Translated Strings and Foreign Language Support in JavaScript Web Apps

Ken Tabor
Twitter: @KenTabor
Shared

bit.ly/KenOscon13
Translating What?

Any and all blocks of text
Translating What?

Date/time formatters
Translating Where?

Single page app
Translating Myth #1

We don’t need translated text, we’re only shipping to North America
Translating Myth #2

No need - it’s just an internal tool
Translating Myth #3

We’re a start-up ... let’s just submit
Translating OMG #1

Just before code freeze marketing asks:
Translating OMG #1

Just before code freeze marketing asks:
Hey, will you send over all the text please? We want to do a polish pass for voice and message before ship.
Translating OMG #1

Just before code freeze marketing asks:
Hey, will you send over all the text please? We want to do a polish pass for voice and message before ship.
Don’t worry we’ll return it by end of day.
Translating OMG #2

CoderSean and PM build create() feature

“Connection error, notifying the admin.”
Translating OMG #2

CoderBurin and BA build add() feature

“Network problem. Notifying the admin.”
Why Work Smarter?

Programmers: Don’t Repeat Yourself
Business: Include More Customers
Marketing: Craft Consistent Message
Solution?

String Tables
Typical String Table

```javascript
TS.tstring = {
    appTitle: 'TString.JS',
    programListing: 'Program Listing',
    english: 'English',
    spanish: 'Spanish',
    german: 'German',
    title: 'Title',
    date: 'Date',
    synopsis: 'Synopsis',
    programDate: 'MMM Do YYYY, h:mm a'
};
```
Architectural Rule #1

Never hard-code text strings
Architectural Rule #2

String-tables must be easily accessed throughout the application.
Architectural Rule #3

Language string-tables must be demand-loadable resources
Is this Difficult?

Couldn’t Google my way out of this task
JavaScript is popular lacks formal support
We’re all rushed after all
Who Am I to Do This?

I’m a front-end product engineer
Working at Sabre Holdings
Building TripCase
Our Dev Strategy

Mobile web browser (webkit)
Our Dev Strategy

iOS and Android native apps (PhoneGap)
Our Dev Strategy

Coders get smart and stay smart
<table>
<thead>
<tr>
<th>Core Languages</th>
</tr>
</thead>
<tbody>
<tr>
<td>App Structure</td>
</tr>
<tr>
<td>Presentation Layer</td>
</tr>
<tr>
<td>Utility Libs</td>
</tr>
<tr>
<td>Code Analysis/Build</td>
</tr>
<tr>
<td>Native Wrapper</td>
</tr>
<tr>
<td><strong>JavaScript + CSS + HTML</strong></td>
</tr>
<tr>
<td>----------------------------</td>
</tr>
<tr>
<td><strong>Backbone + Underscore + RequireJS</strong></td>
</tr>
<tr>
<td><strong>jQuery + SASS + Compass</strong></td>
</tr>
<tr>
<td><strong>Handlebars + Moment + Modernizr</strong></td>
</tr>
<tr>
<td><strong>Jasmine + jsHint + Plato + Grunt</strong></td>
</tr>
<tr>
<td><strong>PhoneGap</strong></td>
</tr>
</tbody>
</table>
Demo

Show the people some running code
String Table

```javascript
TS.tstring = {
    appTitle: 'TString.JS',
    programListing: 'Program Listing',
    english: 'English',
    spanish: 'Spanish',
    german: 'German',
    title: 'Title',
    date: 'Date',
    synopsis: 'Synopsis',
    programDate: 'MMM Do YYYY, h:mm a'
};
```
// encapsulate all code modules into a global namespace

var TS = TS || {};
TS.model = TS.model || {};
TS.view = TS.view || {};
TS.collection = TS.collection || {};
TS.app = null;

TS.tstring = null;
Template User

```
TS.view.Application = Backbone.View.extend({

    template:
      '<h1>' +
      '<h1>TSStringJS</h1>' +
      '<h2>Example of translated strings and foreign language</h2>' +
      '<select id="lang-list">' +
      '<option value="english">English</option>' +
      '<option value="spanish">Spanish</option>' +
      '<option value="french">French</option>' +
      '<option value="italian">Italian</option>' +
    '</select>' +
    '<button id="lang-load">Select</button>'
  });
```
Template User

```
TS.view.Application = Backbone.View.extend({

template:
'\n<h1><%= TS.tstring.appTitle %></h1>'
+'\n<h2><%= TS.tstring.appSubTitle %></h2>'
+'\n<select id="lang-list">
  \n  \n  <option value="english"><%= TS.tstring.english %></option>
  \n  \n  <option value="spanish"><%= TS.tstring.spanish %></option>
  \n  \n  <option value="french"><%= TS.tstring.french %></option>
  \n  \n  <option value="italian"><%= TS.tstring.italian %></option>
  \n  \n</select>
'+'\n<button id="lang-load"><%= TS.tstring.select %></button>
\n```
```javascript
function onSelectChoice(id) {
    var userModel;

    userModel = this.collection.get(id);
    alert(TS.tstring.selectedProgram + ': ' +
          userModel.getDisplayTitle());
}
```
function LanguageLoad(lang) {
    var langFn;

    if (lang === 'english') langFn = 'en-US.js';
    else if (lang === 'spanish') langFn = 'es.js';
    else if (lang === 'klingon') langFn = 'tlh.js';

    $.ajax({
        url: 'code/strings/' + langFn,
        dataType: 'script',
        success: function(data, textStatus) {
            TS.app.onRender();
        }
    });
}
Making the Case

Combine duplicating strings
Stop nearly duplicated strings
Easier word-smithing
Setup for translation services
Abstract away date/time formatting
Handlebars User

Write a “helper” function
Wraps the global resource
Param indexes into the string table
Handlebars Helper

```javascript
Handlebars.registerHelper('tstring', function(key) {
    var tstring = TS.tstring[key];

    if (tstring === undefined)
        console.log('Missing string[' + key + ']');

    return tstring;
});

<h1>{{tstring "appTitle"}}</h1>
<h2>{{tstring "appSubTitle"}}</h2>
```
RequireJS

RequireJS has “i18n” plugin
Auto-loads based on navigator.language
At the cost of user control
Native App

All string files are packaged together

```javascript
TS.tstring.en = {...};
TS.tstring.fr = {...};
TS.tstring.es = {...};
TS.tstring.tlh = {...};

TS.tstrings = TS.tstring.es;
```
Test entire app in English
Native speaker confirms translations
Wait on bug reports
Testing

Automated testing: Jasmine and Selenium

Manual debug: English with “SP-” prefix
The “Ken Bug”

Don’t accidentally ship your debug files
Think About It

Start from the start
Retrofitting costed a 50hr week
Plus 20hr regression testing
Think About It

Server-side error conditions happen
Map return codes rather than text
Keep Thinking

Pluralization functions
Keep Thinking

Coordinate with plugins & libraries
Keep Thinking

Designers must brain-sweat composition
Questions?
Thank-You

bit.ly/KenOscon13

Ken Tabor (@KenTabor)