knowledge14®

April 27–May 1, 2014 • Moscone Center • San Francisco, CA

Integrations 301: Flexible Custom Web Services

John Andersen
SOLUTIONS ARCHITECT
ServiceNow

Jason Petty
EXPERT SERVICES
ServiceNow

servicenow®

© 2014 ServiceNow All Rights Reserved

Agenda

Learn how to create custom web services based off of a static WSDL

Learn how to create custom web services with generated WSDL

Create Scripted Path Processors

Created Scripted Parameter Processors

© 2014 ServiceNow All Rights Reserved

The Standard Out of Box Interfaces

A Review

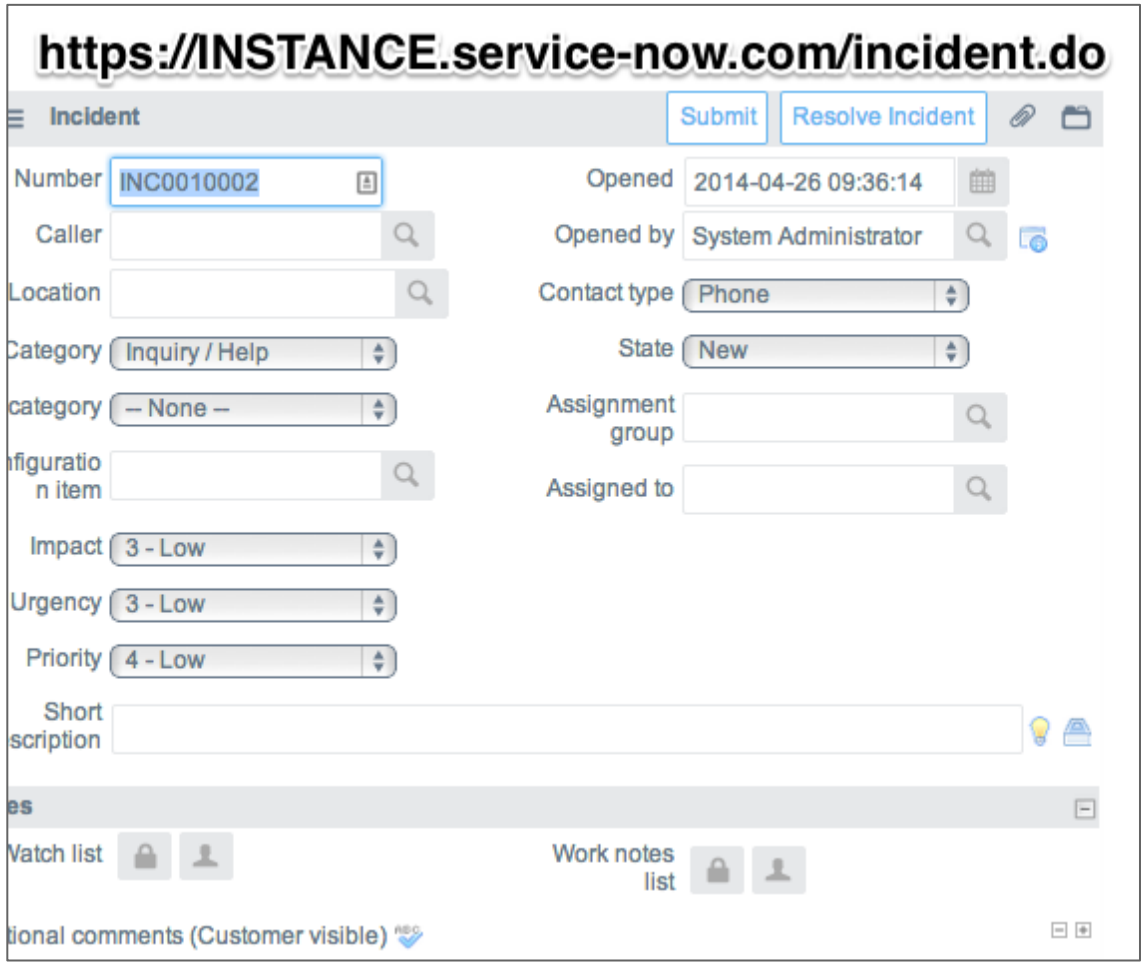
© 2014 ServiceNow All Rights Reserved

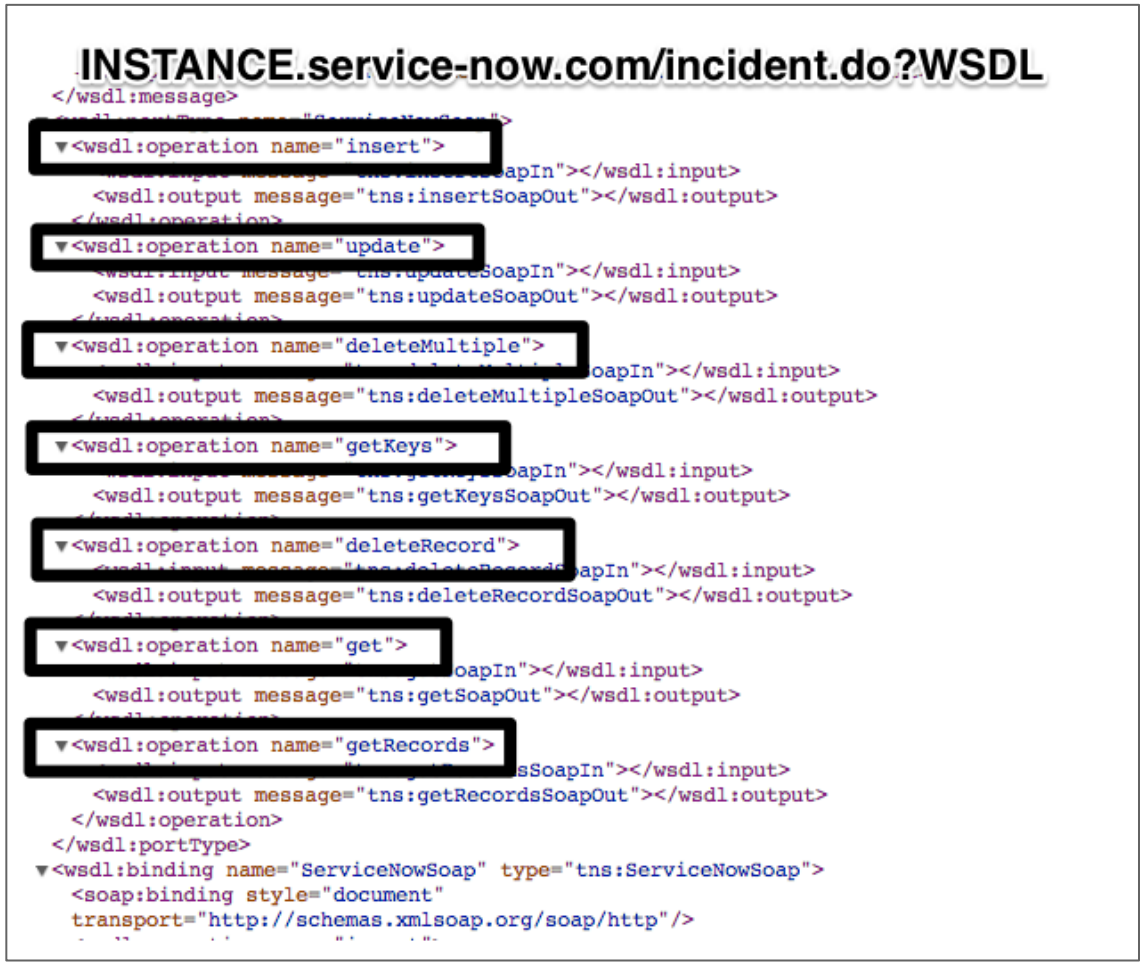
Direct Table Web Service Interface

- CRUD Access to Database Tables
  - Create
  - Read
  - Update
  - Delete
- Benefits
  - Fast & Easy to use
  - Auto generated
- Drawbacks
  - Third party has to understand the database schema
  - Invalid data formats could be set on a record
    - E.g. Choice Lists
  - No qualifying of data

