ISO Developer's Toolkit - mdTranslator in Rails

2015 CDI Workshop

May 11, 2015

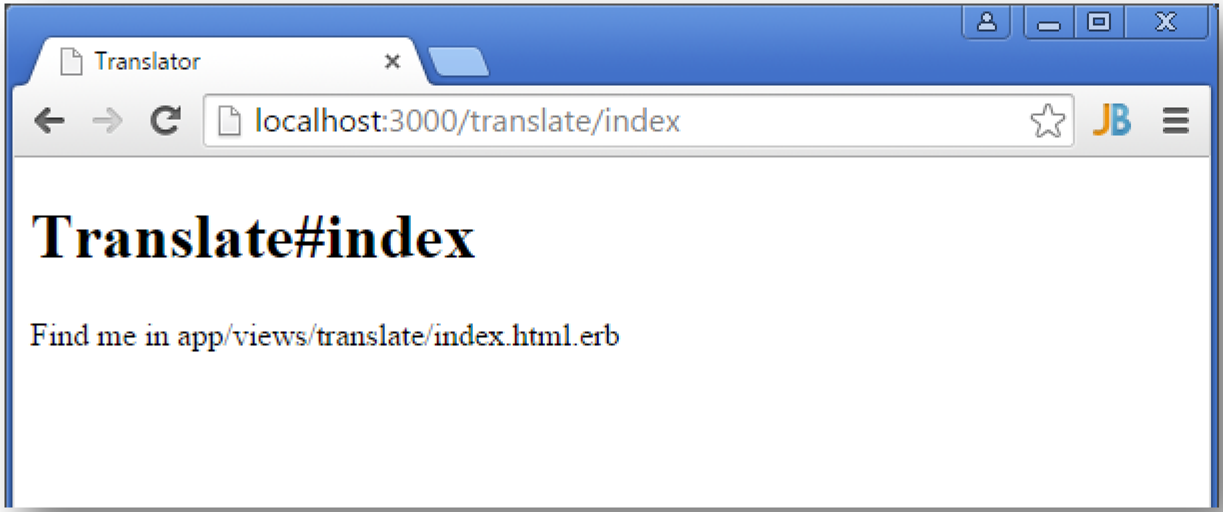
Stan Smith, USGS

mdTranslator on Rails 1

1. From the command line, navigate to the directory to hold your new Rails application. This will become the parent directory of the new Rails application.
2. Generate the new rails application.
   ```
   > rails new translator
   ```
3. A new directory named ‘translator’ will be created and filled with the new Rails application modules.
4. Test the new ‘translator’ rails application.
   - Change the directory to the Rails application just created
     ```
     > cd translator
     ```
   - Start the Rails server.
     ```
     > rails server
     ```
   - From a browser type http://localhost:3000
   - You should see the ‘Welcome aboard’ page.

mdTranslator on Rails 2

5. Generate a home page using a Rails helper.
   > rails generate controller translate index
6. Test the new index page that Rails generated for you …
   http://localhost:3000/translate/index



Translate#index

Find me in app/views/translate/index.html.erb

or maybe not!

- Found trouble on some Windows installations – not sure of cause
- In file
  …/app/views/layouts/application.html.erb
  comment out the line calling the javascript_include_tag so it looks like this:

```
<head>
  <title>mdTranslator API</title>
  <%= stylesheet_link_tag    'application', media: 'all', 'data-turbolinks-track' => true %>
  <!-- javascript_include_tag 'application', 'data-turbolinks-track' => true -->
  <%= csrf_meta_tags %>
</head>
```

- This feature preserves head JavaScript from page to page as a means of saving load and compile time. Not need in this application anyway.

