knowledge15

Everything as a Service

Integration 202:
Accessing the ServiceNow Web Service API

JOHN ANDERSEN
SR SOLUTION ARCHITECT
ServiceNow

CHRIS COLLAZO
SOLUTION ARCHITECT
ServiceNow

servicenow

Agenda

Understanding the Table and Import Set APIs

SOAP Web Service API Basics

REST Web Service API Table API Basics

REST Web Service Import API Basics

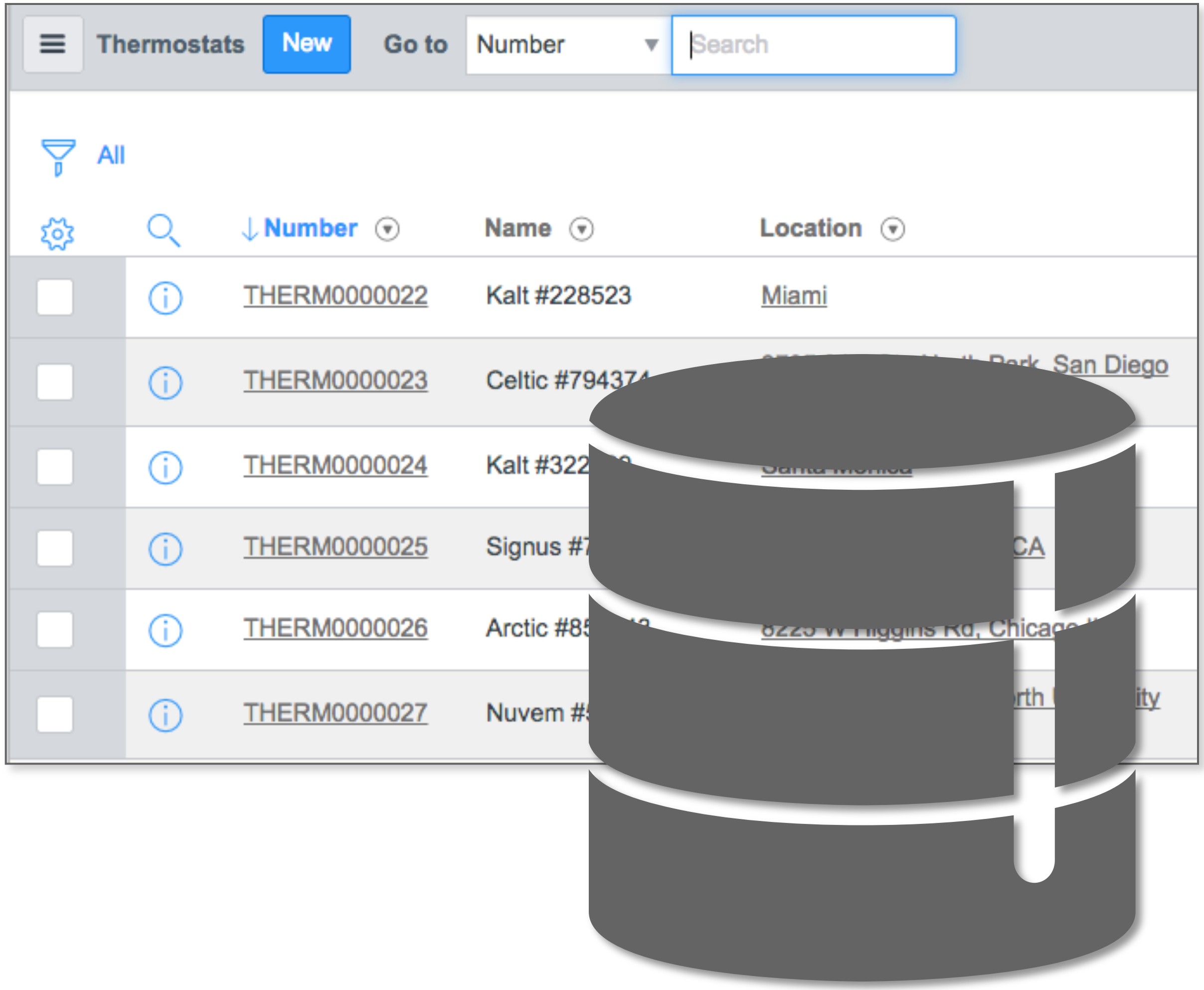
Export-based Web Services

Understanding the Table and Import Set API

Section 1

Table Based Web Service APIs

- Direct Table Access
- Full CRUD operations on a database table
- Access Controls
- Scoped Applications and Custom Tables
- No real transform capabilities
- Third party must understand and comply with ServiceNow scheme

