Engineer

Internationalisering

at udforme software, så den let kan lokaliseres

Hvad det er

Internationalisering er det tekniske arbejde, der gør, at ét produkt kan bruges på mange sprog og i mange regioner, uden at koden skal ændres for hvert af dem. W3C definerer det som design og udvikling af et produkt, en applikation eller et dokument, så det let kan lokaliseres til målgrupper, der adskiller sig i kultur, region eller sprog. På engelsk forkortes ordet i18n, fordi der står 18 bogstaver mellem det første i og det sidste n, og forkortelsen bruges også i danske kodebaser. Nogle kalder den samme idé globalisering.

W3C opsummerer arbejdsdelingen i én sætning: designere og udviklere internationaliserer produkter, så oversættere og lokaliseringsspecialister nemt kan lokalisere dem. Internationaliseringen sker én gang, i koden og skabelonerne. Lokaliseringen sker igen for hvert marked, i data, som koden læser. Et produkt kan være fuldt internationaliseret og stadig kun findes på ét sprog.

Den enhed, begge dele arbejder med, er en locale. CLDR's specifikation definerer en locale som en identifikator for en række brugerpræferencer, der typisk deles over store dele af verden: et sprog, ofte afgrænset til en region, så `en-GB` og `en-US` deler sprog, men skriver en dato forskelligt. Identifikatoren vælger på én gang en oversættelse og et sæt konventioner.

W3C deler arbejdet i fire slags. Den første fjerner forhindringer: produktet behandler tekst som Unicode eller håndterer en ældre tegnkodning bevidst, hvor det er nødvendigt, det bygger ikke sætninger ved at sætte stumper sammen, og koden afhænger aldrig af den ordlyd, brugeren ser. Den anden tilføjer understøttelse, som først et senere sprog får brug for, for eksempel markup til tekst, der blander højre-til-venstre- og venstre-til-højre-skrift, en måde at angive, hvilket sprog et afsnit er skrevet på, og stilregler til lodret tekst. Den tredje lader koden bruge lokale konventioner, som regel fra eksisterende biblioteker frem for

tabeller, teamet selv skriver: dato- og tidsformater, kalendere, talformater og talsystemer, rækkefølgen, lister sorteres i, og håndteringen af personnavne og tiltaleformer. Den fjerde holder alt, der skal oversættes, ude af kildekoden, så den rette udgave kan indlæses til hver læser.

Brug begrebet i en beslutning

Brug »Hvad er internationalisering?« i en konkret beslutning. Vi kan knytte begrebet til det system, I bygger eller køber. Software, der læses naturligt på hvert marked, I sælger på — også datoer, valutaer og den måde, navne sorteres på.

Se vores ydelser Kontakt os →

Undersøg antagelserne bag ordet

Brug forklaringen og det tilknyttede arbejde til at formulere spørgsmål til jeres projekt. Vi kan undersøge antagelserne sammen, før et værktøj vælges. Har bygget et engelsk/dansk i18n-system — 12,027 beskeder pr. sprog, på engelsk og dansk

Har bygget et engelsk/dansk i18n-system Kontakt os →

Hvorfor det betyder noget

W3C's argument handler om timing. At ombygge et produkt, der er bygget op om ét sprog og én kultur, er langt sværere og mere tidskrævende end at designe det til et globalt publikum fra begyndelsen, og W3C sammenligner arbejdet med år 2000-problemet, hvor tocifrede årstalsfelter, der gik ud fra, at hvert år begyndte med 19, skulle bygges om. Kommer internationaliseringen sent, ender den ofte som besværlig og dyr ombygning. Kommer den tidligt, er den en håndfuld vaner, der næsten intet koster den dag, de indføres.

Prisen for en manglende vane vokser med hvert sprog. En sætning, der stykkes sammen af tre stumper i koden, er én fejl i et produkt på dansk og fem fejl i et produkt med fem sprog, og rettelsen er hver gang ikke en bedre oversættelse, men en ændring i koden, der skal gennemgås og udgives som enhver anden. Det samme gælder en dato, der formateres i hånden, en liste, der sorteres ved at sammenligne bytes, og en test, der finder en knap på dens danske tekst.

