Engineer

Internationalisation

Concevoir un logiciel facile à localiser

Ce que c’est

L’internationalisation est le travail d’ingénierie qui permet à un produit de fonctionner dans de nombreuses langues et régions sans modifier son code pour chacune d’elles. Le W3C la définit comme la conception et la construction d’un produit, d’une application ou d’un document qui pourra ensuite être facilement localisé, quels que soient la culture, la région ou la langue de son public. En anglais, le mot s’écrit internationalization ou internationalisation et s’abrège en i18n, car 18 lettres séparent son i initial de son n final. Certaines personnes désignent la même idée par le terme globalisation.

Le W3C résume la répartition du travail en une phrase : les personnes qui conçoivent et construisent les produits les internationalisent « pour qu’ils puissent être facilement localisés par les traducteurs et les spécialistes de la localisation ». L’internationalisation se fait une fois, dans le code et les modèles. La localisation se répète pour chaque marché, dans des données que le code lit. Un produit peut être entièrement internationalisé et n’être toujours distribué que dans une seule langue.

L’unité commune aux deux est la locale. La spécification de CLDR définit une locale comme un identifiant des préférences que les utilisateurs de vastes parties du monde ont tendance à partager : une langue, souvent précisée par une région, de sorte que `en-GB` et `en-US` partagent une langue mais diffèrent dans leur manière d’écrire une date. L’identifiant sélectionne à la fois une traduction et un ensemble de conventions.

Le W3C distingue quatre types de travail. Le premier supprime les obstacles : le produit traite le texte en Unicode, ou gère délibérément un ancien encodage de caractères lorsqu’il le faut ; il ne construit pas les phrases en assemblant des fragments ; et son code ne dépend jamais du libellé présenté à l’utilisateur. Le

deuxième ajoute des capacités dont seule une langue ultérieure aura besoin, comme le balisage d'un texte mêlant écritures de droite à gauche et de gauche à droite, un moyen de déclarer la langue d'un passage et des règles de style pour le texte vertical. Le troisième permet au code d'utiliser les conventions locales, généralement grâce à des bibliothèques existantes plutôt qu'à des tables rédigées par l'équipe : formats de date et d'heure, calendriers, formats des nombres et systèmes de numération, ordre de tri des listes, conventions pour écrire les noms des personnes et s'adresser à elles. Le quatrième garde tout ce qui est traduisible hors du code source, pour charger la bonne version pour chaque lecteur.

Utilisez le concept pour prendre une décision

Utilisez « Qu'est-ce que l'internationalisation ? » pour éclairer une décision réelle. Nous pouvons vous aider à relier ce concept au système que vous construisez ou achetez. Un logiciel qui se lit naturellement sur chaque marché où vous vendez, y compris les dates, les devises et le classement des noms.

Découvrir nos prestations Contactez-nous →

Vérifiez les hypothèses derrière le terme

Appuyez-vous sur l'explication et les réalisations liées pour formuler les questions de votre projet. Nous pouvons examiner les hypothèses avec vous avant le choix d'un outil. Système i18n anglais/danois — 12,027 messages par langue, en anglais et en danois

Système i18n anglais/danois Contactez-nous →

Pourquoi c'est important

L'argument du W3C porte sur le moment de s'y prendre. Adapter au marché mondial un produit construit autour d'une langue et d'une culture demande bien plus d'efforts et de temps que de le concevoir pour le monde entier dès le départ. Le W3C compare ce travail aux efforts consacrés au problème Y2K pour corriger les champs d'année à deux chiffres qui supposaient que toutes les années commençaient par 19. Entreprise tardivement, l'internationalisation impose souvent une réingénierie délicate et coûteuse. Entreprise tôt, elle se résume à quelques habitudes peu coûteuses le jour où on les adopte.

Le coût d'une habitude manquante augmente avec chaque langue. Une phrase assemblée dans le code à partir de trois fragments représente un défaut dans un produit anglais et cinq défauts dans un produit doté de cinq locales. Dans chaque cas, la correction n'est pas une meilleure traduction, mais une modification du code, revue et publiée comme toute autre. Il en va de même d'une date formatée à la main, d'une liste triée en comparant des octets ou d'un libellé qu'un test recherche par son texte anglais.