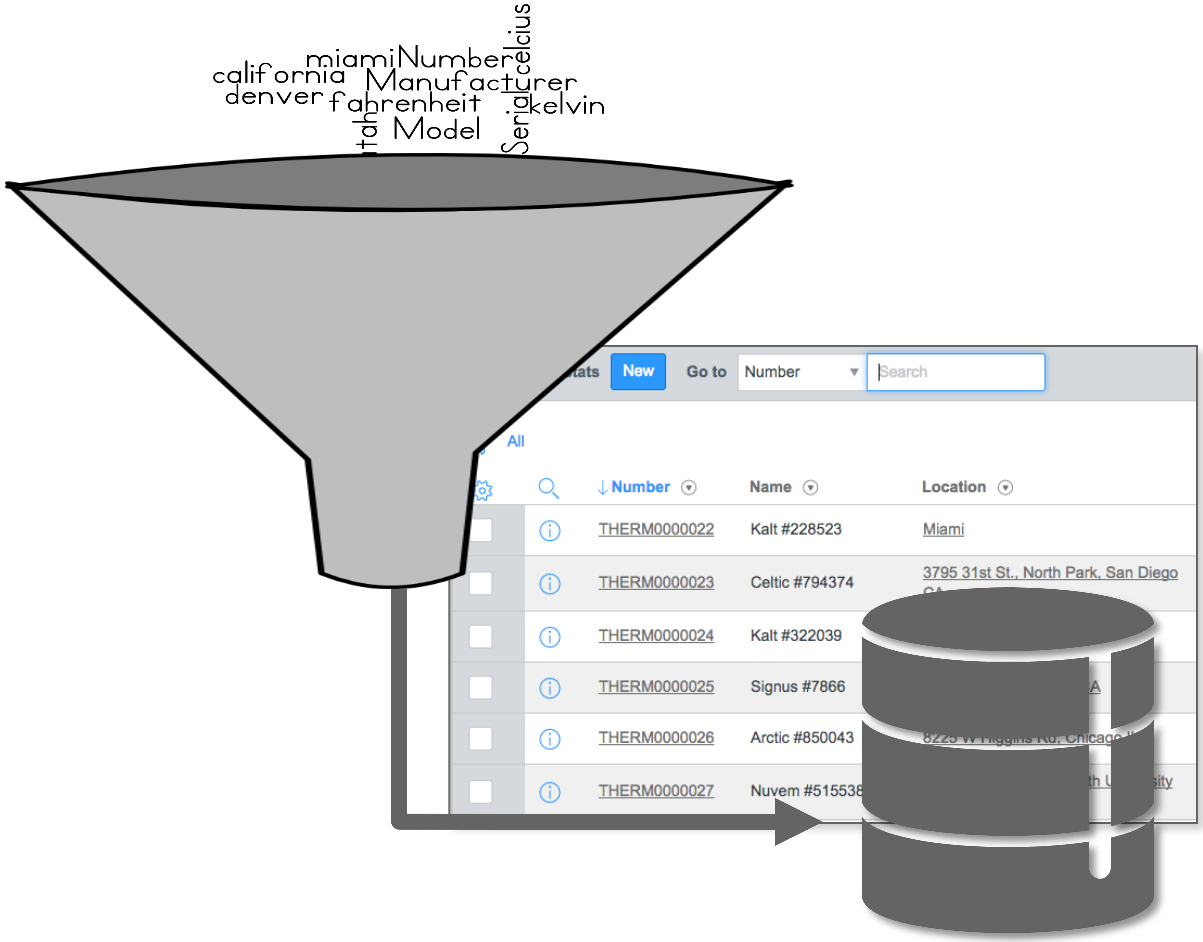


Number	Name	Location
THERM0000022	Kalt #228523	Miami
THERM0000023	Celtic #79437	Park, San Diego
THERM0000024	Kalt #322	Santa Monica
THERM0000025	Signus #7	CA
THERM0000026	Arctic #85	8225 W Higgins Rd, Chicago
THERM0000027	Nuvem #	rth ity

Import Set Web Service APIs

- Data flows through staging table
- Only need to create staging records – the system determines the correct operation
- Access Controls
- Mapping and Transform capabilities
- Coalescing capabilities
- Third party can send data in its own schema and format

Third party data and schema



	↓Number	Name	Location
ⓘ	THERM0000022	Kalt #228523	Miami
ⓘ	THERM0000023	Celtic #794374	3795 31st St., North Park, San Diego CA
ⓘ	THERM0000024	Kalt #322039	
ⓘ	THERM0000025	Signus #7866	A
ⓘ	THERM0000026	Arctic #850043	8225 W Higgins Rd, Chicago I
ⓘ	THERM0000027	Nuvem #515538	th U sity