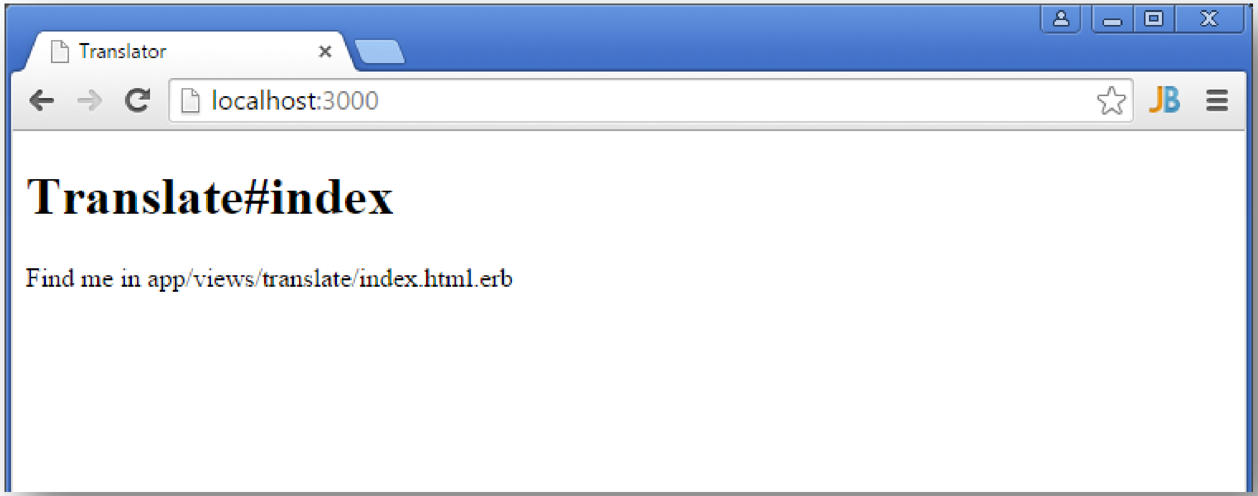
mdTranslator on Rails 3

7. Next set the ‘index’ page as the default page for the website. In the
   …/config/routes.rb
   file set the root to ‘translate#index’.

```
Rails.application.routes.draw do
   get 'translate/index'
   root 'translate#index'
```