© 2014 ServiceNow All Rights Reserved

Accessing the Direct Table Interface





© 2014 ServiceNow All Rights Reserved

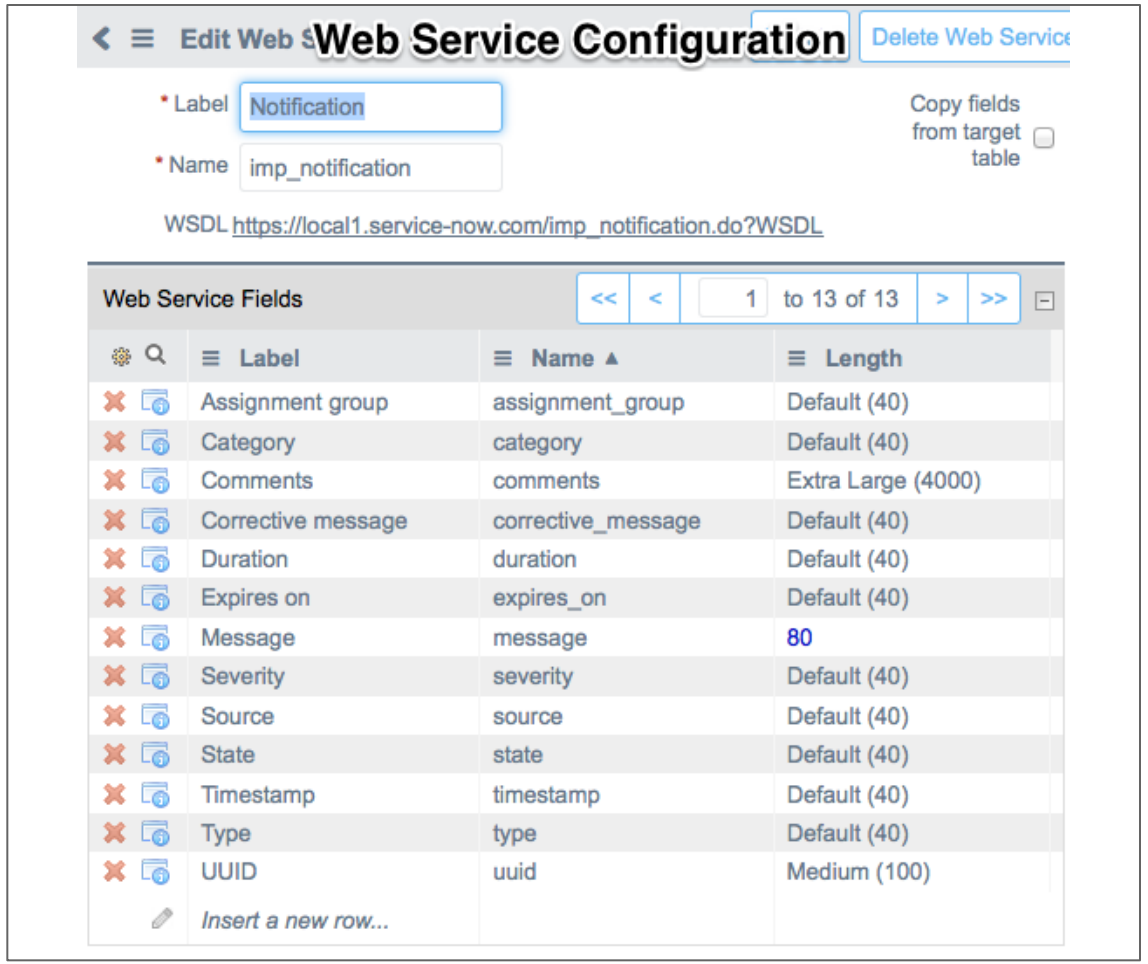
Web Service Import Set

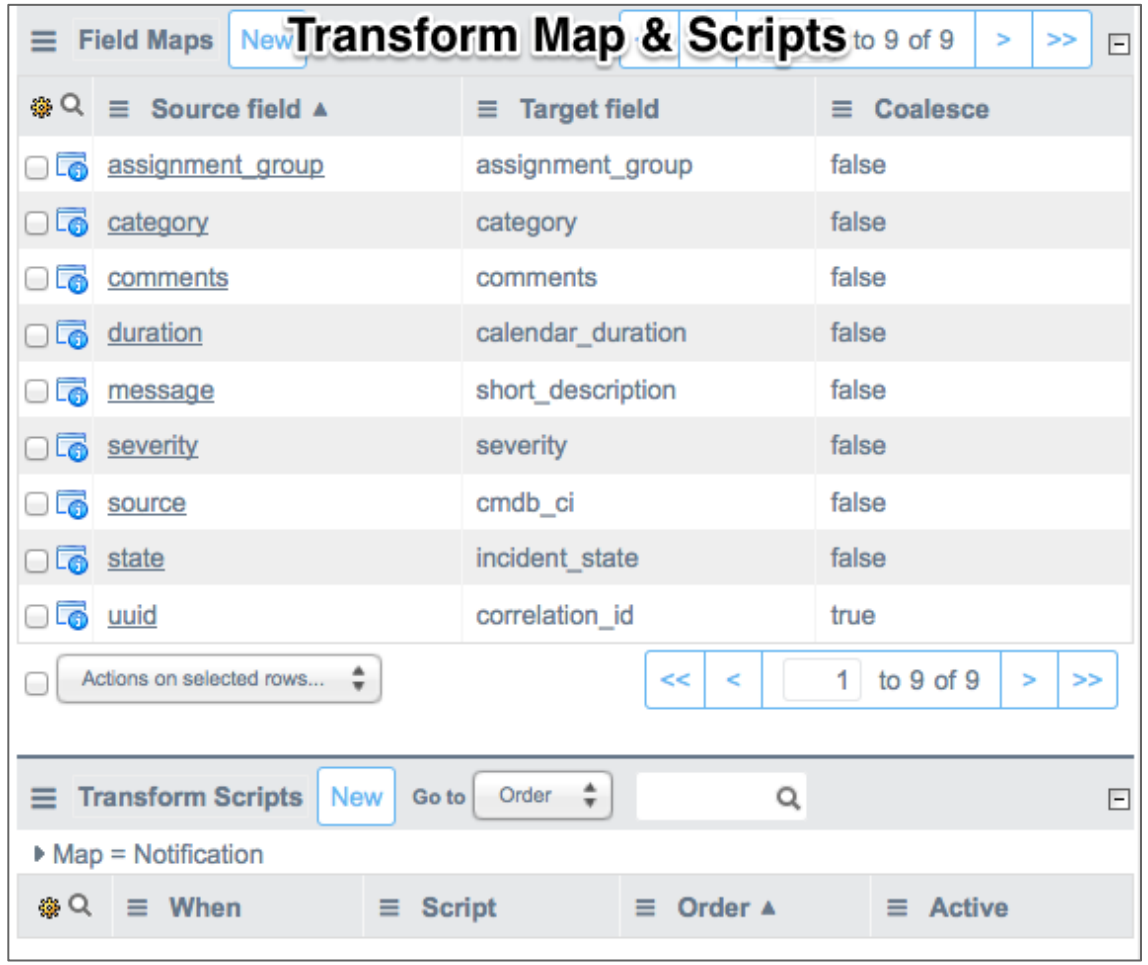
- Configurable Web Service Broker
- Provides interface to one or more database tables
- Maps third party schema to ServiceNow
- Transforms incoming data
- Qualifies incoming data
- Automatically determines actions to take
- Best Practice for Writing/Updating records

- Benefits
  - Third party doesn't need to know the ServiceNow schema
  - Third party only has to post and Insert
  - Can add custom response elements
  - You get a record of data coming in – great for troubleshooting
- Drawbacks
  - These have to be created manually
  - Adds additional database operations and CPU usage

© 2014 ServiceNow All Rights Reserved

Setting up Web Service Import Sets





© 2014 ServiceNow All Rights Reserved

Protocols Supporting Direct Table API and Web Service Import API

- SOAP
  - XML data
- JSON / JSONv2 (unofficial REST API)
  - JSON data
- REST API (available in Eureka)
  - XML data
  - JSON data

© 2014 ServiceNow All Rights Reserved

Custom Web Services

- Expose an interface to do virtually anything in the ServiceNow system
- Provide a Service rather than database table manipulate
- Conform to static requirements of a third party
- Provide asynchronous web services
- Considerations
  - Caution: Consider Security
    - GlideRecordSecure
  - Can still leverage web service import sets
    - ImportSetUtil library
    - Glide Record

© 2014 ServiceNow All Rights Reserved

Custom Web Service Options

- Scripted Web Services
  - SOAP based
  - XML data and response
  - Build a template to auto-generate a WSDL
  - Create your own WSDL
- Processors
  - More RESTful
  - HTTP Functions accepted
  - Incoming data in any format
  - Outgoing data in any format (string based)
  - No templates at this time

© 2014 ServiceNow All Rights Reserved

Custom/Static WSDLs

The basics

© 2014 ServiceNow All Rights Reserved

What is a WSDL

- Web Services Description Language
- Just and XML Document
- Catalog of available web service functions
- Tells you where the SOAP endpoint is located