Quick Demo

Tables and Import Sets Overview

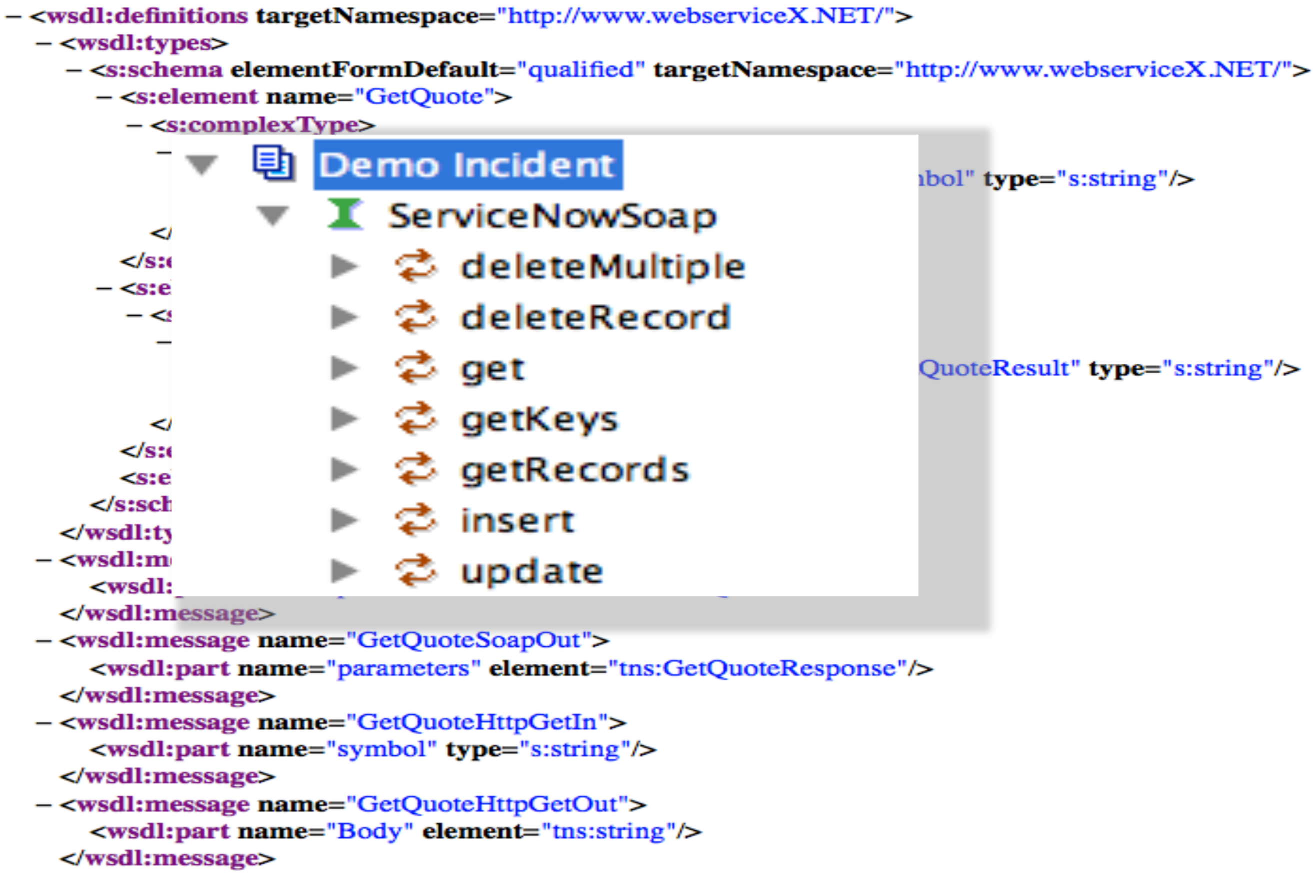
What is SOAP?

What is SOAP?