Vanerne betaler sig også, når intet nogensinde bliver oversat, og det påpeger W3C udtrykkeligt. Et produkt, der kun sælges på dansk, modtager stadig navne med bogstaver, dansk ikke har, adresser skrevet efter et andet lands skik og datoer fra systemer, der sætter måneden før dagen. Tekst, der gemmes som Unicode, og data, der beholder en dato som en dato, indtil den vises, klarer alt det uden særtilfælde.

Sådan ser det ud i praksis

Den hurtigste prøve på, om et produkt er internationaliseret, er en sætning med et tal i. Versionen, der ikke er, bygger sætningen i koden:

```
const label = count + (count === 1 ? " dag" : " dage");
```

Det virker for danske heltal og kan ikke oversættes til ukrainsk. Ukrainsk har tre former for hele tal og en fjerde for brøker: день efter 1, 21 og 101, дні efter 2, 3, 4 og 22, днів efter 0, 5 til 20 og 25, og дня efter et decimaltal. Uanset hvilket ord en oversætter skriver i den anden gren, er det forkert for de fleste tal, fordi 21 tager samme form som 1, og 2 tager en anden form end 5.

Når produktet er internationaliseret, bliver sætningen til en nøgle, og hvert sprogs tabel leverer alle de former, sproget har brug for. I Hugo ligger oversættelsestabellerne i mappen `i18n`, én fil pr. sprog, opkaldt efter sprogkoden:

```
# i18n/da.toml
[days]
one = '{{ . }} dag'
other = '{{ . }} dage'
```

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

Skabelonen beder om nøglen og giver tallet med. Den indeholder ikke et ord på nogen af sprogene og ingen regel om flertal:

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

De danske sider viser `21 dage` og de ukrainske `21 день`; med 2 viser de `2 dage` og `2 дні`, og med 5 `5 dage` og `5 днів`. Navnene `one`, `few`, `many` og `other` er ikke Hugos opfindelse. Det er flertalskategorierne fra CLDR, Unicode-konsortiets samling af lokale konventioner, og Hugos dokumentation henviser til CLDR's oversigt over flertalsregler for at se, hvilke kategorier hvert sprog har. Et nyt sprog kommer til som en ny fil uden ændringer i skabelonerne.

Når produktet vokser, bliver tabellerne dets største samling tekst, og de kræver samme disciplin som koden. Et af dette websteds projekter, et engelsk/dansk i18n-system, rummer 12.027 beskeder pr. locale fordelt på 458 namespace-filer. Hver fil har et budget på 400 linjer, et linter-tjek sammenligner nøglerne på tværs af sprogene, så ingen nøgle findes på det ene sprog og mangler på det andet, og den danske side følger et fastlåst ordforråd, ned til det uformelle »du«, som værktøjerne håndhæver ved hver commit. Siden kalder budgettet den bærende del: uden en grænse pr. fil ville de 458 filer have været tolv.

Hvor det går galt

Sætninger limes sammen. En besked bygget som `"Slet " + n + " filer?"` når frem til oversætteren som to stumper uden nogen sætning omkring dem og i en rækkefølge, kun dansk og engelsk har brug for. Et andet sprog vil måske have tallet til sidst eller en bøjning, der afhænger af navneordet. W3C nævner omhu med sammensætning af strenge som en del af arbejdet. Gør hver sætning til én besked med navngivne pladsholdere, som oversætteren kan flytte rundt på; Hugos dokumentation viser, hvordan `{{ T "age" (dict "name" "Will" "count" 1) }}` fylder både et navn og et tal ind i én oversat sætning.

Koden læser ordene på skærmen. En test, der finder en knap på teksten »Gem«, eller en gren, der sammenligner en status med teksten »Betalt«, virker indtil den første oversættelse og fejler derefter uden en fejlmeddelelse. W3C regner det at undgå, at koden afhænger af brugerfladens tekster, blandt de første ting, internationalisering indebærer. Sammenlign nøgler, identifikatorer og faste værdier, og giv testene en stabil attribut at finde elementet på.

