Engineer

Internationalisering

att utforma programvara så att den lätt kan lokaliseras

Vad det är

Internationalisering är det tekniska arbete som gör att en och samma produkt kan användas på många språk och i många regioner utan att koden behöver ändras för vart och ett av dem. W3C definierar det som utformning och utveckling av en produkt, en applikation eller ett dokument så att det lätt kan lokaliseras för målgrupper som skiljer sig åt i kultur, region eller språk. På engelska förkortas ordet i18n, eftersom det står 18 bokstäver mellan det första i:et och det sista n:et, och förkortningen används även i svenska kodbaser. En del kallar samma idé globalisering.

W3C sammanfattar arbetsfördelningen i en mening: de som designar och utvecklar produkter internationaliserar dem, så att översättare och lokaliseringsspecialister enkelt kan lokalisera dem. Internationaliseringen görs en gång, i koden och mallarna. Lokaliseringen görs på nytt för varje marknad, i data som koden läser. En produkt kan vara helt internationaliserad och ändå bara finnas på ett språk.

Enheten som båda arbetar med är en locale. CLDR:s specifikation definierar en locale som en identifierare för en uppsättning användarpreferenser som ofta delas över stora delar av världen: ett språk, ofta avgränsat till en region, så att `en-GB` och `en-US` delar språk men skriver ett datum olika. Identifieraren väljer på en gång en översättning och en uppsättning konventioner.

W3C delar in arbetet i fyra delar. Den första undanröjer hinder: produkten hanterar text som Unicode eller hanterar en äldre teckenkodning medvetet där det krävs, den bygger inte meningar genom att foga ihop fragment, och koden är aldrig beroende av den ordalydelse som användaren ser. Den andra lägger till stöd som först ett senare språk behöver, till exempel uppmärkning för text som blandar skrift från höger till vänster och från vänster till höger, ett sätt att ange

vilket språk ett stycke är skrivet på och stilregler för lodrät text. Den tredje låter koden använda lokala konventioner, oftast från befintliga bibliotek snarare än från tabeller som teamet skriver själv: datum- och tidsformat, kalendrar, talformat och talsystem, den ordning som listor sorteras i samt hanteringen av personnamn och tilltalsformer. Den fjärde håller allt som ska översättas utanför källkoden, så att rätt version kan laddas för varje läsare.

Använd begreppet i ett beslut

Använd ”Vad är internationalisering?” i ett konkret beslut. Vi kan knyta begreppet till systemet ni bygger eller köper. Mjukvara som känns naturlig på varje marknad du säljer på, inklusive datum, valutor och hur namn sorteras.

Se våra tjänster Hör av er →

Granska antagandena bakom ordet

Använd förklaringen och det länkade arbetet för att formulera frågor till projektet. Vi kan undersöka antagandena tillsammans innan ett verktyg väljs. Byggde ett i18n-system för engelska och danska — 12,027 meddelanden per språkversion, på engelska och danska

Byggde ett i18n-system för engelska och danska Hör av er →

Varför det spelar roll

W3C:s argument handlar om tidpunkt. Att bygga om en produkt som har byggts kring ett språk och en kultur är mycket svårare och mer tidskrävande än att utforma den för en global publik från början, och W3C jämför arbetet med millenniebuggen, då tvåsiffriga årtalsfält som utgick från att varje år började med 19 fick byggas om. Kommer internationaliseringen sent blir den ofta en klumpig och dyr ombyggnad. Kommer den tidigt är den en handfull vanor som nästan inte kostar något den dag de införs.

Priset för en saknad vana växer med varje språk. En mening som fogas ihop av tre fragment i koden är ett fel i en produkt på svenska och fem fel i en produkt med fem språk, och rättningen är varje gång inte en bättre översättning utan en ändring i koden som ska granskas och släppas som vilken annan som helst. Detsamma gäller ett datum som formateras för hand, en lista som sorteras genom att jämföra byte för byte och ett test som hittar en knapp på dess svenska text.

