knowledge14®

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

# Integration 102: Discover the Power of REST

Chris Maloy
Team Lead Expert Services
ServiceNow

Jonatan Jardi
Senior Technical Consultant
ServiceNow

servicenow®

© 2014 ServiceNow All Rights Reserved

# Eureka and Fuji disclaimer

- Eureka REST API – it is awesome and if you want to learn more about it
  - Go to the breakout session right after this “Wed 1:40 PM – 2:30 PM Rapid App Integrations with the New REST API” presented by Mark Brennan, John Andersen from ServiceNow

- The plan is to have Import Set support, Oauth support, and scriptable web service support

© 2014 ServiceNow All Rights Reserved

# Objectives

- Learn how to use the GUI based REST Message feature
- Understand how to make REST based requests through server-side scripts
- Leverage the JSON Web Service to provide third party applications with a REST-ful interface to ServiceNow
- Understand the JSONv2 Processor changes

© 2014 ServiceNow All Rights Reserved

# What is REST?

- **RE**presentational **S**tate **T**ransfer
- Lightweight alternative to SOAP
- Very loose specification
- Better suited to scale & performance
- Most modern web applications are REST

© 2014 ServiceNow All Rights Reserved

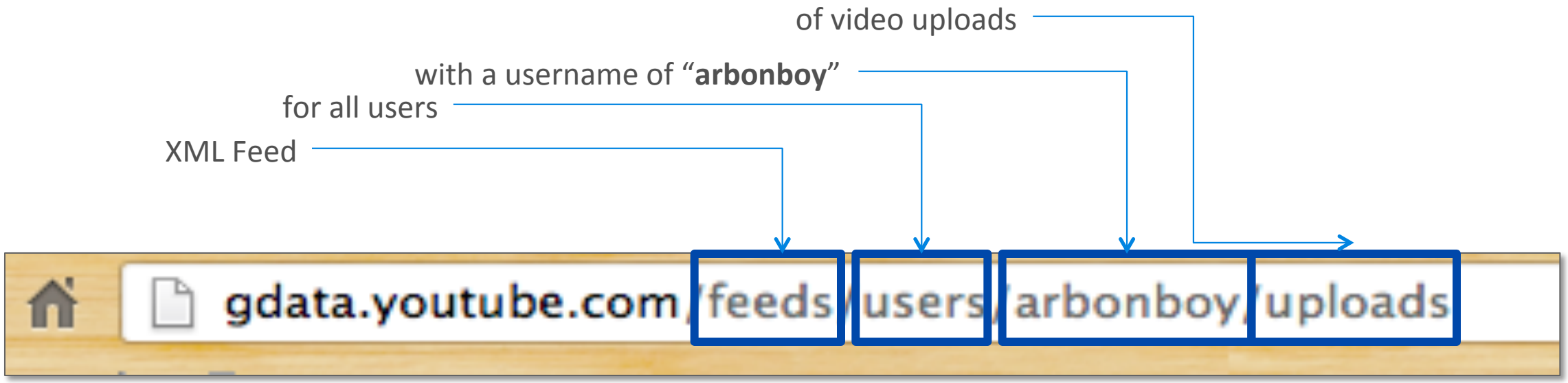
# Programmatic Representation of Data

- Consider viewing a user's uploads on YouTube
  - http://www.youtube.com/user/arbonboy
- Same view, but with REST
  - http://gdata.youtube.com/feeds/users/arbonboy/uploads

© 2014 ServiceNow All Rights Reserved

# Resources with REST

- Componentized access to service resources
- YouTube Example




© 2014 ServiceNow All Rights Reserved

# Support HTTP Methods

- GET
- POST
- PUT
- DELETE

© 2014 ServiceNow All Rights Reserved

# HTTP Request & Response



REQUEST URL

HTTP Headers

Content (optional)