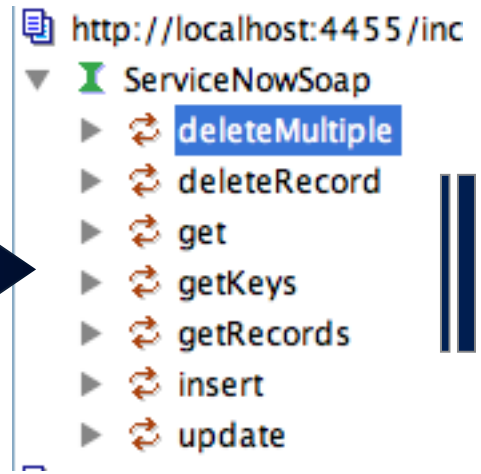
© 2014 ServiceNow All Rights Reserved

Example

Raw WSDL Document



Explains what is available



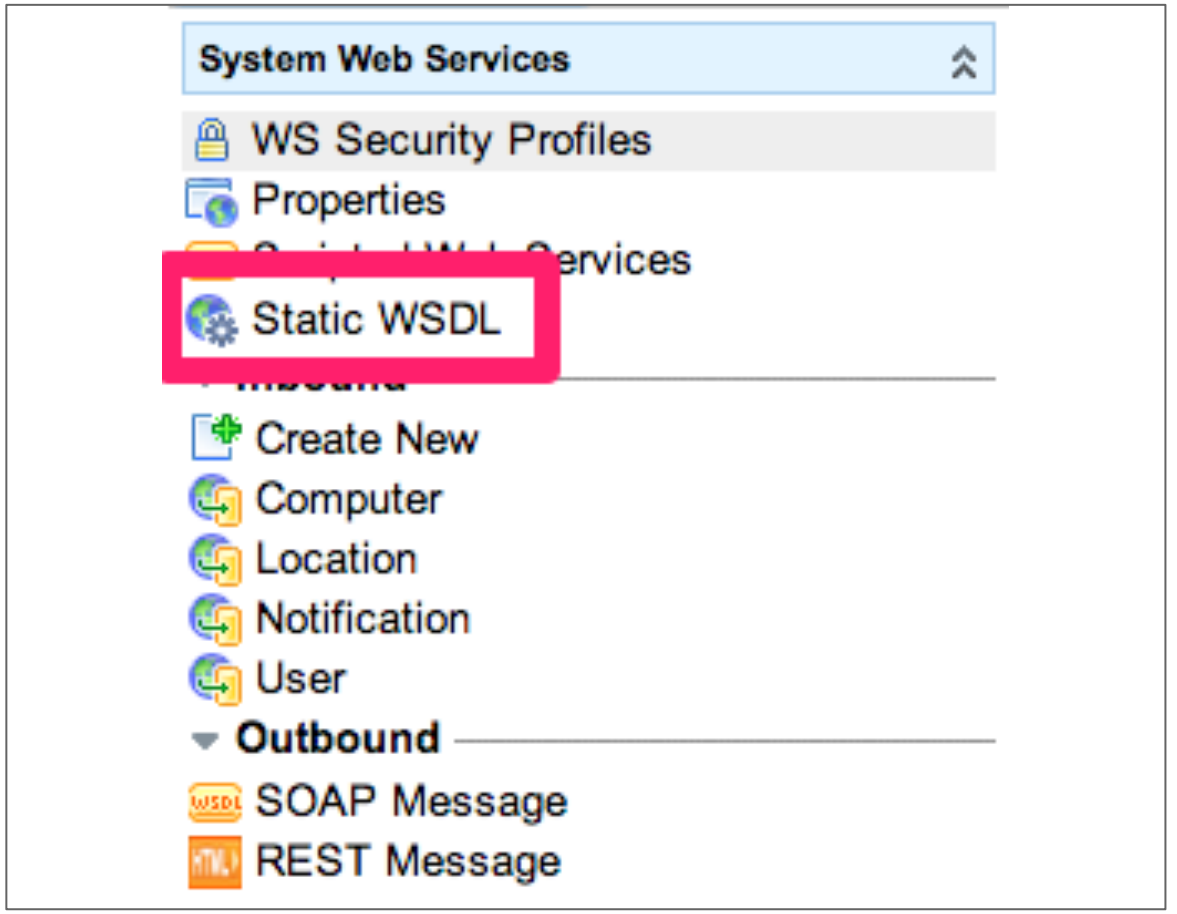
Shows you how to ask for it

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmls
   <soapenv:Header/>
   <soapenv:Body>
      <inc:deleteMultiple>
         <!--Optional:-->
         <active>?</active>
         <!--Optional:-->
         <activity_due>?</activity_due>
         <!--Optional:-->
         <approval>?</approval>
         <!--Optional:-->
         <approval_history>?</approval_history>
         <!--Optional:-->
         <approval_set>?</approval_set>
         <!--Optional:-->
         <assigned_to>?</assigned_to>
         <!--Optional:-->
         <assignment_group>?</assignment_group>
         <!--Optional:-->
         <business_duration>?</business_duration>
         <!--Optional:-->
         <business_stc>?</business_stc>
         <!--Optional:-->
         <calendar_duration>?</calendar_duration>
         <!--Optional:-->
         <calendar_stc>?</calendar_stc>
         <!--Optional:-->
```

© 2014 ServiceNow All Rights Reserved

Custom or Static WSDLs

- Plugin: “Web Service Provider – Custom WSDL”
- Navigate to it through the App Navigation Pane
  - System Web Services > Static WSDL



© 2014 ServiceNow All Rights Reserved

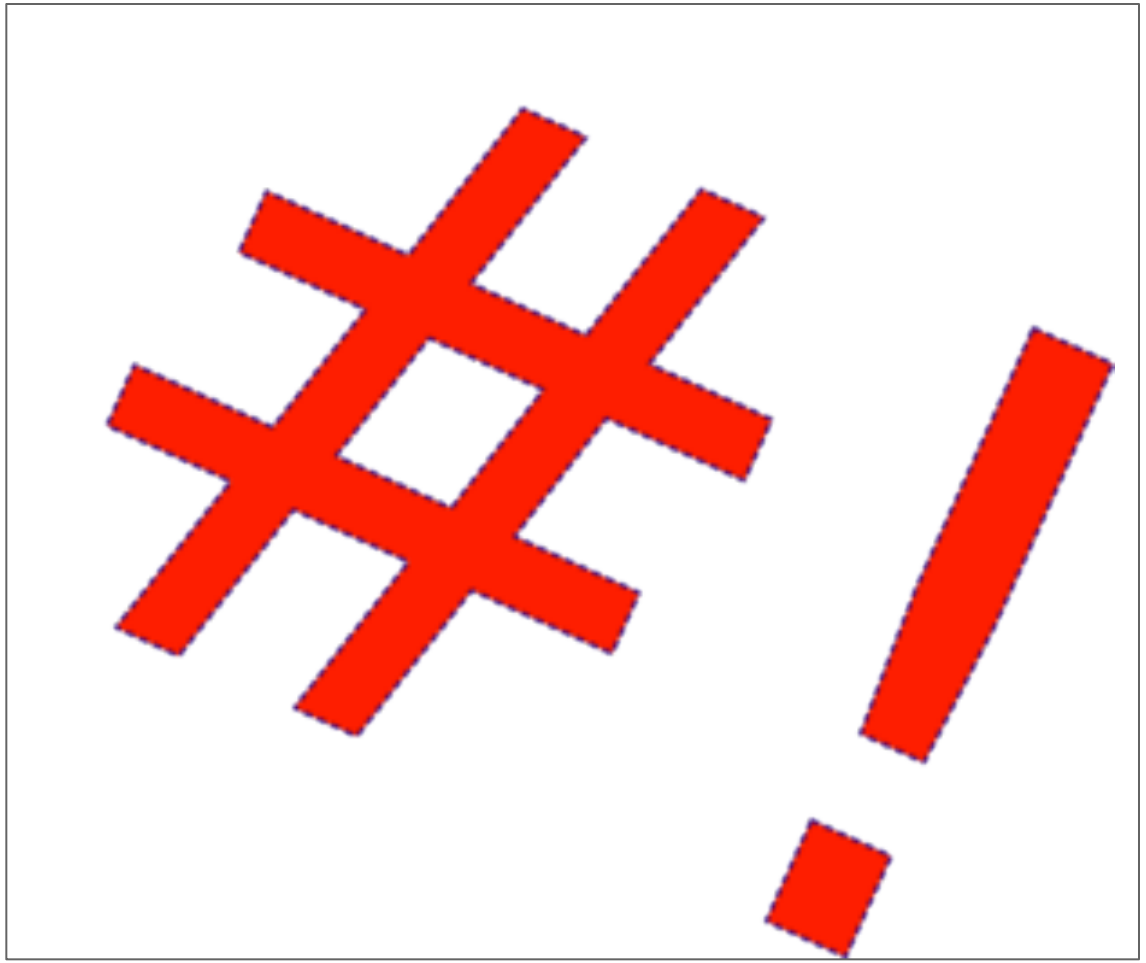
Exercise #1

Adding a Custom WSDL to your Instance

© 2014 ServiceNow All Rights Reserved

Scripted Web Services

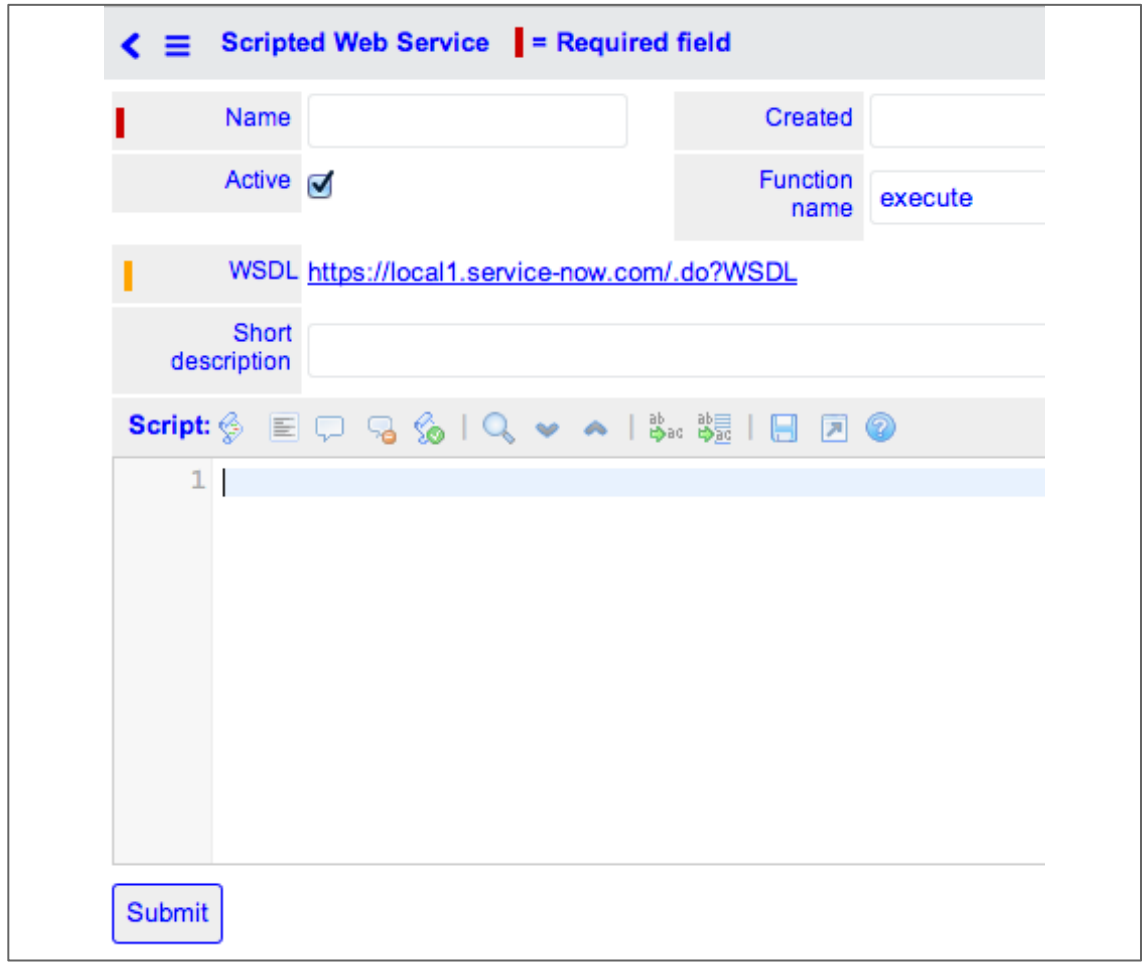
- Leverage two different WSDL Options
  - Custom/Static WSDL
  - Autogenerated WSDL
- WSDL vs. Scripted Web Service
  - WSDL describes the custom service
  - Scripted Web Service implements the service
- When to use Static WSDL