Ces habitudes sont utiles même si rien n'est jamais traduit, comme le W3C le précise expressément. Un produit vendu uniquement en anglais reçoit tout de même des noms écrits avec des lettres absentes de l'anglais, des adresses structurées selon les conventions d'un autre pays et des dates provenant de systèmes qui placent le jour avant le mois. Le stockage du texte en Unicode et la conservation des valeurs de date jusqu'au moment de leur affichage permettent de gérer tout cela sans cas particulier.

À quoi cela ressemble en pratique

Le test le plus rapide pour savoir si un produit est internationalisé est une phrase contenant un nombre. La version qui ne l'est pas construit la phrase dans le code :

```
const label = count + (count === 1 ? " day" : " days");
```

C'est correct en anglais et impossible à traduire. L'ukrainien exige trois formes pour les nombres entiers et une quatrième pour les fractions : день après 1, 21 et 101, дні après 2, 3, 4 et 22, днів après 0, de 5 à 20 et après 25, et дня après une fraction. Quel que soit le mot unique choisi par le traducteur pour la deuxième branche, il sera incorrect pour la plupart des nombres, car 21 prend la même forme que 1, et 2 une forme différente de 5.

Une fois internationalisée, la phrase devient une clé, et la table de chaque langue fournit toutes les formes dont elle a besoin. Dans Hugo, les tables de traduction se trouvent dans le répertoire `i18n`, avec un fichier par langue, nommé d'après son étiquette de langue :

```
# i18n/en.toml
[days]
one = '{{ . }} day'
other = '{{ . }} days'
```

```
# i18n/uk.toml
[days]
one = '{{ . }} день'
few = '{{ . }} дні'
many = '{{ . }} днів'
other = '{{ . }} дня'
```

Le modèle demande la clé et transmet le nombre. Il ne contient aucun mot de l'une ou l'autre langue ni aucune règle de pluriel :

```
{{ T "days" 21 }}
```

Les pages anglaises affichent `21 days` et les pages ukrainiennes `21 день` ; avec 2, elles affichent `2 days` et `2 дні`, et avec 5, `5 days` et `5 днів`. Les noms `one`, `few`, `many` et `other` ne sont pas une invention de Hugo. Ce sont les catégories de pluriel définies par CLDR, la collection de données de locales du Consortium Unicode. La documentation de Hugo renvoie au tableau des règles de pluriel de CLDR pour connaître les catégories de chaque langue. Une nouvelle langue arrive sous la forme d'un nouveau fichier, sans modification d'un modèle.

À grande échelle, les tables deviennent le plus grand ensemble de texte du produit et exigent la même discipline que le code. Un des projets de ce site, un système d'internationalisation anglais-danois, contient 12 027 messages par locale, répartis dans 458 fichiers d'espaces de noms. Chaque fichier est limité à 400 lignes ; un linter compare les clés des deux locales pour qu'aucun message ne soit présent dans l'une et absent de l'autre ; et le danois suit un vocabulaire fixé, jusqu'au tutoiement « du », que l'outillage impose à chaque commit. La page explique que cette limite a permis de garder l'ensemble facile à maintenir : sans limite par fichier, précise-t-elle, il n'y aurait eu que douze fichiers au lieu de 458.

Où les problèmes surviennent

Les phrases sont assemblées par morceaux. Un message construit comme `"Delete " + n + " files?"` parvient au traducteur sous forme de deux fragments privés de leur phrase, dans un ordre propre à l'anglais. Une autre langue peut placer le nombre à la fin ou employer une forme verbale qui dépend du nom. Le W3C inclut l'attention portée à la concaténation des chaînes parmi les travaux nécessaires. Faites de chaque phrase un message unique avec des paramètres nommés que le traducteur peut déplacer ; la documentation de Hugo montre comment `{{ T "age" (dict "name" "Will" "count" 1) }}` insère un nom et une quantité dans une seule phrase traduite.

Le code lit les mots à l'écran. Un test qui retrouve un bouton par son libellé « Save », ou une branche qui compare un état au texte « Paid », fonctionne jusqu'à la première traduction puis échoue sans erreur. Le W3C place l'absence de dépendance du code aux chaînes de l'interface utilisateur parmi les premières

tâches de l'internationalisation. Comparez des clés, des identifiants et des valeurs énumérées, et donnez aux tests un attribut stable pour retrouver un élément.