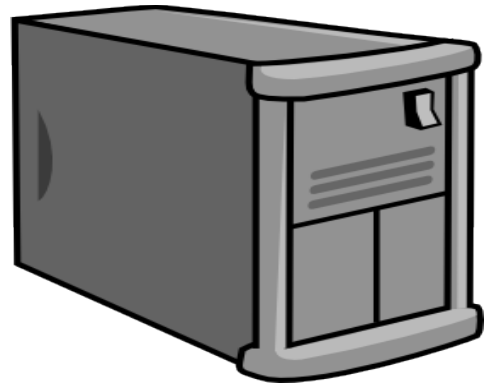
**HTTP Request**

**HTTP Response**

Status Code

HTTP Headers

Content



© 2014 ServiceNow All Rights Reserved

# HTTP Methods used with REST

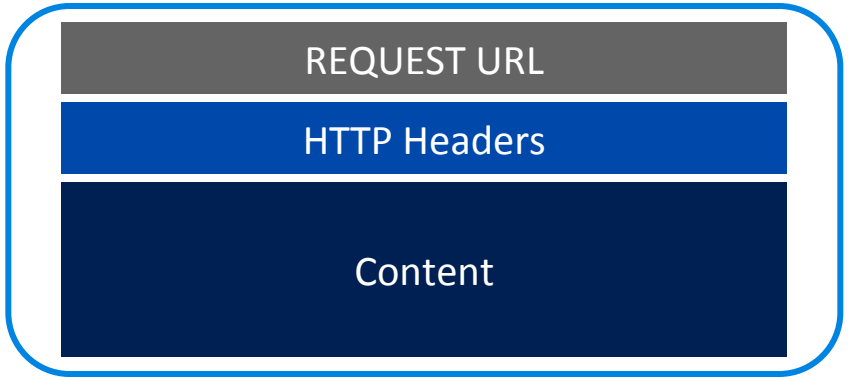
- GET
  - Typically used for queries
  - HTTP Components
    - Request URL & Parameters
    - HTTP Headers




© 2014 ServiceNow All Rights Reserved

# HTTP Methods used with REST

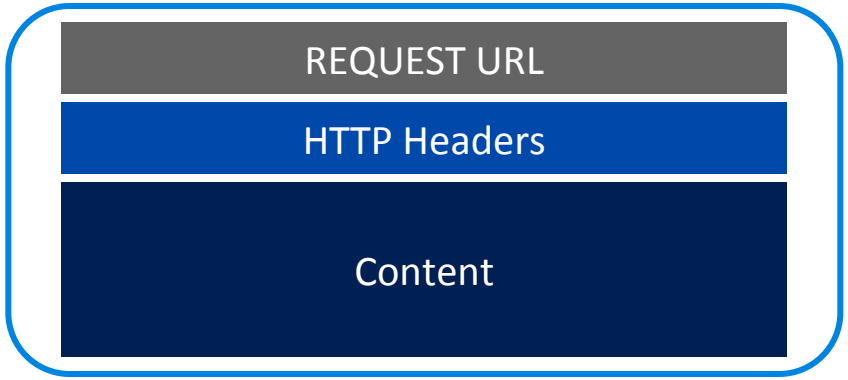
- POST
  - Typically used to create a resource
  - HTTP Components
    - Request URL & Parameters
    - HTTP Headers
    - Content




© 2014 ServiceNow All Rights Reserved

# HTTP Methods used with REST

- PUT
  - Typically used to modify a resource
  - HTTP Components
    - Request URL & Parameters
    - HTTP Headers
    - Content




© 2014 ServiceNow All Rights Reserved

# HTTP Methods used with REST

- DELETE
  - Typically used to delete a resource
  - HTTP Components
    - Request URL & Parameters
    - HTTP Headers




© 2014 ServiceNow All Rights Reserved

# Making REST calls from ServiceNow

© 2014 ServiceNow All Rights Reserved

# Outbound REST Calls

- GUI based tool – “REST Message”
- Similar to the “SOAP Message” tool
- Requires an “endpoint”
- Generates the four HTTP Methods as functions
- Endpoint can use variables
  - ${varName}

© 2014 ServiceNow All Rights Reserved

# REST Function Options

- Variables can be used in various parts of the function
  - Endpoint URL
  - Header names or values
  - Function parameter definition names or values
  - Content

© 2014 ServiceNow All Rights Reserved

# REST Function Options

- Connectivity options
  - Basic Authentication
  - MID Server
- Access to HTTP request components
  - Headers
  - Parameters
  - Content (POST, PUT only)

© 2014 ServiceNow All Rights Reserved

# REST Function Options

- Set variable values
  - String vs. XML
- Test the request
- Preview script usage

© 2014 ServiceNow All Rights Reserved

# Exercise 1

© 2014 ServiceNow All Rights Reserved

# Scripting with REST Message

- Make REST requests in scripts
  - Business Rules
  - Script Includes
  - Workflows
  - Background scripts
- REST Message Library
  - Script Includes
    - RESTMessage
    - RESTResponse
    - RESTECCResponse
    - RESTMessageGenerator

© 2014 ServiceNow All Rights Reserved

# Handling REST response

- Synchronous
  - getStatusCode()
  - getHeader(<headerName>)
  - getHeaders()
  - getBody()
  - haveError()
  - getErrorCode()
  - getErrorMessage()

- Asynchronous
  - getStatusCode()
  - getBody()
  - haveError()
  - getErrorCode()
  - getErrorMessage()

