Engineer

Internasjonalisering

å utforme programvare slik at den lett kan lokaliseres

Hva det er

Internasjonalisering er det tekniske arbeidet som gjør at ett produkt kan brukes på mange språk og i mange regioner uten at koden må endres for hvert av dem. W3C definerer det som utforming og utvikling av et produkt, en applikasjon eller et dokument slik at det lett kan lokaliseres for målgrupper som skiller seg fra hverandre i kultur, region eller språk. På engelsk forkortes ordet i18n, fordi det står 18 bokstaver mellom den første i-en og den siste n-en, og forkortelsen brukes også i norske kodebaser. Noen kaller den samme ideen globalisering.

W3C oppsummerer arbeidsdelingen i én setning: designere og utviklere internasjonaliserer produkter, slik at oversettere og lokaliseringsspesialister enkelt kan lokalisere dem. Internasjonaliseringen gjøres én gang, i koden og malene. Lokaliseringen gjøres på nytt for hvert marked, i data som koden leser. Et produkt kan være fullt internasjonalisert og likevel bare finnes på ett språk.

Enheten begge deler arbeider med, er en locale. CLDRs spesifikasjon definerer en locale som en identifikator for et sett brukerpreferanser som gjerne deles over store deler av verden: et språk, ofte avgrenset til en region, slik at `en-GB` og `en-US` deler språk, men skriver en dato forskjellig. Identifikatoren velger på én gang en oversettelse og et sett konvensjoner.

W3C deler arbeidet inn i fire typer. Den første fjerner hindringer: produktet behandler tekst som Unicode eller håndterer en eldre tegnkoding bevisst der det er nødvendig, det bygger ikke setninger ved å skjøte sammen biter, og koden er aldri avhengig av ordlyden brukeren ser. Den andre legger til støtte som først et senere språk får bruk for, for eksempel oppmerking for tekst som blander skrift fra høyre mot venstre og fra venstre mot høyre, en måte å angi hvilket språk et avsnitt er skrevet på, og stilregler for loddrett tekst. Den tredje lar koden bruke lokale konvensjoner, som regel fra eksisterende biblioteker heller enn fra tabeller

teamet skriver selv: dato- og tidsformater, kalendere, tallformater og tallsystemer, rekkefølgen lister sorteres i, og håndteringen av personnavn og tiltaleformer. Den fjerde holder alt som skal oversettes, utenfor kildekoden, slik at riktig versjon kan lastes inn for hver leser.

Bruk begrepet i en beslutning

Bruk «Hva er internasjonalisering?» i en konkret beslutning. Vi kan knytte begrepet til systemet dere bygger eller kjøper. Programvare som leses naturlig i alle markeder du selger i, også datoene, valutaene og måten navn sorteres på.

Se tjenestene våre Ta kontakt →

Undersøk antakelsene bak ordet

Bruk forklaringen og det tilknyttede arbeidet til å formulere spørsmål for prosjektet. Vi kan undersøke antakelsene sammen før et verktøy velges. Bygde et engelsk/dansk i18n-system — 12,027 meldinger per språkversjon, på engelsk og dansk

Bygde et engelsk/dansk i18n-system Ta kontakt →

Hvorfor det er viktig

W3Cs argument handler om timing. Å bygge om et produkt som er laget rundt ett språk og én kultur, er langt vanskeligere og mer tidkrevende enn å utforme det for et globalt publikum fra starten av, og W3C sammenligner arbeidet med år 2000-problemet, da tosifrede årstallsfelt som gikk ut fra at hvert år begynte med 19, måtte bygges om. Kommer internasjonaliseringen sent, ender den ofte som tungvint og dyr ombygging. Kommer den tidlig, er den en håndfull vaner som nesten ikke koster noe den dagen de innføres.

Prisen for en manglende vane vokser med hvert språk. En setning som skjøtes sammen av tre biter i koden, er én feil i et produkt på norsk og fem feil i et produkt med fem språk, og rettelsen er hver gang ikke en bedre oversettelse, men en endring i koden som må gjennomgås og rulles ut som enhver annen. Det samme gjelder en dato som formateres for hånd, en liste som sorteres ved å sammenligne bytes, og en test som finner en knapp på den norske teksten.