Une seule règle de pluriel pour toutes les langues. `count === 1` est la règle de l'anglais écrite dans le code, et elle est fausse pour la plupart des langues qu'un produit peut ajouter. La spécification de CLDR définit six catégories de pluriel, zero, one, two, few, many et other, dont seule other est obligatoire, et donne à chaque langue ses règles pour choisir entre elles : l'anglais en utilise deux, l'ukrainien quatre. Transmettez le nombre au message et laissez la table de traduction contenir toutes les catégories utilisées par la langue.

Une traduction manquante passe inaperçue. Lorsqu'une clé manque dans la langue courante, `T` de Hugo se replie sur la langue par défaut et renvoie une chaîne vide si la clé y manque aussi. Une page peut donc être publiée avec un blanc ou une expression anglaise au milieu d'une phrase danoise. Lancez la génération avec `--printI18nWarnings` pour recenser toutes les traductions manquantes et les replis, activez `enableMissingTranslationPlaceholders` dans un environnement de prévisualisation pour rendre les blancs visibles sur la page, et comparez les clés de toutes les locales en intégration continue.

Les dates et les nombres sont formatés à la main. Un modèle qui écrit `MM/DD/YYYY`, ou un nombre avec une virgule tous les trois chiffres, a inscrit les conventions d'un pays dans le code. Le même jour s'écrit `10/27/26` en anglais américain, `27/10/2026` en anglais britannique, `27.10.2026` en danois et `2026-10-27` en suédois ; le nombre 12,027.5 s'écrit `12.027,5` en danois, tandis qu'en norvégien, suédois et ukrainien il prend une espace comme séparateur de milliers et une virgule avant les décimales. La spécification de CLDR recommande de stocker et d'échanger des données indépendantes de la langue, puis de les formater pour le lecteur. Conservez les dates et les montants comme des valeurs sans format jusqu'à la dernière étape, et formatez-les alors avec les fonctions de la plateforme qui tiennent compte de la locale. Dans Hugo, `time.Format ":date_full"` affiche une date sous la forme `Wednesday, November 3, 2021` sur une page anglaise et `Mittwoch, 3. November 2021` sur une page allemande.

Le texte n'est pas stocké en Unicode. Un produit qui écrit « blåbærsyltetøj » en UTF-8 puis le relit en Windows-1252 affiche « blÃ¥bÃ¦rsyltetÃ¸j » ; s'il stocke le texte dans un ancien encodage limité à une langue, il ne peut pas même contenir un nom ukrainien. Activer Unicode est le premier point de la liste du W3C.

Utilisez un même encodage Unicode, généralement UTF-8, de la base de données au navigateur, et déclarez-le à chaque frontière où les octets deviennent du texte.

Pour aller plus loin

- Localisation et internationalisation — les définitions de travail des deux termes par le W3C, les quatre types de tâches de l'internationalisation et les raisons de s'y prendre dès le départ.
- Hugo : mode multilingue — tables de traduction, dates et devises localisées, et signalement des traductions manquantes par Hugo.
- Hugo : lang.Translate — ordre de recherche entre les langues, catégories de pluriel des tables de traduction et messages à paramètres nommés.
- UTS #35, partie 1 : socle — la spécification de CLDR : identifiants de locales, héritage entre elles et conseil de conserver les données indépendantes de la langue jusqu'à leur affichage.
- UTS #35, partie 3 : nombres — symboles et modèles numériques par locale, catégories de pluriel et règles pour les choisir.



Glossaire

Choisissez selon vos contraintes réelles

Présentez le processus de travail, les interfaces existantes et les utilisateurs qui ont besoin du changement. Nous pouvons définir une livraison testable et une transmission permettant à votre équipe d'assurer la maintenance des éléments livrés.

Consulter les tarifs Contactez-nous →

Rendez la terminologie concrète

Demandez comment « Qu'est-ce que l'internationalisation ? » s'applique à votre organisation. Un exemple concret tiré de votre système actuel suffit pour engager la discussion.

Découvrir nos prestations

welcome@engineer.company +45 50 10 61 84 Copenhague, Danemark

Emportez cette page avec vous

Fiche de contact

Créé avec soin et amour par D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

Droit d'auteur © 2025 – aujourd'hui · Engineer ApS · Tous droits réservés.

Imprimé depuis https://engineer.company/fr/glossary/internationalization/