Engineer

Lokalisierung

Software an Sprache und Konventionen eines Markts anpassen

Was es ist

Lokalisierung ist die Arbeit, ein Produkt an einen bestimmten Markt anzupassen: an dessen Sprache und damit an die Konventionen, die seine Leser für Datumsangaben, Zahlen, Preise, Namen, Adressen und die Reihenfolge einer Liste erwarten. Das W3C definiert sie als Anpassung eines Produkts, einer Anwendung oder des Inhalts eines Dokuments an einen Zielmarkt, damit es dessen Sprache und Kultur entspricht und alle weiteren Anforderungen dieses Marktes erfüllt. Diesen Markt nennt es Locale. Das englische Wort wird sowohl localization als auch localisation geschrieben und zu l10n verkürzt, nach den 10 Buchstaben zwischen seinem ersten l und seinem letzten n.

Das W3C ergänzt, dass Lokalisierung oft als Synonym für die Übersetzung von Oberfläche und Dokumentation verstanden wird, häufig aber eine wesentlich komplexere Aufgabe ist. Seine Liste möglicher Anpassungen umfasst Zahlen-, Datums- und Zeitformate; Währungen; Tastaturbelegung; Kollation und Sortierung; Symbole, Icons und Farben; Texte und Grafiken, die in einer Kultur missverstanden oder als unsensibel empfunden werden können; rechtliche Anforderungen; die Formate von Adressen, Postleitzahlen und Telefonnummern sowie die Reihenfolge von Namen; und die Anforderungen jeder Schrift an die Lesbarkeit. Unterscheiden sich die Geschäftspraktiken, etwa in der vom W3C als Beispiel genannten Buchhaltung, kann die Anpassung bedeuten, Logik, visuelle Gestaltung oder Präsentation des Produkts neu zu durchdenken.

Lokalisierung setzt Internationalisierung voraus und folgt auf sie. Internationalisierung geschieht einmal im Code, damit darin nichts auf einen einzelnen Markt festgelegt ist. Lokalisierung geschieht für jede Locale in Daten, die der Code liest: den übersetzten Meldungen und den Konventionen, die ein gespeichertes Datum oder einen Betrag in den vom Leser erwarteten Text

verwandeln. Ein gut internationalisiertes Produkt kann eine Locale ohne Codeänderung hinzubekommen. Ein nicht ausreichend internationalisiertes Produkt muss für jeden neuen Markt erneut geändert werden.

Die meisten Konventionsdaten muss das Team nicht selbst schreiben. Das CLDR-Projekt des Unicode Consortium sammelt sie für über 100 Sprachen und lässt sie von Fachleuten und Muttersprachlern der jeweiligen Sprache prüfen. Sein eigenes Beispiel ist die Dateiliste auf einem Telefon: Datums- und Zahlenformate, die Einheiten der Dateigrößen und die alphabetische Reihenfolge unterscheiden sich je nach Sprache und werden gewöhnlich von CLDR geliefert. Das Projekt erklärt, dass seine Daten in allen modernen Betriebssystemen und Browsern sowie in Sprachen wie Java, C#, Swift und JavaScript stecken. Ein Produkt erhält die meisten Formate von der Plattform, sofern der Code sie dort abfragt, statt sie selbst zu schreiben.

Das Konzept für eine Entscheidung nutzen

Nutzen Sie „Was ist Lokalisierung?“ für eine konkrete Entscheidung. Wir können helfen, das Konzept mit dem System in Verbindung zu bringen, das Sie entwickeln oder einkaufen. Software, die in jedem Ihrer Absatzmärkte natürlich klingt — einschließlich Datumsangaben, Währungen und der Sortierung von Namen.

Unsere Leistungen entdecken Kontakt aufnehmen →

Die Annahmen hinter dem Begriff prüfen

Nutzen Sie die Erklärung und die verlinkten Arbeiten, um Fragen für Ihr Projekt zu formulieren. Wir können die Annahmen gemeinsam mit Ihnen prüfen, bevor ein Werkzeug ausgewählt wird. Englisch-dänisches i18n-System aufgebaut — 12,027 Nachrichten je Sprachversion, auf Englisch und Dänisch

Englisch-dänisches i18n-System aufgebaut Kontakt aufnehmen →

Warum es wichtig ist

Ein übersetztes, aber nicht lokalisiertes Produkt wirkt fremd. Jede Beschriftung ist dänisch, doch beim Datum steht weiterhin der Monat zuerst, der Preis trägt noch ein Dollarzeichen, und die Ortsliste sortiert Aarhus weiterhin unter A ein. Leser bemerken jede dieser Eigenheiten sofort. Jede sagt ihnen, dass das Produkt für jemand anderen gemacht wurde.