mdTranslator on Rails 4

8. Now the index page will display at root rather than ‘Welcome aboard’.



Translate#index

Find me in app/views/translate/index.html.erb

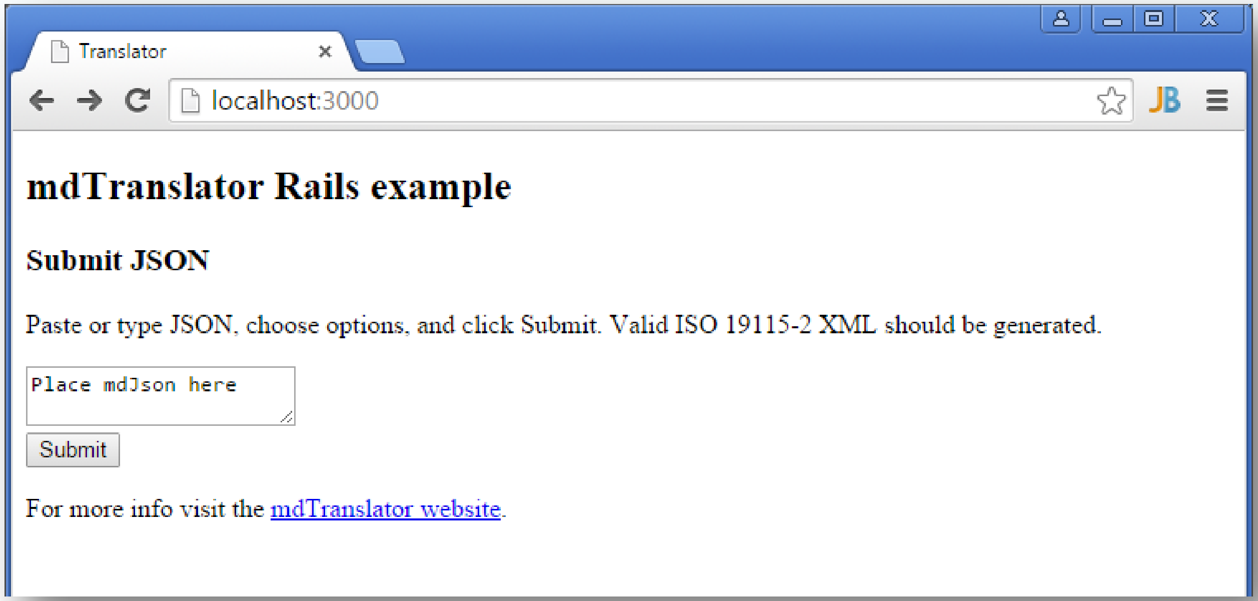
mdTranslator on Rails 5

9. Code a simple HTML form in the file *…/app/views/translate/index.html.erb* replacing all the Rails generated code.

```
<!--
# Alaska Data Integration working group - ADIwg

# REST endpoint controller for demonstration of mdTranslator
-->

<h2> mdTranslator Rails example</h2>

<div id="form_container">
   <%= form_tag('/translate') do %>
      <div>
         <h3>Submit JSON</h3>
         <p>
            Paste or type JSON, choose options, and click Submit.
            Valid ISO 19115-2 XML should be generated. <br>
         </p>

         <div>
            <textarea name="file">Place mdJson here</textarea>
         </div>
      </div>

      <div class="buttons">
         <input type="hidden" name="form_id" value="759352"/>
         <input id="saveForm" class="button_text" type="submit"
            name="submit" value="Submit"/>
      </div>

      <div>
         <p>
            For more info visit the
            <a href="http://www.adiwg.org/mdTranslator">mdTranslator
               website</a>.
         </p>
      </div>
   <% end %>
</div>
```

mdTranslator on Rails 6

10. The updated index page now looks like this …



Translator

localhost:3000

mdTranslator Rails example

Submit JSON

Paste or type JSON, choose options, and click Submit. Valid ISO 19115-2 XML should be generated.