Én flertalsregel til alle sprog. `count === 1` er den engelske regel skrevet som kode. Den holder for danske heltal, men er forkert for de fleste af de sprog, et produkt kan få. CLDR's specifikation definerer seks flertalskategorier, zero, one, two, few, many og other, hvoraf kun other er obligatorisk, og giver hvert sprog

sine egne regler for, hvilken der gælder: dansk bruger to, ukrainsk fire. Giv tallet med til beskeden, og lad oversættelsestabellen rumme alle de kategorier, sproget bruger.

En manglende oversættelse fejler lydløst. Hugos `T` falder tilbage på standardsproget, når en nøgle mangler i det aktuelle sprog, og returnerer en tom streng, når standardsproget heller ikke har den. En side kan altså gå i luften med et hul eller med en engelsk vending midt i en dansk sætning. Byg med `--printI18nWarnings` for at få listet alle manglende oversættelser og fallbacks, slå `enableMissingTranslationPlaceholders` til i et testmiljø, så hullerne ses på siden, og sammenlign alle sprogs nøgler i den kontinuerlige integration.

Datoer og tal formateres i hånden. En skabelon, der skriver `MM/DD/YYYY`, eller et tal med komma for hvert tredje ciffer, har kodet ét land ind. Samme dag er `10/27/26` på amerikansk engelsk, `27/10/2026` på britisk engelsk, `27.10.2026` på dansk og `2026-10-27` på svensk, og det, der på dansk skrives 12.027,5, er `12,027.5` på engelsk og har mellemrum som tusindtalsseparator på norsk, svensk og ukrainsk. CLDR's specifikation anbefaler at gemme og udveksle sprogneutrale data og først formatere dem til læseren. Hold datoer og beløb som rene værdier til sidste trin, og formatér dem der med platformens sprogbevidste funktioner. I Hugo viser `time.Format ":date_full"` samme dato som `Wednesday, November 3, 2021` på en engelsk side og `Mittwoch, 3. November 2021` på en tysk.

Teksten gemmes ikke som Unicode. Et produkt, der skriver »blåbærsyltetøj« i UTF-8 og læser det tilbage som Windows-1252, viser »blÃ¥bÃ¦rsyltetÃ¸j«, og et produkt, der gemmer tekst i en ældre kodning beregnet til ét sprog, kan slet ikke rumme et ukrainsk navn. At tage Unicode i brug er det første punkt på W3C's liste. Brug én Unicode-kodning, normalt UTF-8, hele vejen fra databasen til browseren, og angiv den ved hver grænse, hvor bytes bliver til tekst.

Læs mere

- Localization vs. Internationalization — W3C's arbejdsdefinitioner af begge begreber, de fire slags arbejde, internationalisering omfatter, og hvorfor det hører hjemme i begyndelsen.
- Hugo: Multilingual mode — oversættelsestabeller, lokaliserede datoer og valutaer, og hvordan Hugo melder en manglende oversættelse.

- Hugo: lang.Translate — opslagsrækkefølgen på tværs af sprog, flertalskategorier i oversættelsestabeller og beskeder med navngivne pladsholdere.
- UTS #35, Part 1: Core — CLDR's specifikation: locale-identifikatorer, nedarvning mellem locales og rådet om at holde data sprogneutrale, indtil de vises.
- UTS #35, Part 3: Numbers — talsymboler og talmønstre pr. locale samt flertalskategorierne og reglerne for at vælge mellem dem.

Ordliste

Vælg ud fra jeres faktiske vilkår

Beskriv arbejdsgangen, de eksisterende grænseflader og de brugere, der har brug for ændringen. Vi kan definere en testbar leverance og en overdragelse, der gør teamet i stand til at vedligeholde løsningen.

Se, hvad det koster Kontakt os →

Gør begreberne praktiske

Spørg, hvordan »Hvad er internationalisering?« gælder i jeres organisation. Et konkret eksempel fra jeres nuværende system er nok til at begynde samtalen.

Se vores ydelser

welcome@engineer.company +45 50 10 61 84 København, Danmark

Tag denne side med dig

Visitkort

Lavet med kærlighed af D.

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

Copyright © 2025 – nu · Engineer ApS · Alle rettigheder forbeholdes.

Udskrevet fra https://engineer.company/da/glossary/internationalization/