Manche Abweichungen verändern, was der Leser versteht. Die Zeichenfolge 10/11/2026 bedeutet für einen amerikanischen Leser den 11. Oktober und für einen britischen den 10. November. Beides sind gültige Daten, also schlägt nichts fehl, und einer der beiden Leser handelt am falschen Tag. Die Zeichenfolge 1,234 bedeutet im Englischen eintausendzweihundertvierunddreißig und im Dänischen

eins Komma zwei drei vier: ein Faktor von tausend bei einer Menge oder einem Preis. Eine falsche Übersetzung sieht falsch aus. Eine falsche Konvention sieht richtig aus und wird falsch gelesen.

Die Liste des W3C reicht auch über den Bildschirm hinaus. Rechtliche Anforderungen und Geschäftspraktiken unterscheiden sich zwischen Märkten. Eine Locale kann deshalb andere Bedingungen, andere Rechnungsangaben oder einen anderen Ablauf beim Kauf brauchen. Das sind geschäftliche Entscheidungen. Sie gehören in die Planung einer Locale, bevor die erste Meldung übersetzt wird.

Die Kosten fallen pro Locale an und enden nicht. Jede neue Meldung muss in jede Sprache übersetzt werden. Jede Veröffentlichung muss vor der Auslieferung in jeder Locale gelesen werden, und für jeden Markt sind die eigenen Konventionen zu prüfen. Ein Team, das eine Sprache hinzufügt, nimmt diese Arbeit in jede spätere Veröffentlichung auf. Deshalb ist die Entscheidung für einen Markt zugleich eine Entscheidung über den Veröffentlichungsprozess.

Wie es in der Praxis aussieht

Die Plattform erledigt den Großteil der Arbeit, wenn der Code ihr einen Wert und eine Locale übergibt. In JavaScript nehmen die `Intl` -Formatierer beides entgegen und geben Text zurück:

```
const when = new Date("2026-10-27T12:00:00Z");
const money = { style: "currency", currency: "DKK" };
for (const locale of ["en-US", "en-GB", "da", "nb", "sv", "uk"]) {
  const date = new Intl.DateTimeFormat(locale, { dateStyle: "short" });
  const price = new Intl.NumberFormat(locale, money);
  console.log(locale, date.format(when), price.format(800));
}
```

Unter Node.js 25 mit Daten aus CLDR 47 gibt das Programm ein Datum und einen Preis von 800 dänischen Kronen auf sechs Arten aus:

```
en-US 10/27/26 DKK 800.00
en-GB 27/10/2026 DKK 800.00
da 27.10.2026 800,00 kr.
```

```
nb 27.10.2026 800,00 DKK
sv 2026-10-27 800,00 Dkr
uk 27.10.26 800,00 DKK
```

Der Code hat nichts davon ausgewählt. Der dänische Leser sieht kr., der schwedische Dkr, weil kr in Schweden schwedische Kronen bedeutet; norwegische und ukrainische Leser sehen den Währungscode. Die Reihenfolge von Tag und Monat, die Trennzeichen, das Dezimalkomma und die Position der Währung stammen sämtlich aus den Locale-Daten. In der tatsächlichen Ausgabe steht zwischen Betrag und Währung ein geschütztes Leerzeichen, U+00A0, das beide in einer Zeile hält.