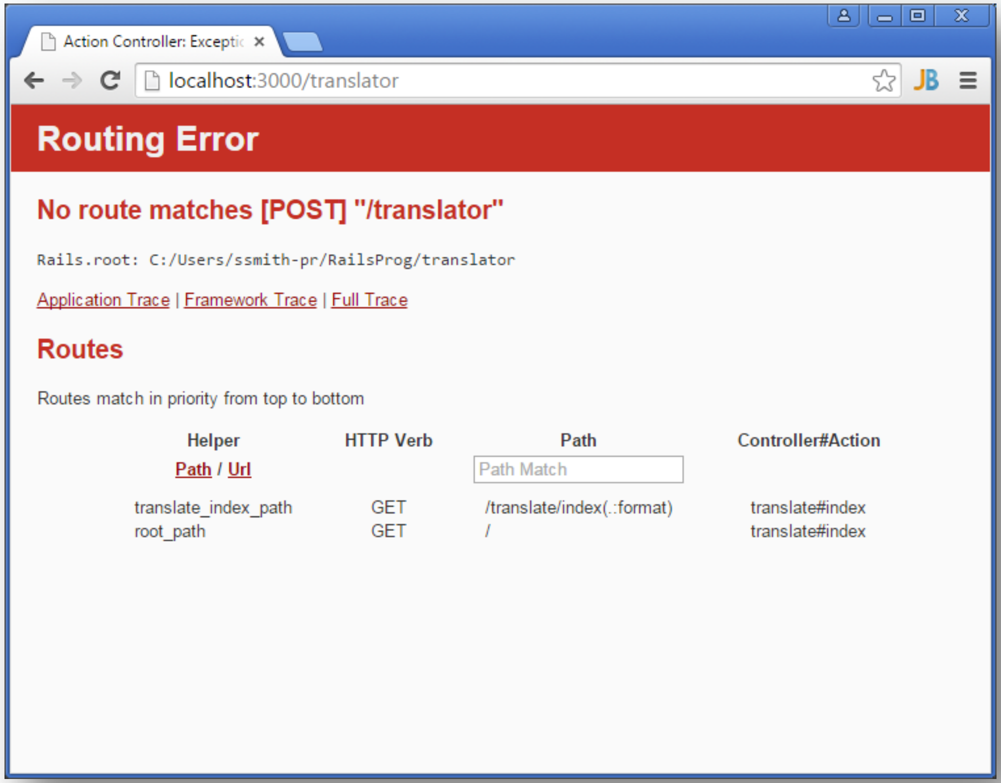
Place mdJson here

Submit

For more info visit the mdTranslator website.

mdTranslator on Rails 7

11. Click 'Submit' and Rails shows a 'Routing Error'. We have not handled the HTTP POST in our routing or controller.



Routing Error

No route matches [POST] "/translator"

```
Rails.root: C:/Users/ssmith-pr/RailsProg/translator
```

Application Trace | Framework Trace | Full Trace

Routes

Routes match in priority from top to bottom

Helper Path / Url	HTTP Verb	Path	Controller#Action
translate_index_path	GET	/translate/index(.:format)	translate#index
root_path	GET	/	translate#index

mdTranslator on Rails 8

12. Rails only generated code in the router and controller to process the HTTP GET verb. Add a ‘resources’ statement to *…/config/routes.rb* to have Rails automatically handle all HTTP verbs, including POST.

```
Rails.application.routes.draw do
    get 'translate/index'
    root 'translate#index'
    resources :translate
```

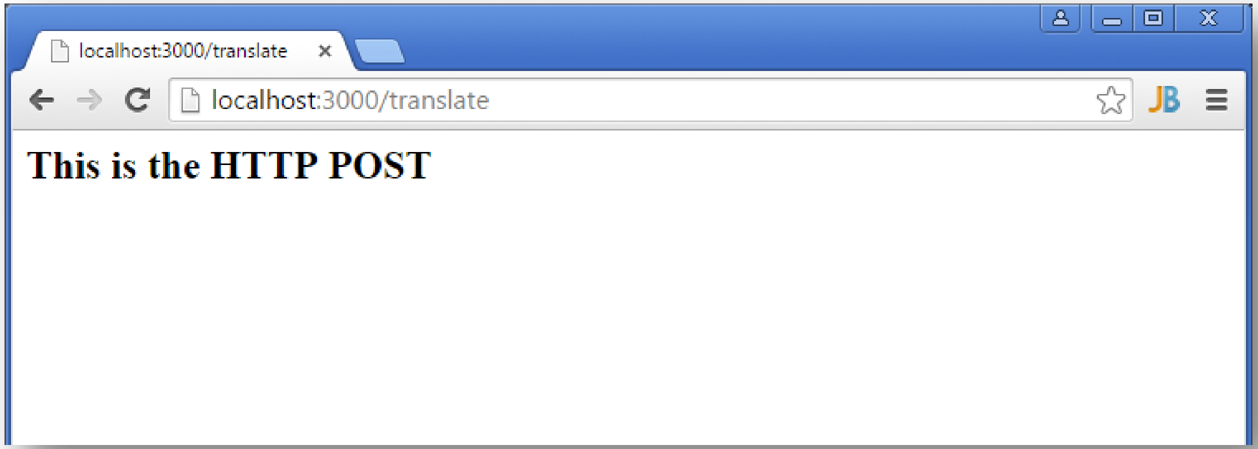
mdTranslator on Rails 9

13. Add code in the *…/app/controllers/translate_controller.rb* file intercept and process the HTTP POST request.

```
class TranslateController < ApplicationController

    # process GETs
    def index
    end

    # process POSTs
    def create
        render inline: '<h2>This is the HTTP POST</h2>'
    end

end
```

mdTranslator on Rails 10

14. Now the HTTP POST is properly intercepted. When the ‘Submit’ button is pressed we see…



This is the HTTP POST

mdTranslator on Rails 11

15. With the basic structure of the website complete connect the mdTranslator gem to the website. Add the adiwg-mdtranslator gem request to the .../Gemfile.

```
# Alaska Data Integration working group metadata translator
gem 'adiwg-mdtranslator', '~> 1.0'
```

16. Update the website's gems. From the command line type:
    > bundle update.
    The adiwg-mdtranslator gem and all its dependencies will be loaded to your Rails website.
17. Remember to restart the Rails server after adding new gems.