© 2014 ServiceNow All Rights Reserved

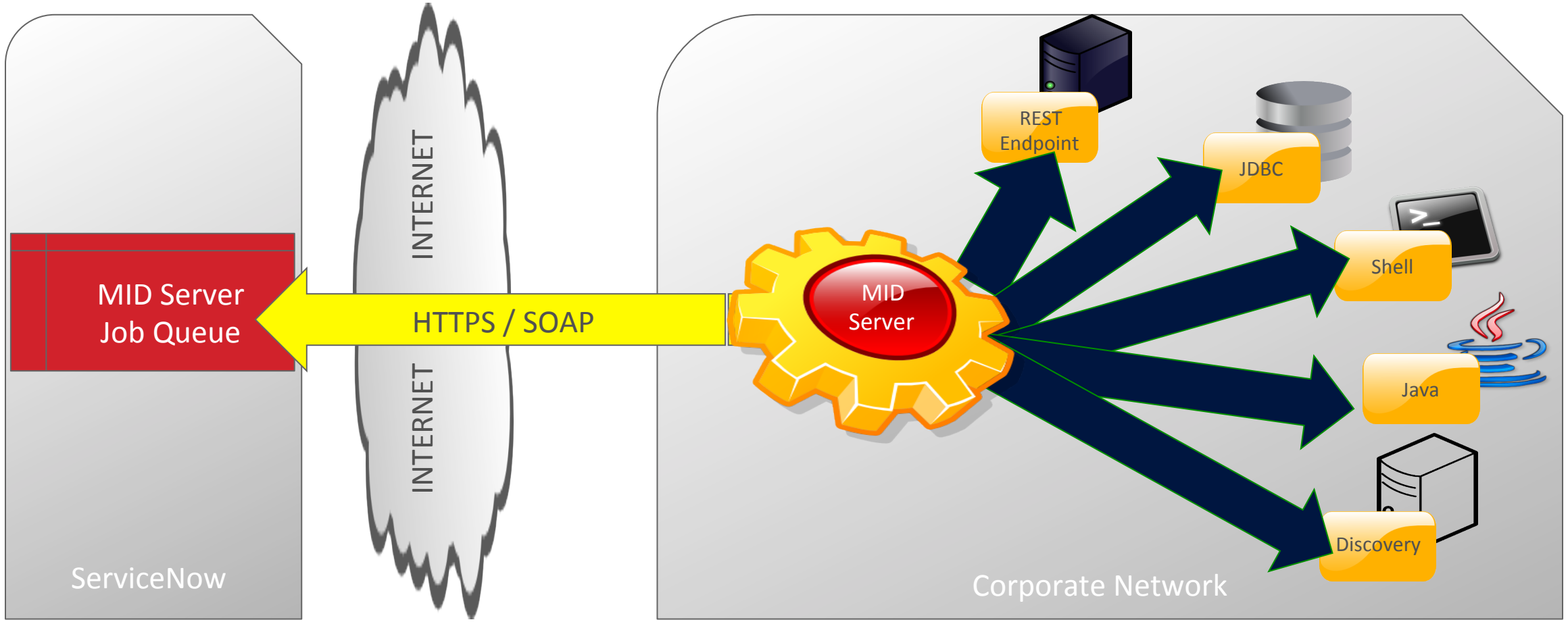
# Handling a JSON Response

- JSONParser
  - Takes a JSON string and generates a Javascript Object

```
var jsonText = "{'name':'Yoda', 'age':'896'}";
var parser = new JSONParser();
var parsed = parser.parse(jsonText);
```

© 2014 ServiceNow All Rights Reserved

# MID Server – Working behind the firewall




© 2014 ServiceNow All Rights Reserved

# REST Message through MID Server

- When would you want to do this?
- Process
  - Outbound message placed on ECC Queue
  - MID Server takes queue record and processes
  - Places response back on ECC Queue
- Limitations
  - MID server is delayed

© 2014 ServiceNow All Rights Reserved

# Getting a response from MID Server

- execute() does not return response
- Have to do a loop using getResponse(ms)

```
r.execute();
var k = 1;
var response = r.getResponse();
while (response == null) {
    gs.log("waiting ... " + k + " seconds");
    //wait 1 second before looking for the response
    response = r.getResponse(1000); k++;

    if (k > 30) {
        break;
    }
}
var body = response.getBody();
```

© 2014 ServiceNow All Rights Reserved

# Exercise 2

© 2014 ServiceNow All Rights Reserved

# Inbound REST Calls

© 2014 ServiceNow All Rights Reserved

# Does ServiceNow have a REST API?

- Yes!
- JSONv2 Processor
- http://wiki.servicenow.com/index.php?title=JSONv2_Web_Service
- Standard
  - CRUD Operations on all tables
  - Web Service Import Sets
- Custom-ability
  - Scripted Web Services
  - Processors

© 2014 ServiceNow All Rights Reserved

# JSONv2 Processor

- Enabled by default
- Allows JSON-based interaction with ServiceNow
  - Queries (GET)
  - Modification of data (POST)
- HTTP POST / GET (no PUT, DELETE)
- *The content-type of the POST should be application/json*
- Protected via Basic Authentication
- All same functions supported by old JSON plugin still work
- Utilizes streaming and Java libraries to improve performance.