Sortieren funktioniert genauso. Eine Liste mit sechs dänischen Ortsnamen, für drei Locales sortiert:

```
const towns = ["Aabenraa", "Ærø", "Åkirkeby", "Odense", "Østerbro", "Vejle"];
for (const locale of ["en", "da", "sv"]) {
  const order = new Intl.Collator(locale).compare;
  console.log(locale, towns.toSorted(order).join(", "));
}
```

```
en Aabenraa, Ærø, Åkirkeby, Odense, Østerbro, Vejle
da Odense, Vejle, Ærø, Østerbro, Aabenraa, Åkirkeby
sv Aabenraa, Odense, Vejle, Åkirkeby, Ærø, Østerbro
```

Das Englische behandelt æ, ø und å als Varianten von a und o. Das Dänische stellt Æ, Ø und Å hinter Z und ordnet die alte Schreibweise Aa wie Å ein. Aabenraa rückt deshalb vom ersten auf den vorletzten Platz. Das Schwedische stellt sein eigenes Å, Ä und Ö hinter Z und ordnet Æ bei Ä und Ø bei Ö ein. Jede Reihenfolge ist für ihre Leser richtig. Ein dänischer Leser, der Aabenraa nahe dem Ende einer englisch sortierten Liste sucht, hat Grund zu glauben, dass es fehlt.

Eine Hugo-Website hat dieselben Werkzeuge in ihren Templates. Hugos Dokumentation übergibt einen Betrag an `lang.FormatCurrency`:

```
{{ 512.5032 | lang.FormatCurrency 2 "USD" }}
```

Eine englische Seite zeigt ihn als $512.50, eine französische als 512,50 $US und eine deutsche als 512,50 $ an. `lang.FormatNumber` und `lang.FormatPercent` tun dasselbe für gewöhnliche Zahlen und Prozentangaben.

Lokalisierung ist Bestandteil einer Leistung dieser Website. Die Leistung Internationalisierung und Lokalisierung beschreibt das Ziel als Software, die sich in jedem Absatzmarkt eines Kunden natürlich liest, einschließlich Datumsangaben, Währungen und der Sortierung von Namen. Zu ihren Ergebnissen gehört eine sprachübergreifende Inhaltsprüfung. Sie schlägt fehl, wenn eine Übersetzung eine Zahl aus dem Englischen auslässt oder eine Sprache eine für sie unübliche Schreibweise verwendet. Sie fand zwei dänische Beschreibungen ohne eine Kennzahl und sechzehn ukrainische Textstellen mit englischer Zitierweise.

Wo es schiefgeht

Lokalisierung wird als Übersetzung budgetiert. Der Plan für einen neuen Markt deckt nur die Meldungsdateien ab. Das Produkt startet deshalb in der neuen Sprache mit Datumsangaben, Preisen, Adressen und Sortierreihenfolge des alten Marktes, und Kunden finden die Lücken. Planen Sie jede Locale anhand der vollständigen W3C-Liste, entscheiden Sie, welche Punkte auf diesen Markt zutreffen, und lassen Sie einen Muttersprachler das gebaute Produkt statt der Meldungsdateien prüfen.

Eine Sprache wird mit einer Locale gleichgesetzt. `en` sagt nicht, ob der 27. Oktober `10/27/26` oder `27/10/2026` heißt. Die Ausgabe oben zeigt, dass sich die beiden englischen Locales sowohl in der Reihenfolge als auch beim Jahr unterscheiden. Speichert ein Produkt nur die Sprache, muss es den Rest erraten und wendet die Konventionen eines Marktes auf alle anderen an. Halten Sie für jeden Nutzer ein vollständiges Locale-Kennzeichen wie `en-GB` oder `da-DK` fest, übernehmen Sie es aus einer ausdrücklichen Einstellung, wenn es sie gibt, und übergeben Sie es jedem Formatierer.

Die Währung wird aus der Locale erraten. Die CLDR-Spezifikation ist eindeutig: Ein Betrag sollte nie ohne bekannten Währungscode zwischen Systemen weitergereicht werden. Sonst kann dieselbe Zahl 3,5 „von einem Nutzer als $3.50 und von einem anderen als €3.50 interpretiert“ werden. Sie ergänzt, dass die Währung eines Betrags unabhängig von der Sprache des Lesers ist und nur schlecht entworfene Software den Code weglässt und aus der Locale ableitet. In Skandinavien geht das Raten unbemerkt schief, weil Dänisch, Norwegisch und

Schwedisch die eigene Währung jeweils mit kr abkürzen. Ein als 800 gespeicherter Preis bedeutet dann für einen Leser 800 dänische und für den nächsten 800 norwegische Kronen. Speichern Sie jeden Betrag mit seinem ISO-4217-Code und formatieren Sie ihn für den Leser. Ein dänischer Leser sieht einen norwegischen Preis dann als 800,00 NOK.