Vanorna lönar sig även när ingenting någonsin översätts, och det påpekar W3C uttryckligen. En produkt som bara säljs på svenska får ändå namn med bokstäver som svenskan inte har, adresser skrivna enligt ett annat lands sed och datum från system som sätter månaden före dagen. Text som lagras som Unicode, och data som behåller ett datum som ett datum tills det visas, klarar allt detta utan specialfall.

Så ser det ut i praktiken

Det snabbaste sättet att se om en produkt är internationaliserad är en mening med ett tal i. Versionen som inte är det bygger meningen i koden:

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

Det fungerar för svenska heltal och går inte att översätta till ukrainska. Ukrainskan har tre former för heltal och en fjärde för bråktal: день efter 1, 21 och 101, дні efter 2, 3, 4 och 22, днів efter 0, 5 till 20 och 25, och дня efter ett decimaltal. Vilket ord en översättare än skriver i den andra grenen blir det fel för de flesta tal, eftersom 21 tar samma form som 1 och 2 tar en annan form än 5.

När produkten är internationaliserad blir meningen en nyckel, och varje språks tabell innehåller alla de former som språket behöver. I Hugo ligger översättningstabellerna i katalogen `i18n`, en fil per språk, namngiven efter språkkoden:

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

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

Mallen ber om nyckeln och skickar med talet. Den innehåller inte ett ord på något av språken och ingen regel om plural:

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

De svenska sidorna visar `21 dagar` och de ukrainska `21 день` ; med 2 visar de `2 dagar` och `2 дні` , och med 5 `5 dagar` och `5 днів` . Namnen `one` , `few` , `many` och `other` är inte Hugos påfund. Det är pluralkategorierna från CLDR, Unicode-konsortiets samling av lokala konventioner, och Hugos dokumentation hänvisar till CLDR:s översikt över pluralregler för att se vilka kategorier varje språk har. Ett nytt språk tillkommer som en ny fil, utan ändringar i mallarna.

När produkten växer blir tabellerna dess största textmassa, och de kräver samma disciplin som koden. Ett av den här webbplatsens projekt, ett i18n-system för engelska och danska, har 12 027 meddelanden per språkversion i 458 namnrymdsfiler. Varje fil har en budget på 400 rader, en kontroll i lintern jämför nycklarna mellan språkversionerna så att ingen nyckel finns på ett språk och saknas på det andra, och den danska sidan följer ett låst ordförråd, ända ner till det informella ”du”, som verktygen kontrollerar vid varje commit. Sidan kallar budgeten den bärande delen: utan en gräns per fil hade de 458 filerna varit tolv.

Var det går fel

Meningar limmas ihop. Ett meddelande som byggs som `"Radera " + n + " filer?"` når översättaren som två fragment utan någon mening runt omkring och i en ordning som bara svenska och engelska behöver. Ett annat språk kanske vill ha talet sist eller en verbform som beror på substantivet. W3C nämner försiktighet med sammanfogning av strängar som en del av arbetet. Gör varje mening till ett enda meddelande med namngivna platshållare som översättaren kan flytta; Hugos dokumentation visar hur `{{ T "age" (dict "name" "Will" "count" 1) }}` fyller i både ett namn och ett tal i en och samma översatta mening.

Koden läser orden på skärmen. Ett test som hittar en knapp på texten ”Spara”, eller en gren som jämför en status med texten ”Betald”, fungerar fram till den första översättningen och går sedan sönder utan något felmeddelande. W3C räknar att undvika att koden beror på texterna i användargränssnittet till det första som internationalisering innebär. Jämför nycklar, identifierare och fasta värden, och ge testerna ett stabilt attribut att hitta elementet med.