  - Third part is dictating the SOAP format
  - If you want to handle multiple functions in a single service



© 2014 ServiceNow All Rights Reserved

Implementing the Scripted Web Service

- Name (this will be the access point)
  - https://myinstance/NAME.do?SOAP
  - If using a Static WSDL the name will need to be the same as that of the Static WSDL
- Function Name
  - Anything you wish for auto-generated WSDLs
  - Should match a function name of a static WSDL if that static WSDL only has one function
- Short Description
  - Optional and Arbitrary
  - Used to help describe the web service
- Script
  - The code that implements the web service



© 2014 ServiceNow All Rights Reserved

Scripted Web Service with Auto WSDL Generation

- Input Parameters
  - XML elements that will be expected in the SOAP Body
- Output Parameters
  - XML Fields that the client should expect in the response



© 2014 ServiceNow All Rights Reserved

Special Variables offered in Scripted Web Service record

INCOMING REQUEST

- soapRequestXML
  - XML String of the incoming SOAP Envelope
  - Options
    - Query it with string functions
    - Convert it to a javascript object with XMLHelper
    - Convert it to an XMLDocument
- request
  - Contains input values from the request
  - Useful for auto-generated WSDL scenarios
  - request.INPUT_PARAM_NAME to get value

OUTGOING RESPONSE

- response
  - Simple: .fieldname
  - Complex: soapResponseElement

© 2014 ServiceNow All Rights Reserved

Grabbing incoming SOAP parameters with XMLDocument

```
<soapenv:Envelope>
   <soapenv:Header/>
   <soapenv:Body>
      <rtes:execute>
         <firstname>John</firstname>
         <lastname>Andersen</lastname>
      </rtes:execute>
   </soapenv:Body>
</soapenv:Envelope>
```