Listen werden nach Codepunkt sortiert. JavaScripts standardmäßiges `sort` vergleicht UTF-16-Codeeinheiten. Dadurch stehen Є und І vor dem übrigen ukrainischen Alphabet und Ґ hinter Я, weil ihre Codepunkte außerhalb des Bereichs von A bis я liegen. So sortiert ergibt `["Ялта", "Ґанок", "Гора", "Єнот", "Іван", "Дуб"]` die Folge Єнот, Іван, Гора, Дуб, Ялта, Ґанок. Mit `new Intl.Collator("uk").compare` entsteht Гора, Ґанок, Дуб, Єнот, Іван, Ялта, die Reihenfolge des ukrainischen Alphabets. Sortieren Sie angezeigten Text mit einem Kollator für die Locale des Lesers, im Code oder über die Kollation der Datenbank. Behalten Sie die Codepunktreihenfolge für Schlüssel bei, die kein Leser sieht.

Tests vergleichen von Hand eingegebenen formatierten Text. Ein Test, der `800,00 kr.` mit einem gewöhnlichen Leerzeichen erwartet, scheitert am Formatierer, der dort U+00A0 einfügt. Dasselbe geschützte Leerzeichen trennt im Norwegischen, Schwedischen und Ukrainischen die Tausender. Locale-Daten werden außerdem zwischen CLDR-Versionen überarbeitet. Eine Zeichenfolge, die vor einem Plattformupgrade passte, kann danach abweichen. Erzeugen Sie den erwarteten Text mit demselben Formatierer und derselben Locale wie im Code oder prüfen Sie den gespeicherten Wert und die übergebene Locale statt der dargestellten Zeichenfolge.

Das Layout ist für Englisch bemessen. Ein Button, auf den Save passt, muss auch Зберегти mit doppelt so vielen Buchstaben aufnehmen. Ein Menü für Settings muss auch dem dänischen Indstillinger Platz bieten. Feste Breiten schneiden das längere Wort ab oder schieben es in eine zweite Zeile, und oft sieht ein Leser im jeweiligen Markt das zuerst. Lassen Sie Container mit ihrem Text wachsen, vermeiden Sie feste Breiten um übersetzte Wörter und prüfen Sie jede Locale auf dem schmalsten vom Produkt unterstützten Bildschirm.

Weiterführende Lektüre

- Lokalisierung im Vergleich zur Internationalisierung — die Arbeitsdefinition des W3C, seine Liste möglicher Anpassungen für eine Locale und der Zusammenhang zwischen Lokalisierung und Internationalisierung.

- Hugo: Mehrsprachiger Modus — lokalisierte Datumsangaben, Währungen, Zahlen und Prozentangaben in Hugo-Templates, neben Übersetzungstabellen und sprachbezogenen URLs.
- UTS #35, Teil 3: Zahlen — CLDRs Zahlen- und Währungsformate einschließlich der Regel, dass ein Betrag immer mit seinem Währungscode weitergegeben wird.
- Unicode-CLDR-Projekt — das Projekt, das Locale-Daten für über 100 Sprachen sammelt und Betriebssysteme, Browser und Programmiersprachen damit versorgt.

Glossar

Für Ihre tatsächlichen Rahmenbedingungen auswählen

Beschreiben Sie den Arbeitsablauf, die vorhandenen Schnittstellen und die Nutzer, die die Veränderung benötigen. Wir können eine prüfbare Lieferung definieren und die Übergabe so gestalten, dass Ihr Team die gelieferten Bestandteile warten kann.

Preise ansehen Kontakt aufnehmen →

Die Begriffe praktisch nutzbar machen

Fragen Sie, wie „Was ist Lokalisierung?“ auf Ihre Organisation zutrifft. Ein konkretes Beispiel aus Ihrem bestehenden System reicht aus, um das Gespräch zu beginnen.

Unsere Leistungen entdecken

welcome@engineer.company +45 50 10 61 84 Kopenhagen, Dänemark

Nehmen Sie diese Seite mit

Kontaktkarte

Mit Sorgfalt und Liebe gestaltet von D.

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

Copyright © 2025 – heute · Engineer ApS · Alle Rechte vorbehalten.

Gedruckt von https://engineer.company/de/glossary/localization/