : Générer votre configuration CircleCI pour app ou addons

[GreatWizard/ember-service-worker-unregistration](#) : Désenregistrer vos services workers

[peopledoc/ember-feature-controls](#) : Activer / désactiver vos feature flags avec une interface d'administration et permettre de garder l'état dans le localStorage.



[@ggerard88](#)



[@GreatWizard](#)

Migrer de ember-i18n à ember-intl

- Guillaume Gérard

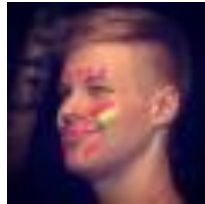


Migrer de ember-i18n à ember-intl

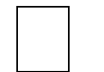
Nouvelle annoncée le 26 oct. 2018 :

emberjs.com/blog/2018/10/26/the-ember-times-issue-70.html

Ember-i18n est déprécié en faveur de [ember-intl](#) :

- Union des efforts: [buschtoens](#)  et [jasonmit](#) 
- Utilise l'API Intl
https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Intl

Fonctionnalités

-  Syntaxe de message ICU. Pluralisation et segments formatés (nombres, dates / heures, etc.)
-  Prise en charge de plus de 150 langues (avec fallback)
-  Construit en grande partie sur des normes : Syntaxe de message ICU & API Native Intl
- ⚡ Service Ember pour le formatage et la traduction

Fonctionnalités tenant compte des paramètres régionaux

💰 Mise en forme des devises, des décimales et des pourcentages

```
{{format-number num style='currency' currency='EUR'}}  
//> 123.456,79 €
```

📅 Formatage des dates et heures

```
let now = new Date(2018, 10, 29, 20, 45, 00);  
{{format-date now day='numeric', month='short' hour='numeric' minute='numeric' hour12=false}}  
//> 29 nov. à 20:45  
{{format-time now format='hhmmss'}}  
//> 29/11/2018
```

🕒 Affichage de l'heure relative: “maintenant”, “hier”, “il y a 2 mois”

```
let date = new Date().setDate(new Date().getDate() - 3);  
{{format-relative date}}  
//> il y a 3 jours
```

Recette de migration

Temps
30 min

Très facile

Bon marché

1. **Supprimer ember-i18n** de vos dépendances
2. **Supprimer la config i18n** du fichier **config/environment.js**
3. Installer ember-intl :

```
ember install ember-intl
```

4. Installer l'outil de migration de Dockyard et transformer les fichiers de traductions :

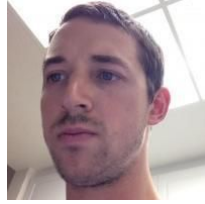
```
npx ember-i18n-intl-migrator --type=yaml
```

5. Exécuter les codemods :

```
npx jscodeshift -t  
https://raw.githubusercontent.com/DockYard/ember-i18n-to-intl-migrator/master/lib/codemod.js app/
```

Problèmes ?!

- Et si on utilise ember-cp-validations avec **ember-i18n-cp-validations** ?

Il existe aussi [@ember-intl/cp-validations](#) (par [jasonmit](#) ).

- Et si une bibliothèque utilise encore ember-i18n?

L'API publique de l'addon ember-intl est entièrement compatible avec ember-i18n. Vous pouvez donc enregistrer le service intl en tant que i18n dans un instance-initializer :

```
export function initialize(appInstance) {
  let intl = appInstance.resolveRegistration('service:intl')
  appInstance.register('service:i18n', intl)
}

export default {
  initialize,
  after: 'ember-intl'
}
```

Migrer de ember-i18n à ember-intl