```
1 var xml = new XMLDocument(soapRequestXML);
2 firstname = xml.getNodeText("//execute/firstname");
3 lastname = xml.getNodeText("//execute/lastname");
```

© 2014 ServiceNow All Rights Reserved

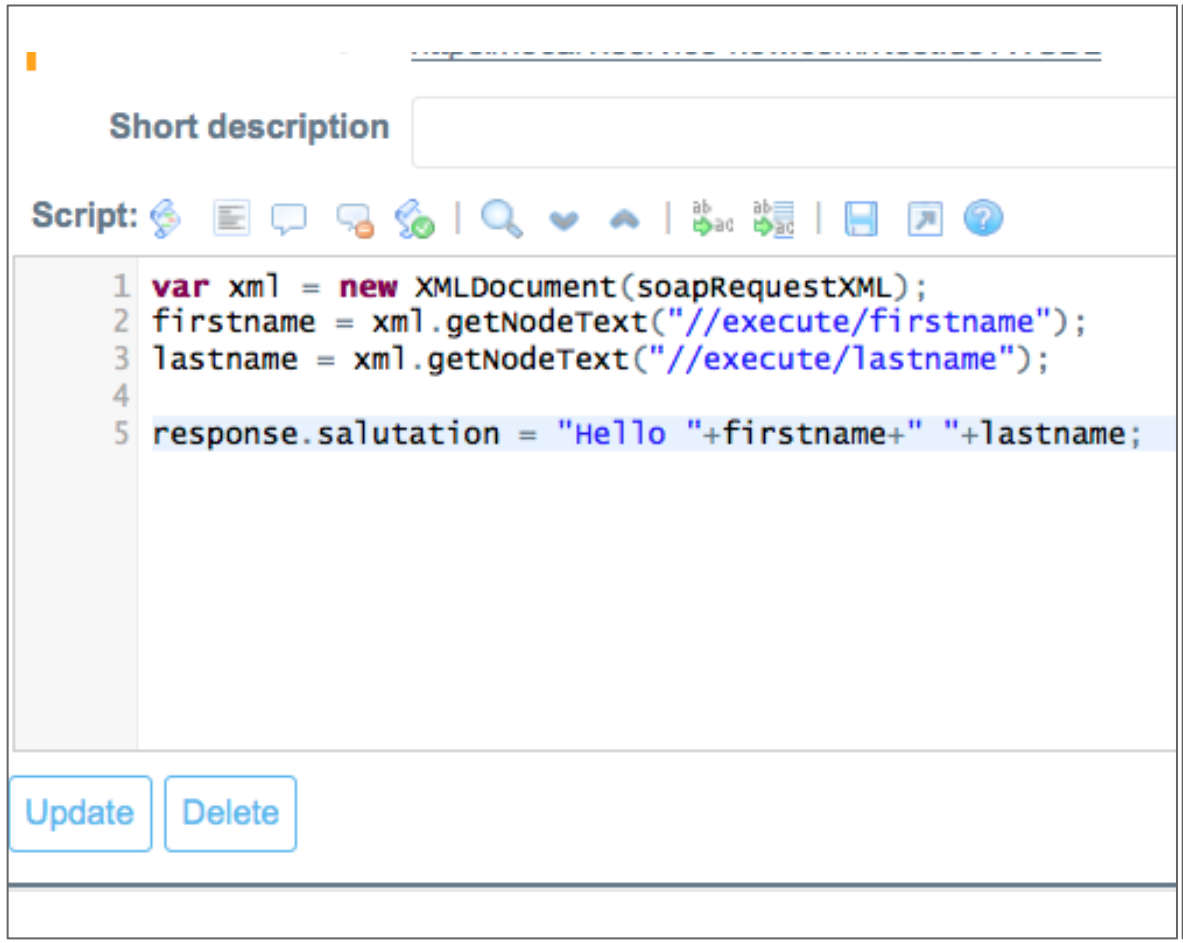
Grabbing incoming SOAP parameters in Auto-Generated WSDL mode

```
<soapenv:Envelope>
   <soapenv:Header/>
   <soapenv:Body>
      <rtes:execute>
         <firstname>John</firstname>
         <lastname>Andersen</lastname>
      </rtes:execute>
   </soapenv:Body>
</soapenv:Envelope>
```

```
1 var firstname = request.firstname;
2 var lastname = request.lastname;
```

© 2014 ServiceNow All Rights Reserved

Building a Simple Response in Scripted Web Service




```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap
   <SOAP-ENV:Body>
      <executeResponse>
         <salutation>Hello John Andersen</salutation>
      </executeResponse>
   </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

© 2014 ServiceNow All Rights Reserved

Building a Complex Response in Scripted Web Service

Script:

```
1  var xml = new XMLDocument(soapRequestXML);
2  firstname = xml.getNodeText("//execute/firstname");
3  lastname = xml.getNodeText("//execute/lastname");
4
5  var xmlstring = "<person> " +
6   "  <name>"+firstname+" "+lastname+"</name> " +
7   "  <sex>M</sex> " +
8   "  <location> " +
9   "    <city>Shelley</city> " +
10  "    <state>ID</state> " +
11  "  </location> " +
12  "</person> ";
13 var rXML = new XMLDocument(xmlstring);
14
15 response.soapResponseElement = rXML.getDocumentElement();
```

Update Delete

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="h
   <SOAP-ENV:Body>
      <person>
         <name>John Andersen</name>
         <sex>M</sex>
         <location>
            <city>Shelley</city>
            <state>ID</state>
         </location>
      </person>
   </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