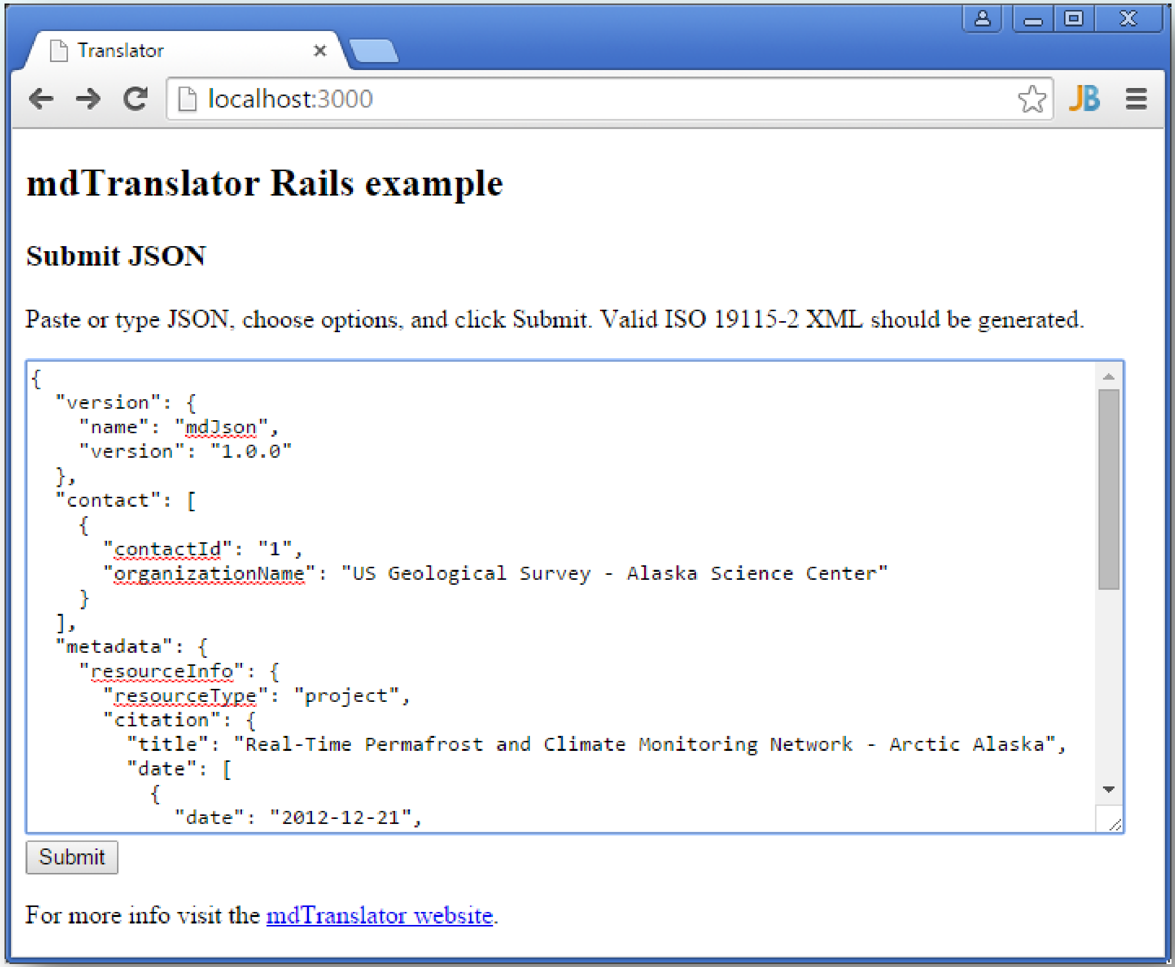
mdTranslator on Rails 12

18. Now we need to write a simple script to process the HTTP POST. In the ../app/controllers/translate_controller.rb
file replace the 'render inline:' statement we entered to test the connection and routing with something like this…

```
# process POSTs
def create

    # load file and parameter from POST
    fileObj = params[:file]

    # call the ADIwg metadata translator
    @mdReturn = ADIWG::Mdtranslator.translate(
        file: fileObj, reader: 'mdJson', validate: 'normal',
        writer: 'iso19115_2', showAllTags: false)

    render xml: @mdReturn[:writerOutput]

end
```

mdTranslator on Rails 13