En pluralregel för alla språk. `count === 1` är den engelska regeln skriven som kod. Den håller för svenska heltal men är fel för de flesta språk som en produkt kan få. CLDR:s specifikation definierar sex pluralkategorier, zero, one, two, few, many och other, av vilka bara other är obligatorisk, och ger varje språk egna

regler för vilken som gäller: svenskan använder två, ukrainskan fyra. Skicka med talet till meddelandet och låt översättningstabellen rymma alla kategorier som språket använder.

En saknad översättning misslyckas tyst. Hugos `T` faller tillbaka på standardspråket när en nyckel saknas i det aktuella språket och returnerar en tom sträng när standardspråket inte heller har den. En sida kan alltså publiceras med ett hål eller med en engelsk fras mitt i en svensk mening. Bygg med `--printI18nWarnings` för att få en lista över alla saknade översättningar och reservvärden, slå på `enableMissingTranslationPlaceholders` i en testmiljö så att hålen syns på sidan, och jämför alla språks nycklar i den kontinuerliga integrationen.

Datum och tal formateras för hand. En mall som skriver `MM/DD/YYYY`, eller ett tal med kommatecken efter var tredje siffra, har kodat in ett enda land. Samma dag är `10/27/26` på amerikansk engelska, `27/10/2026` på brittisk engelska, `27.10.2026` på danska och norska och `2026-10-27` på svenska, och ett tal som skrivs 12 027,5 på svenska är `12,027.5` på engelska och `12.027,5` på danska. CLDR:s specifikation rekommenderar att data lagras och utbyts i språkneutral form och formateras först för läsaren. Håll datum och belopp som rena värden till sista steget och formatera dem där med plattformens språkmedvetna funktioner. I Hugo visar `time.Format ":date_full"` samma datum som `Wednesday, November 3, 2021` på en engelsk sida och `Mittwoch, 3. November 2021` på en tysk.

Texten lagras inte som Unicode. En produkt som skriver ”räksmörgås” i UTF-8 och läser tillbaka ordet som Windows-1252 visar ”rÃ¤ksmÃ¶rgÃ¥s”, och en produkt som lagrar text i en äldre kodning avsedd för ett enda språk kan inte rymma ett ukrainskt namn över huvud taget. Att ta Unicode i bruk är den första punkten på W3C:s lista. Använd en enda Unicode-kodning, normalt UTF-8, hela vägen från databasen till webbläsaren, och ange den vid varje gräns där byte blir text.

Läs mer

- Localization vs. Internationalization — W3C:s arbetsdefinitioner av båda begreppen, de fyra typer av arbete som internationalisering omfattar och varför det hör hemma i början.
- Hugo: Multilingual mode — översättningstabeller, lokaliserade datum och valutor, och hur Hugo rapporterar en saknad översättning.
- Hugo: lang.Translate — uppslagsordningen mellan språk, pluralkategorier i översättningstabeller och meddelanden med namngivna platshållare.

- UTS #35, Part 1: Core — CLDR:s specifikation: identifierare för locales, arvet mellan dem och rådet att hålla data i språkneutral form fram till visningen.
- UTS #35, Part 3: Numbers — talsymboler och talmönster per locale samt pluralkategorierna och reglerna för att välja mellan dem.

Ordlista

Välj utifrån era faktiska förutsättningar

Beskriv arbetsflödet, befintliga gränssnitt och användarna som behöver förändringen. Vi kan definiera en testbar leverans och en överlämning som gör att teamet kan underhålla lösningen.

Se vad det kostar Hör av er →

Gör begreppen praktiska

Fråga hur ”Vad är internationalisering?” gäller i er organisation. Ett konkret exempel från ert nuvarande system räcker för att börja samtalet.

Se våra tjänster

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

Ta med dig den här sidan

Visitkort

Omsorgsfullt skapad med kärlek av D.

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

Upphovsrätt © 2025 – nu · Engineer ApS · Alla rättigheter förbehållna.

Utskrivet från https://engineer.company/sv/glossary/internationalization/