© 2014 ServiceNow All Rights Reserved

Exercise 2

Implement the SpaceCurrency web service

© 2014 ServiceNow All Rights Reserved

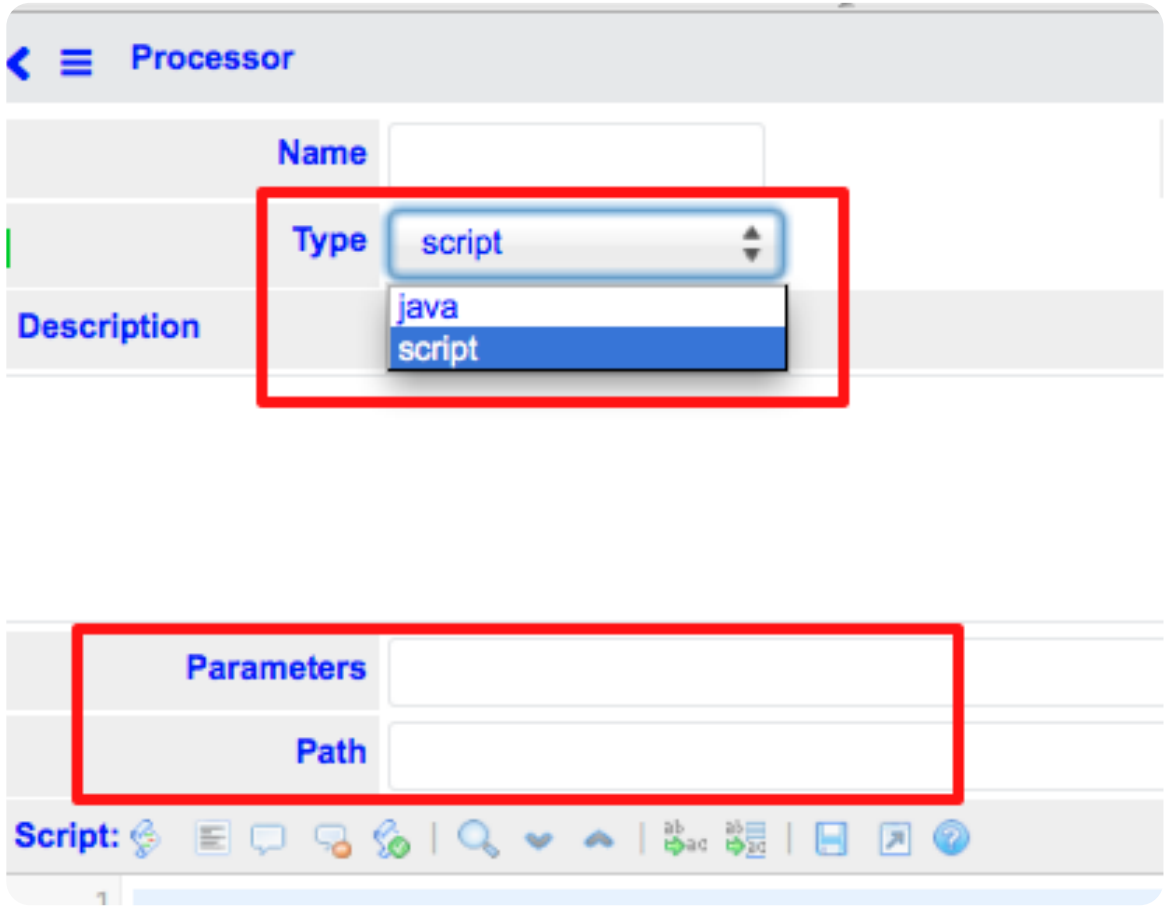
Exercise 3

Using Auto Generated WSDLs

© 2014 ServiceNow All Rights Reserved

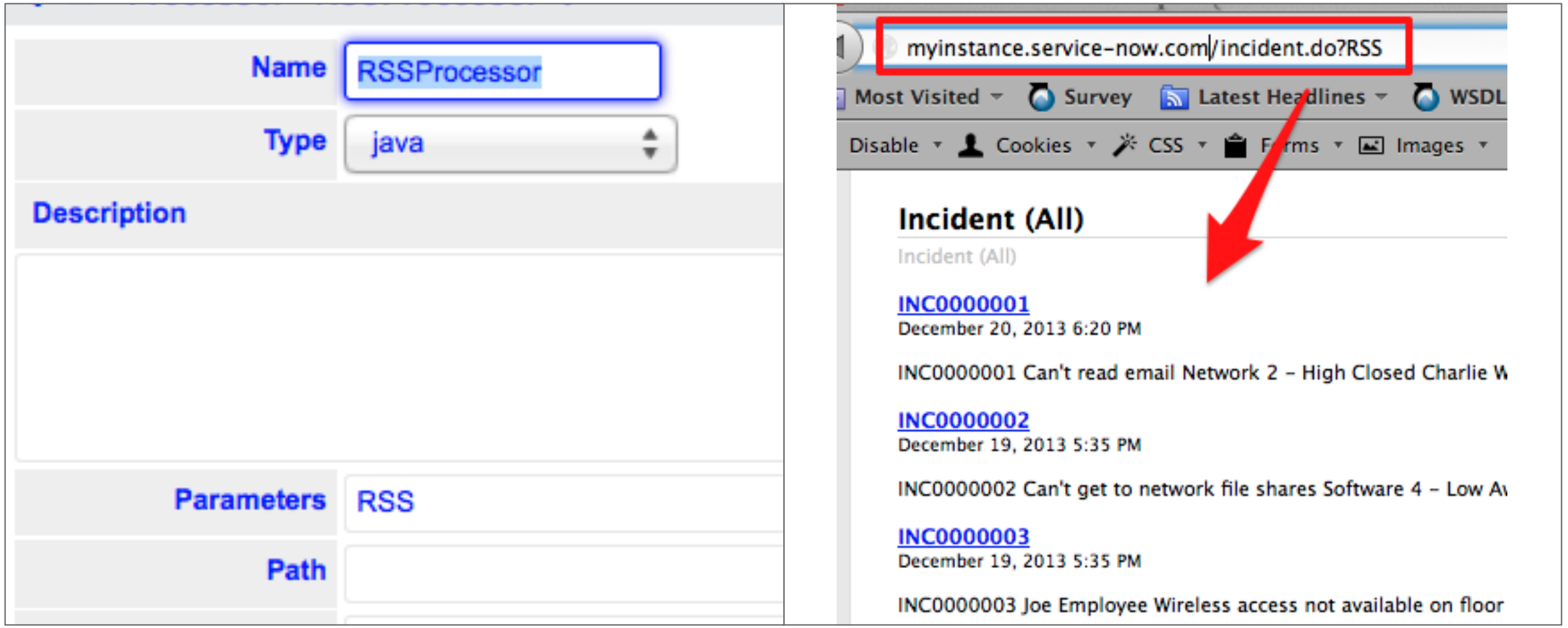
Introduction to Processors

- Processors are to REST as Scripted Web Services are to SOAP…almost
- Configurable records that can trigger a script or action
- Scripted vs. Java Processors
  - Scripted are our only options
  - Java is for ServiceNow developers only
- Parameter Processors
  - Triggered by a special parameter on the request
  - Great for table specific actions
- Path Processors
  - Triggered by a specified url page
  - Good for specific or global service