18. After the server restarts, navigate to the website root and enter some valid mdJson.



mdTranslator Rails example

Submit JSON

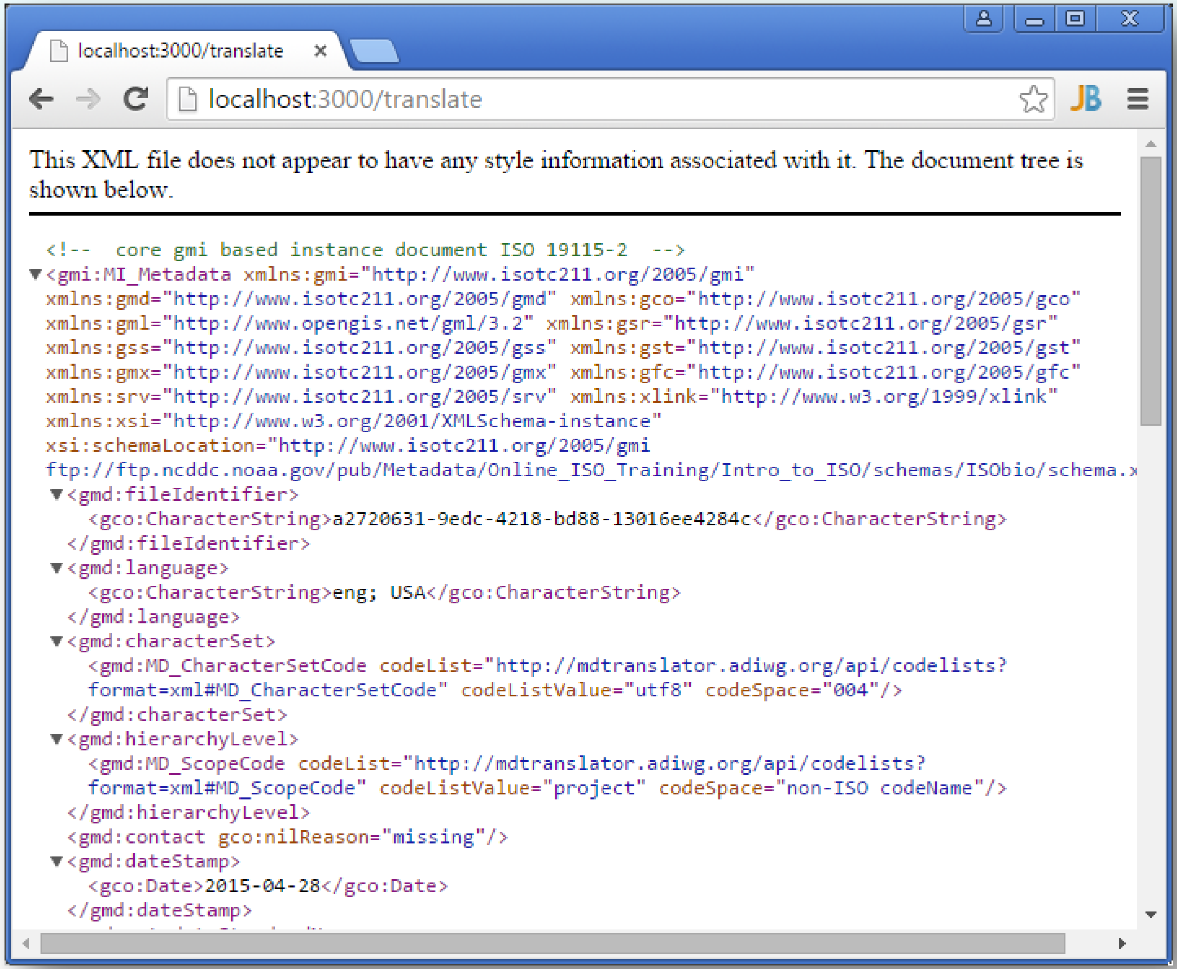
Paste or type JSON, choose options, and click Submit. Valid ISO 19115-2 XML should be generated.

```
{
  "version": {
    "name": "mdJson",
    "version": "1.0.0"
  },
  "contact": [
    {
      "contactId": "1",
      "organizationName": "US Geological Survey - Alaska Science Center"
    }
  ],
  "metadata": {
    "resourceInfo": {
      "resourceType": "project",
      "citation": {
        "title": "Real-Time Permafrost and Climate Monitoring Network - Arctic Alaska",
        "date": [
          {
            "date": "2012-12-21",
```