Vanene lønner seg også når ingenting noen gang blir oversatt, og det påpeker W3C uttrykkelig. Et produkt som bare selges på norsk, får likevel navn med bokstaver norsk ikke har, adresser skrevet etter et annet lands skikk og datoer fra systemer som setter måneden foran dagen. Tekst som lagres som Unicode, og data som beholder en dato som en dato til den vises, klarer alt dette uten særtilfeller.

Slik ser det ut i praksis

Den raskeste prøven på om et produkt er internasjonalisert, er en setning med et tall i. Versjonen som ikke er det, bygger setningen i koden:

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

Det fungerer for norske heltall og kan ikke oversettes til ukrainsk. Ukrainsk har tre former for hele tall og en fjerde for brøker: день etter 1, 21 og 101, дні etter 2, 3, 4 og 22, днів etter 0, 5 til 20 og 25, og дня etter et desimaltall. Uansett hvilket ord en oversetter skriver i den andre grenen, blir det feil for de fleste tall, fordi 21 tar samme form som 1, og 2 tar en annen form enn 5.

Når produktet er internasjonalisert, blir setningen en nøkkel, og hvert språks tabell leverer alle formene språket trenger. I Hugo ligger oversettelsestabellene i katalogen `i18n`, én fil per språk, oppkalt etter språkkoden:

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

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

Malen ber om nøkkelen og sender med tallet. Den inneholder ikke ett ord på noen av språkene og ingen regel om flertall:

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

De norske sidene viser `21 dager` og de ukrainske `21 день`; med 2 viser de `2 dager` og `2 дні`, og med 5 `5 dager` og `5 днів`. Navnene `one`, `few`, `many` og `other` er ikke Hugos oppfinnelse. Det er flertallskategoriene fra CLDR, Unicode-konsortiets samling av lokale konvensjoner, og Hugos dokumentasjon viser til CLDRs oversikt over flertallsregler for å se hvilke kategorier hvert språk har. Et nytt språk kommer til som en ny fil, uten endringer i malene.

Når produktet vokser, blir tabellene den største tekstmassen i det, og de krever samme disiplin som koden. Et av dette nettstedets prosjekter, et engelsk/dansk i18n-system, har 12 027 meldinger per språkversjon fordelt på 458 navneromsfiler. Hver fil har et budsjett på 400 linjer, en lintersjekk sammenligner nøklene på tvers av språkene, så ingen nøkkel finnes på ett språk og mangler på det andre, og den danske siden følger et låst vokabular, helt ned til det uformelle «du», som verktøyene håndhever ved hver commit. Siden kaller budsjettet den bærende delen: uten en grense per fil ville de 458 filene ha vært tolv.

Hvor det går galt

Setninger limes sammen. En melding bygd som `"Slett " + n + " filer?"` kommer fram til oversetteren som to biter uten noen setning rundt og i en rekkefølge bare norsk og engelsk trenger. Et annet språk vil kanskje ha tallet til slutt eller en bøyning som avhenger av substantivet. W3C nevner varsomhet med sammenskjøting av strenger som en del av arbeidet. Gjør hver setning til én melding med navngitte plassholdere som oversetteren kan flytte på; Hugos dokumentasjon viser hvordan `{{ T "age" (dict "name" "Will" "count" 1) }}` fyller både et navn og et tall inn i én oversatt setning.

Koden leser ordene på skjermen. En test som finner en knapp på teksten «Lagre», eller en gren som sammenligner en status med teksten «Betalt», fungerer fram til den første oversettelsen og feiler deretter uten noen feilmelding. W3C regner det å unngå at koden avhenger av tekstene i brukergrensesnittet, blant de første tingene internasjonalisering innebærer. Sammenlign nøkler, identifikatorer og faste verdier, og gi testene et stabilt attributt å finne elementet på.

