Elementary!

Incorporating BlueMix, Node-RED and Watson in Domino applications

Our Amazing Sponsors

panagenda

IBM

RED PILL NOW

BCC

psc
It's all in the way we listen.

Ytria

cisco

riva
Riva CRM Integration

crossware mail signature

p* prominic.net

MartinScott

Phora Group

teamstudio

DOCOVA

FIRM hadsl www.hadsl.com

TeamCentric TECHNOLOGIES

workplaceON

Karl-Henry Martinsson

CEO, Demand Better Solutions, LLC

- In the IT industry for 29 years
- Full time programmer since 1991
- Worked with Notes and Domino since 1994
- Web development since 1995
- IBM Champion since 2014
- Worked as IT journalist for 5 years

IBM CHAMPION

Demand Better Solutions

Innovative software solutions to real business headaches

- Focus on user experience and interface
  - Make the software easy to use
  - To the user the UI is the application!
- Development for Notes and browser
  - New development, both Notes client and browser
  - Modernization of existing applications
  - Application migration from and to IBM Domino
- Take advantage of modern tools and technology
  - jQuery, Bootstrap, jQuery Mobile, BlueMix, etc

Agenda

- What is IBM BlueMix?
- What is Node-RED?
- Watson Services
  - Translation (English to Spanish)
  - Text-to-Speech
- Integrate into your own solutions
  - Notes client
  - Web application

What is IBM BlueMix?

- Platform as a Service (PaaS)
- Based on Cloud Foundry
  - Open source, multi cloud platform
- Running on SoftLayer infrastructure
- Supports multiple tools and services
  - Node-RED
  - Watson API

What is Node-RED?

- “Flow-based programming for IoT”
- Wire together services, APIs and hardware
- One-click deployment
- Built on Node.js
  - Event driven, non-blocking (asyncronous)
  - Light-weight runtime, can run on Raspberry Pi
  - Available as Docker container and on BlueMix

Node-RED Editor

- Browser based editor
- Click-and-drag of components (nodes)
- Input and output nodes (interface)
  - http, web sockets, email, twitter
- Function nodes
  - switch, split, join, change
- Javascript used to modify payload

Many Available Node

IBM Watson Services

- Conversation (bots and virtual agents)
- Document Conversion
- Natural Language Classifier
- Visual Recognition
- Language Translator
- Speech to Text / Text to Speech
- And many more…

Where Do We Start?

- Create a BlueMix account – free for 30 days
- Setup a Node-RED instance
- Add some Watson Services
  - Language Translator
  - Text to Speech

Docs | 1391425 Karl-Henry Martinsson's A... | US South : texasswede : texasswede

IBM Bluemix Apps | Catalog | Support | Manage

Search Items

All Apps (1)

Create App

Cloud Foundry Apps 512 MB/512 GB Used

NAME	ROUTE	MEMORY (...	INSTANCES	RUNNING	STATE	ACTIONS
TexasSwedeNodeRed	TexasSwedeNodeRed.mybluemix.net	512	1	1	Running	

All Services (3)

Create Service

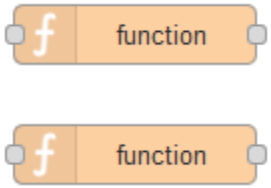
Services 3/2000 Used

NAME	SERVICE OFFERING	PLAN	ACTIONS
TexasSwedeNodeRed-cloudantNoSQLDB	Cloudant NoSQL DB	Lite	
Demand Better Text to Speech	Text to Speech	Standard	
Demand Better Translator	Language Translator	Lite	

Create Translation Flow

- Create and configure main nodes
  - http input
  - Watson Translate
  - http response

- Nodes to manipulate JSON msg object
  - Process incoming JSON for translation
  - Create JSONP for output



- Connect nodes

Node-RED

Deploy

filter nodes

split
join
csv
html
json
xml
yaml
tcp request
OpenWhisk
rbe

social

e mail
twitter
e mail
twitter

storage

mongodb
cloudant

Translation | Text to Speech | Flow 3

English text in
Process arguments
Translate using IBM Watson
Create JSONP to return
Spanish text out

info | debug

Node

Name	Translate using IBM Watson
Type	watson-translator
ID	ca13578d.cd2e18

Properties

The Watson Language Translator service enables you to translate text from one language to another and to add your own translation models.