© 2014 ServiceNow All Rights Reserved

Parameter Processor Example



© 2014 ServiceNow All Rights Reserved

Path Processor Example

Processor

Name: DownloadFile

Type: script

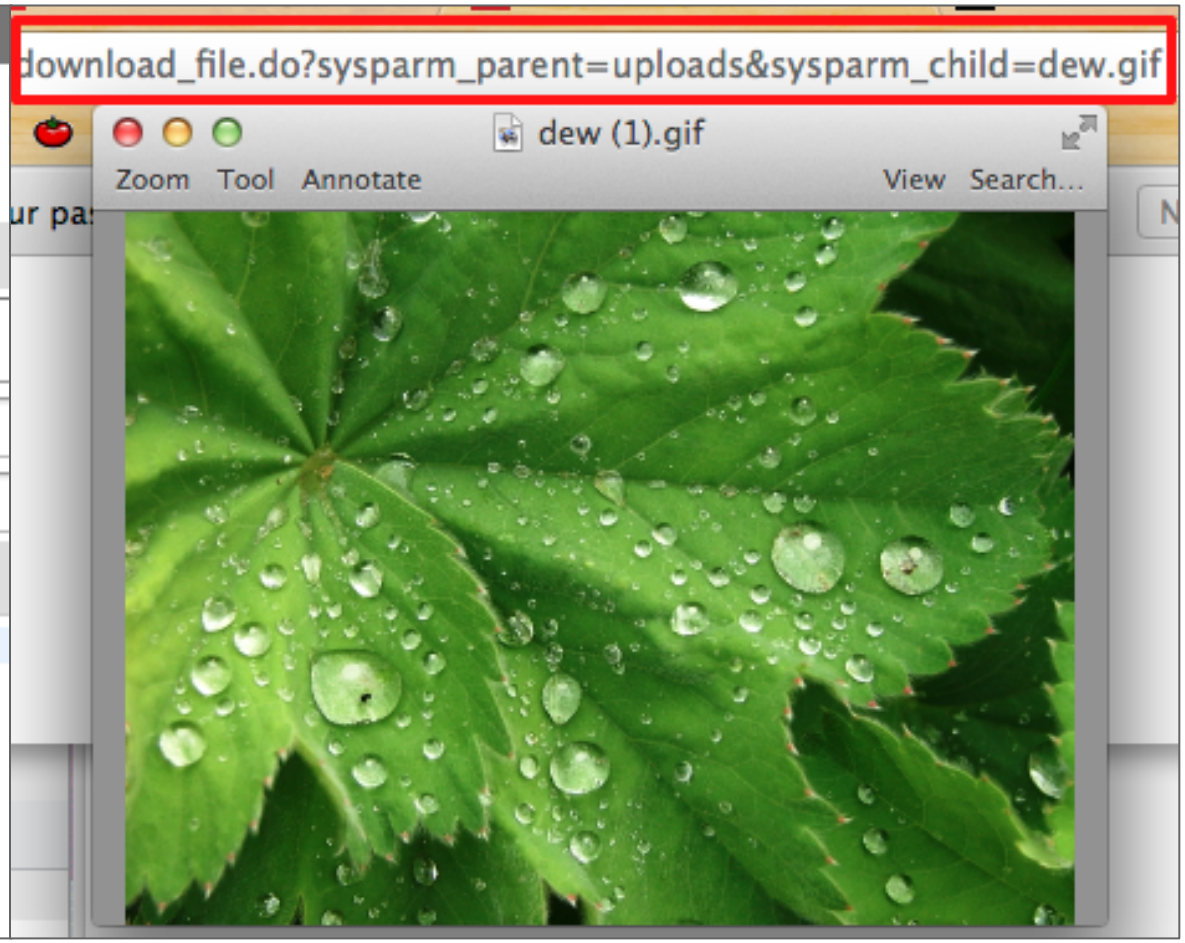
Description: File downloader

Parameters:

Path: download_file

Script:

```
1 var parent = g_request.getParameter("sysparm_parent");
2 var child = g_request.getParameter("sysparm_child");
3 var gzip = false;
4
5 if (child.endsWith('.out') || child.endsWith('.txt'))
6     gzip = true;
7
8 var sa = new SecurelyAccess(parent, child);
9 sa.download(g_response, gzip);
```



© 2014 ServiceNow All Rights Reserved

Available Functions in Processor Scripts

- g_request – the HTTP Request Object
  - getMethod() – PUT, POST, GET, DELETE, etc
  - getHeaderNames() – returns HTTP Headers
  - getHeader(headerName) – returns header value
  - getParameterNames() – returns HTTP Params
  - getParameter(paramName) – returns HTTP param
  - getInputStream() – streaming data from request
- g_response – the HTTP Response Object
  - setHeader(name, val) – set return headers to client
- g_processor – the processor object
  - writeOutput(output) – the string to be written back to the client

© 2014 ServiceNow All Rights Reserved

Exercise 4

Create a Path Processor

© 2014 ServiceNow All Rights Reserved

Exercise 5

Create a Parameter Processor

© 2014 ServiceNow All Rights Reserved

Top Takeaways

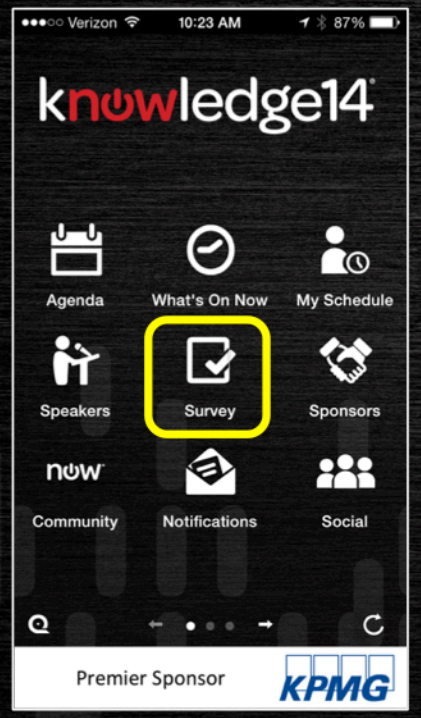
1 Scripted Web Services are to SOAP as Processors are to REST

2 Scripted Web Services allow for customized SOAP web services

3 Scripted Processors can provide customizable RESTful services

© 2014 ServiceNow All Rights Reserved

Tell us what you think!



Click on the survey link in your Knowledge14 mobile app. Help us make Knowledge 15 even more valuable!

Thank You

John Andersen

Sr. Solutions Architect

ServiceNow

Twitter: @arbonboy

Jason Petty

Expert Services

ServiceNow

jason.petty@servicenow.com

Want a copy of this deck?

Sign into the BuzzHub to unlock all the slides and lab guides from Knowledge14 sessions

