motionpoint

The Challenges and Solutions of Localizing Dynamic Content

Discover how industry-leading technologies make it easy to localize dynamic content.

Introduction

Today's B2B and B2C websites lean heavily on personalizing the user experience by intelligently serving *relevant content*.

For example, when a user arrives on a site, a custom product module may serve **specific recommended items** from a product catalog based on the user's **geolocation** and **previous behavior** on the site.

These applications are often powered by some form of JavaScript—increasingly leveraging JSON, XML and frameworks such as ReactJS or AngularJS. In the browser, during the user's experience, the content appears to come from a single, static URL.

This kind of dynamic content can pose challenges for most digital content translation vendors.

The Standard Approach

A typical proxy-based translation approach relies on its technology to detect *new URLs*, which trigger the need to translated content to appear on a localized page.

A URL change can be initiated through the appearance of location-based information, cookies or actions such as user profile logins. **But single-page applications and other technologies that deliver dynamic content won't initiate a URL change.** This means most translation technologies cannot detect that new content needs to be served to a user.

Other technologies like CMS connectors struggle to detect translatable content served using complex JavaScript frameworks such as AngularJS or ReactJS. As a result, the content never gets discovered or translated at all.

The on-page result for users is **a clumsy and alienating mixed-language experience**, since applications are serving *translated content* alongside *untranslated content*.

A Modern Solution

Superior translation solutions, however, have found ways to solve this problem. Cutting-edge, proprietary proxy translation technologies are different than others in the industry.

These advanced solutions are designed to detect the *actual calls to the server* used by applications that deliver dynamic content on-site. Their crawlers note when a server call happens and follow the call directly back to the servers to find the new content—such as a set of customized product recommendations.

The superior technology then automatically routes that content to the proxy server for translation, no matter how many times those server calls occur.

Since server calls happen *no matter where the content originates*, this approach ensures that the content gets detected from any and every source that populates content on the site. Further, the technologies of advanced, leading solutions are designed to work with content delivered in any language or format.

Handling JavaScript

World-class proxy translation solutions work by:

- ✓ Examining the code base that powers a website
- ✓ Finding all of the content-related and translatable text within that code
- ✓ And breaking it into smaller chunks called *segments* that are then easily translated

For simple text- or HTML-based sites in which the text is fully visible in the code, this is a straightforward process.

But for certain technologies and file formats such as JavaScript, this content doesn't actually appear in the code for the page itself.

Industry-leading solutions teach their translation proxies to parse this content differently, so they can identify any text that requires translation and ignore the rest.

JavaScript is often written with nodes or functions in the code that can be "activated" through specific algorithms to help the translation technology find and isolate translatable content.

For example, in this code...

```
//function to confirm item was added to cart
function cartConfirm()
{
  //alert function triggered
   alert("Item added to cart!");
}
//function called
cartConfirm();
```

...the *alert* function can be detected and activated through the algorithm. A great vendor's technology can identify the content that will be served to a user and requires translation.

Other functions like *variables* can also be identified and activated so the corresponding values can be captured and stored as translatable segments. A similar process can also be used for inline JavaScript functions.

Localizing Data Formats: JSON and XML

A similar approach also works for capturing translatable content within JSON and XML files.

During the parsing process, world-class solutions use proprietary algorithms to identify and queue only the JSON key/value pairs that have translatable content within them.

In the following example, the technology would note that the product name, description and sizes are translatable. It would automatically segment the content and route it to linguists for translation.

```
{
"product_ID": 33w0lg6d,
"product_name": "Winter Pants",
"product_description": "Shield you bottom half from the elements with Burton snow pants, featuring insulated and waterproof options to stay warm and dry all winter",
“product_Sizes”:{
“product_LG”: “Large”,
“product_MD”: “Medium”,
“product_SM”: “Small”
}
  }
```

Anything deemed “non-translatable” passes directly through the proxy technology untouched. This can include data such as serial numbers or a product ID number.

Similarly, superior solutions can be configured to parse required XML files and will only translate the node values that are

needed to serve localized content to site visitors. This can be seen in the highlighted sections of the following example.

Any node values that are not needed or intended for translation—such as the SKU, brand name, price and rating in our example here—would be ignored.

```
<Product>
        <sku>ABJ-252</sku>
        <name>Nike Track Jacket</name>
        <brand>Nike</brand>
        <description> This is a really great
        jacket. It's available in several
        colors and will keep you warm.
        </description>
        <price>49.99</price>
        <category>Apparel</category>
        <color>Red</color>
        <rating>5</rating>
</Product>
```

Integrations and APIs

Instead of relying on static product libraries, many complex sites often rely on Product Information Management technologies (PIMs) that handle immense amounts of inventory, product specifications and logistics details, and deliver them to the website via integrations.

The best vendors offer APIs that connect their proxy translation technologies directly to a PIM. Using that API as a pathway, a company could:

- ✓ Send the translatable content from a PIM database
- ✓ Automatically queue this content for translation to the vendor's translation server
- ✓ Translate this content and store it in *translation memory*—a database of all a customer's translations—for later use.

Because the API integration with the PIM is completely independent of the site structure and other applications, content translated through this process can be easily and seamlessly deployed elsewhere on the site, wherever it might appear. This sidesteps the high costs of repetitive, duplicate content translation.

About MotionPoint

MotionPoint solves the operational complexity and cost of website localization. Unlike all other approaches, our technology and turn-key solution are built specifically for this purpose.

We translate, deploy, and operate multilingual websites, optimizing the customer experience across all channels.

motionpoint

MotionPoint Corporation

info@motionpoint.com

www.motionpoint.com