Translation Mode.

The text to translate should be passed in on `msg.payload`.

The translated text will be returned on `msg.payload`.

The full response from the service will be returned on `msg.translation`

Source and destination language parameters can be configured through the editor panel or set dynamically using the language codes in the following properties, `msg.srclang` and `msg.destlang`. Set the values to either two or five character IETF language Codes, eg. `en` for English or `pt-BR` for Brazilian Portuguese Please see the documentation linked below for the currently supported source and destination languages translation permutations.

Customised Translation Mode.

Process Incoming JSON

```
// Set language to translate from (default is English)
if(msg.payload.from === null || msg.payload.from == '') {
    msg.srclang = "en-US";
} else {
    msg.srclang = msg.payload.from;}

// Get language to translate to (default is Spanish)
if(msg.payload.to === null || msg.payload.to == '') {
    msg.destlang = "es";
} else {
    msg.destlang = msg.payload.to;
}

// Set payload to text to translate
msg.payload = msg.payload.message;
// Pass object to the next node
return msg;
```

Create JSONP to Return

```
// Wrap payload in callback function
msg.payload = "translated({'translated':'" + msg.payload + "'})";

// Pass object to final node, to return to browser
return msg;
```

Call Flow From Web Page

HTML:

```
<input type="text" id="text" placeholder="Enter text to translate here">
<button id="btnTranslate" class="btn btn-primary">Translate</button>
<div id="translated"></div>
```

jQuery:

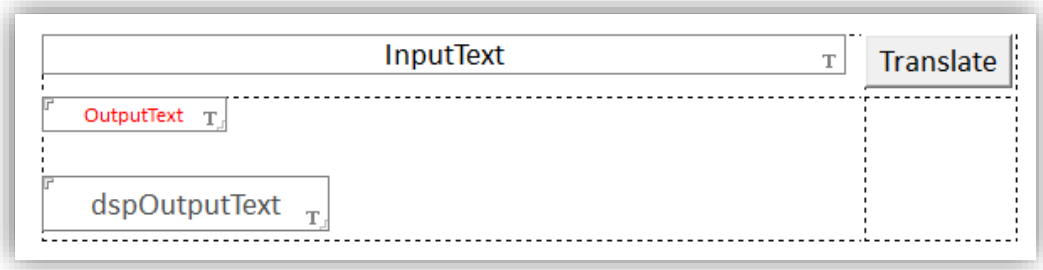
```
$("#btnTranslate").on("click", function (e) {
    e.preventDefault();
    // Store values to pass to flow in a JSON object
    var json = {
        "message":$("#text").val(),
        "from":"en",
        "to":"es"
    }
    // Call Node-RED flow using Ajax
    $.ajax({
        url: "https://texasswedenodered.mybluemix.net/translate",
        dataType: "jsonp",
        data: json
    });
});

// Call-back function for translation
function translated(data) {
    $("#translated").html(data.translated);
}
```

Demo

Call Flow From Notes

- Simple form
  - Two fields
  - Computed text
  - One button
    - Lotusscript to call URL and parse JSON



- Using my HTTP retrieval class
  - Using MSXML2.ServerXMLHTTP
  - Windows only
  - Code available on my blog