For more info visit the mdTranslator website.

mdTranslator on Rails 14

19. Click submit and you should see ISO 19115_2 returned from you locally hosted web service.



This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<!--  core gmi based instance document ISO 19115-2  -->
▼<gmi:MI_Metadata xmlns:gmi="http://www.isotc211.org/2005/gmi"
 xmlns:gmd="http://www.isotc211.org/2005/gmd" xmlns:gco="http://www.isotc211.org/2005/gco"
 xmlns:gml="http://www.opengis.net/gml/3.2" xmlns:gsr="http://www.isotc211.org/2005/gsr"
 xmlns:gss="http://www.isotc211.org/2005/gss" xmlns:gst="http://www.isotc211.org/2005/gst"
 xmlns:gmx="http://www.isotc211.org/2005/gmx" xmlns:gfc="http://www.isotc211.org/2005/gfc"
 xmlns:srv="http://www.isotc211.org/2005/srv" xmlns:xlink="http://www.w3.org/1999/xlink"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://www.isotc211.org/2005/gmi
 ftp://ftp.ncddc.noaa.gov/pub/Metadata/Online_ISO_Training/Intro_to_ISO/schemas/ISObio/schema.x
 ▼<gmd:fileIdentifier>
    <gco:CharacterString>a2720631-9edc-4218-bd88-13016ee4284c</gco:CharacterString>
  </gmd:fileIdentifier>
 ▼<gmd:language>
    <gco:CharacterString>eng; USA</gco:CharacterString>
  </gmd:language>
 ▼<gmd:characterSet>
    <gmd:MD_CharacterSetCode codeList="http://mdtranslator.adiwg.org/api/codelists?
    format=xml#MD_CharacterSetCode" codeListValue="utf8" codeSpace="004"/>
  </gmd:characterSet>
 ▼<gmd:hierarchyLevel>
    <gmd:MD_ScopeCode codeList="http://mdtranslator.adiwg.org/api/codelists?
    format=xml#MD_ScopeCode" codeListValue="project" codeSpace="non-ISO codeName"/>
  </gmd:hierarchyLevel>
  <gmd:contact gco:nilReason="missing"/>
 ▼<gmd:dateStamp>
    <gco:Date>2015-04-28</gco:Date>
  </gmd:dateStamp>
```

Alaska Data Integration working group

mdTranslator on Rails 15

- This ‘non-award winning’ website only demonstrates the simplicity of building a website that can interface with the ADIWG ISO Metadata Toolkit. A real website would need to be more robust checking for errors, handling all mdTranslator options, and handling all response types returned from the mdTranslator (XML. JSON, JSONp, text, plain). But it’s a start.

Questions?

Discussion?