Én flertallsregel for alle språk. `count === 1` er den engelske regelen skrevet som kode. Den holder for norske heltall, men er feil for de fleste språkene et produkt kan få. CLDRs spesifikasjon definerer seks flertallskategorier, zero, one,

two, few, many og other, der bare other er obligatorisk, og gir hvert språk egne regler for hvilken som gjelder: norsk bruker to, ukrainsk fire. Send tallet med til meldingen, og la oversettelsestabellen romme alle kategoriene språket bruker.

En manglende oversettelse feiler lydløst. Hugos `T` faller tilbake på standardspråket når en nøkkel mangler i det gjeldende språket, og returnerer en tom streng når standardspråket heller ikke har den. En side kan altså gå ut med et hull eller med en engelsk frase midt i en norsk setning. Bygg med `--printI18nWarnings` for å få listet alle manglende oversettelser og reserveverdier, slå på `enableMissingTranslationPlaceholders` i et testmiljø så hullene synes på siden, og sammenlign alle språkenes nøkler i den kontinuerlige integrasjonen.

Datoer og tall formateres for hånd. En mal som skriver `MM/DD/YYYY`, eller et tall med komma for hvert tredje siffer, har kodet inn ett land. Samme dag er `10/27/26` på amerikansk engelsk, `27/10/2026` på britisk engelsk, `27.10.2026` på norsk og dansk og `2026-10-27` på svensk, og et tall som skrives 12 027,5 på norsk, er `12,027.5` på engelsk og `12.027,5` på dansk. CLDRs spesifikasjon anbefaler å lagre og utveksle språknøytrale data og først formatere dem for leseren. Hold datoer og beløp som rene verdier til siste trinn, og formater dem der med plattformens språkbevisste funksjoner. 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 lagres ikke som Unicode. Et produkt som skriver «blåbærsyltetøy» i UTF-8 og leser det tilbake som Windows-1252, viser «blÃ¥bÃ¦rsyltetÃ¸y», og et produkt som lagrer tekst i en eldre koding beregnet på ett språk, kan ikke romme et ukrainsk navn i det hele tatt. Å ta i bruk Unicode er det første punktet på W3Cs liste. Bruk én Unicode-koding, vanligvis UTF-8, hele veien fra databasen til nettleseren, og angi den ved hver grense der bytes blir til tekst.

Les mer

- Localization vs. Internationalization — W3Cs arbeidsdefinisjoner av begge begrepene, de fire typene arbeid som internasjonalisering omfatter, og hvorfor det hører hjemme i begynnelsen.
- Hugo: Multilingual mode — oversettelsestabeller, lokaliserte datoer og valutaer, og hvordan Hugo melder fra om en manglende oversettelse.
- Hugo: lang.Translate — oppslagsrekkefølgen på tvers av språk, flertallskategorier i oversettelsestabeller og meldinger med navngitte plassholdere.

- UTS #35, Part 1: Core — CLDRs spesifikasjon: identifikatorer for locales, arven mellom dem og rådet om å holde data språknøytrale til de vises.
- UTS #35, Part 3: Numbers — tallsymboler og tallmønstre per locale samt flertallskategoriene og reglene for å velge mellom dem.

Ordliste

Velg ut fra deres faktiske rammer

Beskriv arbeidsflyten, eksisterende grensesnitt og brukerne som trenger endringen. Vi kan definere en testbar leveranse og en overlevering som gjør teamet i stand til å vedlikeholde løsningen.

Se hva det koster Ta kontakt →

Gjør begrepene praktiske

Spør hvordan «Hva er internasjonalisering?» gjelder i organisasjonen deres. Et konkret eksempel fra dagens system er nok til å starte samtalen.

Se tjenestene våre

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

Ta med deg denne siden

Visittkort

Laget med kjærlighet av D.

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

Opphavsrett © 2025 – nå · Engineer ApS · Alle rettigheter forbeholdt.

Skrevet ut fra https://engineer.company/nb/glossary/internationalization/