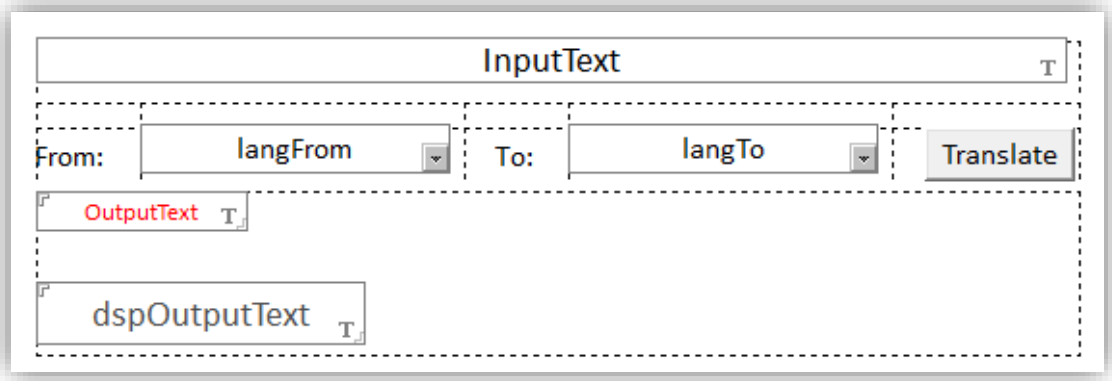
Lotusscript Code

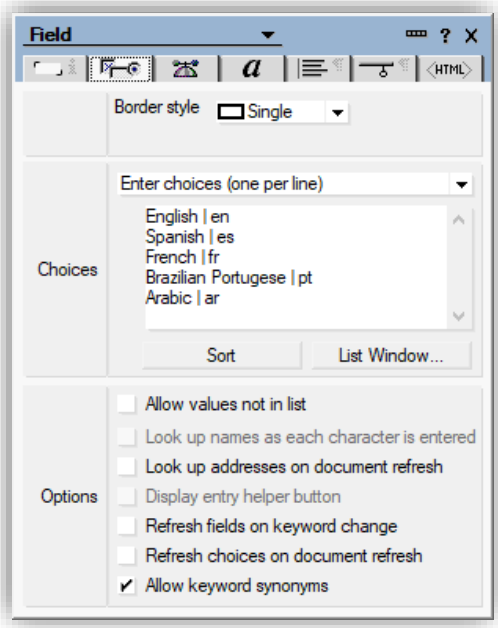
```
Use "class.RemoteHTTP"
Sub Click(Source As Button)
    Dim ws As New NotesUIWorkspace
    Dim uidoc As NotesUIDocument
    Dim http As New RemoteHTTP()
    Dim textEnglish As String
    Dim textSpanish As String
    Dim url As String
    Dim response As String
    Set uidoc = ws.CurrentDocument
    textEnglish = uidoc.FieldGetText("InputText")
    url = "https://texasswedenodered.mybluemix.net/translate"
    url = url + "?message="
    url = url + Replace(textEnglish, " ", "+")
    response = http.GetHTTP(url)
    textSpanish = parseJSON(response, "translated")
    Call uidoc.FieldSetText("OutputText", textSpanish)
    Call uidoc.Refresh()
End Sub
```