© 2014 ServiceNow All Rights Reserved

# JSON Web Service Processor basics

- URL of the table
  - https://<instance name>.service-now.com/mytable.do?**JSONv2**
- URL Parameters
  - JSONv2
  - sysparm_action
  - sysparm_query (filtering)
  - sysparm_view
  - sysparm_sys_id
  - sysparm_record_count
  - displayvalue
  - displayvariables
- HTTP Header
  - Basic Auth credentials
- Content (JSON)

© 2014 ServiceNow All Rights Reserved

# Data Retrieval and Modification

- Data Modification
  - Insert
  - insertMultiple
  - update
  - deleteRecord
  - deleteMultipe
- READ
  - getKeys
  - getRecords
  - get

© 2014 ServiceNow All Rights Reserved

# Data Retrieval

- You can query for data by issuing an HTTPS GET request. By default, a GET request is interpreted as a get function if a sysparm_sys_id parameter is present. Otherwise, it is interpreted as a getRecords function. You can also specify a URL parameter sysparm_action=get.
- To modify the JSON web service, issue an HTTPS POST request. The HTTP POST **must** contain a sysparm_action parameter to indicate the type of action to be performed, with the incoming **JSON object post in the body**.

© 2014 ServiceNow All Rights Reserved

# JSON Response

- **JSON Success Response**
- Each JSON success response includes a record array containing the records retrieved by the given action. Each JSON object contains one or more metadata elements, prefixed with __, regarding the status for the action on each record, as illustrated in the previous examples. The JSON success responses use the following syntax:
- **__status**
- "__status": "<value>"
- where <value> is **success** or **failure**.

© 2014 ServiceNow All Rights Reserved

# JSON Web Service Processor basics

- **__error**
- When the status element returns **failure**, the error element is added to identify the error and reason.
- "__error": { "message": "<error value>", "reason": "<reason value> "}
- where <error value> is the error message text and <reason value> is the reason the error was triggered.
- **JSON Failure Response**
- Each JSON failure response contains only the error and reason elements, which indicate the error message and more details about the error. Generally, this indicates that the whole JSON operation failed and no records can be processed.
- For example:
- {"error":"Cannot update with empty sysparm_query","reason":null}

© 2014 ServiceNow All Rights Reserved

# New Noteworthy Properties

- glide.processor.json.row_limit
- glide.basicauth.required.jsonv2

© 2014 ServiceNow All Rights Reserved

# Exercise 3

© 2014 ServiceNow All Rights Reserved

# Top Takeaways

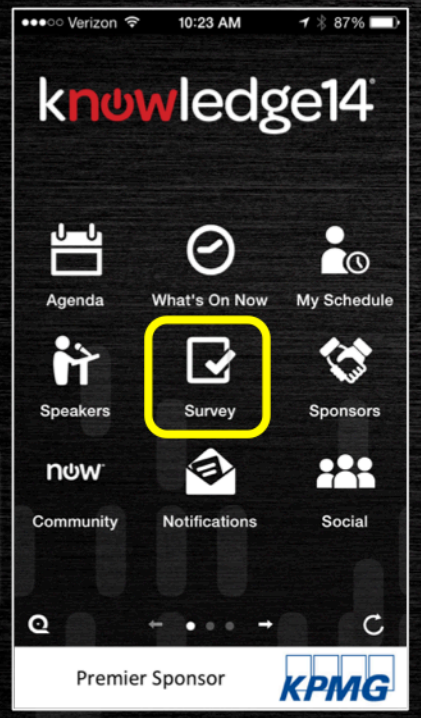
1. The REST Message feature provides a User Interface to setup and send REST calls.

2. You can have REST Message generate scripts that can be leveraged by business rules, workflows, etc.

3. The JSON Web Service provides a RESTful API into ServiceNow

© 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

**Chris Maloy**

*Expert Services Team Lead*

ServiceNow

chris.maloy@servicenow.com

**Jonatan Jardi**

*Senior Technical Consultant*

ServiceNow

jonatan.jardi@servicenow.com

# Want a copy of this deck?

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