Demo

Improvements

- Select source and destination languages
- Add two dropdowns to form

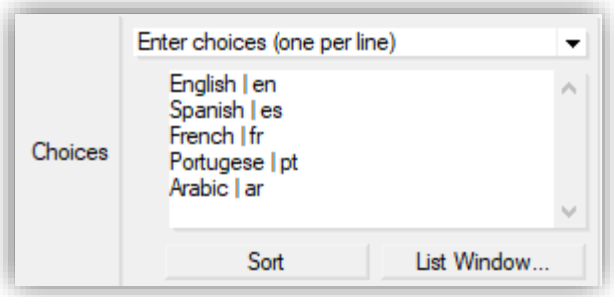




- Pass languages as arguments
  - Just a few more lines of code

Lotusscript Code

```
Use "class.RemoteHTTP"
Sub Click(Source As Button)
    Dim ws As New NotesUIWorkspace
    Dim uidoc As NotesUIDocument
    Dim http As New RemoteHTTP()
    Dim textEnglish As String
    Dim textSpanish As String
    Dim url As Strings String
    Dim response As Stringing
    Set uidoc = ws.CurrentDocument
    textEnglish = uidoc.FieldGetText("InputText")
    url = "https://texasswedenodered.mybluemix.net/translate"
    url = url + "?"from="+uidoc.document.GetItemValue("langFrom")(0)+"&to="+ _
        uidoc.document.GetItemValue("langTo")(0)+"message="
    url = url + Replace(textEnglish, " ", "+")To")(0)
    response = http.GetHTTP(url)ered.mybluemix.net/translate"
    textSpanish = parseJSON(response, "translated")essage="
    Call uidoc.FieldSetText("OutputText", textSpanish)
    Call uidoc.Refresh()TTP(url)
End SubtSpanish = parseJSON(response, "translated")
    Call uidoc.FieldSetText("OutputText", textSpanish)
    Call uidoc.Refresh()
End Sub
```



Demo

Text-to-Speech Flow

Node-RED

Deploy

filter nodes

Translation

input

inject

catch

status

link

mqtt

http

websocket

tcp

mqlight

ibmiot

output

debug

link

Text in

Edit text to speech node

Delete

Cancel

Done

Name: Convert to Speech

Language: US English

Voice: Michael

Format: OGG

info

debug

Node

Name	Convert to Speech
Type	watson-text-to-speech
ID	4a90a2f2.e2c0bc

Properties

lang	"en-US"
langhidden	"en-US"
langcustom	"NoCustomisationSetting"
langcustomhidden	""
voice	"en-US_MichaelVoice"
voicehidden	"en-US_MichaelVoice"
format	"audio/ogg; codecs=opus"
password	""

The Text To Speech service understands text and natural language to generate synthesized audio output complete with appropriate cadence and intonation.

You can choose different voices for a range of languages:.

The text to be converted should be passed in on `msg.payload`.

Process Incoming Request

```
// Set the payload for the next node to the text we want Watson to speak
msg.payload = msg.payload.text;
// Pass object to the next node (text-to-speech)
return msg;
```

Modify Response

```
// Set the payload for the response to the audio file
// created and passed to us in msg.speech
msg.payload = msg.speech;
// Pass the updated object to the HTTP Response node
return msg;
```

Call Node From Web

HTML:

```
<input type="text" id="text" placeholder="Enter text here">
<button id="btnSpeak" class="btn btn-primary"><i class="fa fa-volume-up"></i>Play</button>
<div id="iframeHost"></div>
```

jQuery:

```
$("#btnSpeak").on("click", function(e) {
    e.preventDefault();
    // Remove any existing iframe
    $("#iframeHost").html("");
    // Set the location of Node-RED flow to use
    var baseURL = "https://texasswedenodered.mybluemix.net/speak";
    // Get text to convert to speech
    var text = $("#text").val();
    // Calculate the URL of speech file to insert into iframe
    var url = baseURL+"?text=" + text;
    // Create iframe element in memory
    var iframe = $('<iframe id="audioframe" src="'+url+'" frameborder="1" width="10px"
    height="10px" style="display:none;"></iframe>');
    // Insert iframe element into div on page
    $("#iframeHost").append(iframe);
});
```

Demo

Bonus Demo

Conclusion

- BlueMix
  - Easy to setup
  - Convenient to use
  - Pay-as-you-go
- Node-RED
  - Click-and-drag of nodes
  - Javascript to manipulate
- Integration in web pages and Notes client

Questions?

Thank You

Karl-Henry Martinsson

karl-henry@demandbettersolutions.com

@texasswede

@texasswede

http://www.demandbettersolutions.com

http://www.texasswede.com

http://blog.texasswede.com