- Simple Object Access Protocol
- Communication between two systems
- Communication over the Internet
- Platform & Language independent
- XML based protocol

SOAP WSDL

- WSDL
  - Catalog of available functions
  - Templates of requests

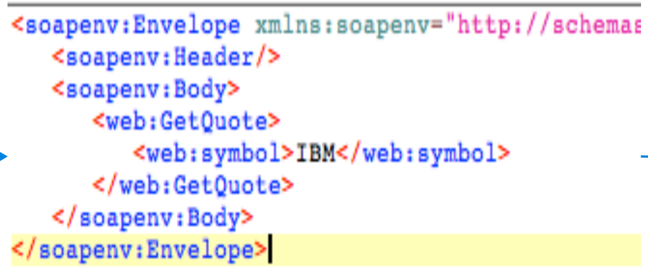


```
– <wsdl:definitions targetNamespace="http://www.webserviceX.NET/">
  – <wsdl:types>
    – <s:schema elementFormDefault="qualified" targetNamespace="http://www.webserviceX.NET/">
      – <s:element name="GetQuote">
        – <s:complexType>
          –                                                    bol" type="s:string"/>
          </
        </s:
      – <s:e
        – <
          –
                                                               QuoteResult" type="s:string"/>
          </
        </s:
        <s:e
      </s:sch
    </wsdl:ty
  – <wsdl:m
      <wsdl:
  </wsdl:message>
  – <wsdl:message name="GetQuoteSoapOut">
      <wsdl:part name="parameters" element="tns:GetQuoteResponse"/>
  </wsdl:message>
  – <wsdl:message name="GetQuoteHttpGetIn">
      <wsdl:part name="symbol" type="s:string"/>
  </wsdl:message>
  – <wsdl:message name="GetQuoteHttpGetOut">
      <wsdl:part name="Body" element="tns:string"/>
  </wsdl:message>
```

SOAP Envelope

- Main Document
- Header (optional)
  - containing processing instructions for the envelope
- Body
  - contains the actual message to be delivered
- Fault element (optional)
  - providers error information

SOAP Communication Flow

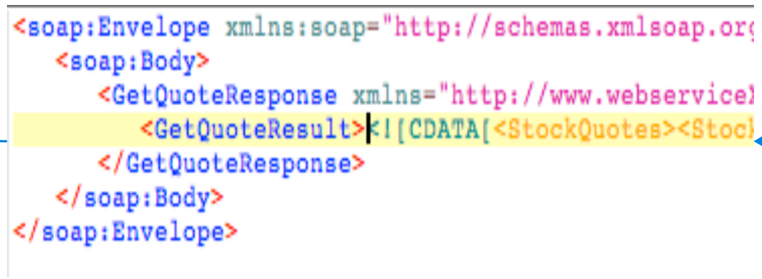


```
<soapenv:Envelope xmlns:soapenv="http://schemas
   <soapenv:Header/>
   <soapenv:Body>
      <web:GetQuote>
         <web:symbol>IBM</web:symbol>
      </web:GetQuote>
   </soapenv:Body>
</soapenv:Envelope>
```

1) Client sends SOAP Request

2) Server processes the SOAP Request

3) Server sends a SOAP Response



```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.or
   <soap:Body>
      <GetQuoteResponse xmlns="http://www.webservice
         <GetQuoteResult><![CDATA[<StockQuotes><Stoc
      </GetQuoteResponse>
   </soap:Body>
</soap:Envelope>
```

SOAP Web Service API Basics

SOAP Web Service APIs

- WSDL follows the format:
  - /incident.do?WSDL
  - /u_my_custom_table.do?WSDL
- Direct Table Web Service API
  - Create
  - Read
  - Update
  - Delete

- Web Service Import Set API
  - Create (insert)
  - Other CRUD functions available but not typically applicable

CRUD Operations

CREATE

- Insert
  - All fields are optional
  - Every non-system field on the table is available
- Update
  - Requires “sys_id” of the record to update
  - All other fields are optional
  - All non-system field on the table is available

CRUD Operations

READ

- GetKeys
  - Returns sys_id's of records matching the query
  - All fields are optional
  - System and non-system fields are available

READ (cont)

- Special Fields
  - __use_view
  - __encoded_query
  - __limit
  - __first_row
  - __last_row

CRUD Operations

READ

- GetRecords
  - Returns all data of records matching the query
  - All fields are optional
  - System, non-system, and special fields are available
    - __use_view
    - __encoded_query
    - __limit
    - __first_row
    - __last_row
    - __order_by
    - __order_by_desc
    - __exclude_columns

CRUD Operations

READ

- Get
  - Requires a sys_id
  - Returns all data of the single record

CRUD Operations

UPDATE

- Update
  - Requires a sys_id
  - All other fields optional
  - Any set field element will set the value in the record
  - Returns sys_id of the record that was modified

CRUD Operations

DELETE

- DeleteRecord
  - Requires a sys_id of record you wish to delete
  - Returns the count of the records that were deleted
    - 0 or 1 in this case
- DeleteMultiple
  - All fields optional
  - System and non-system fields available in query
  - Returns the count of the records that were deleted

Lab 1

SOAP Web Service API

Understanding REST

Section 3

What is REST?

A web service methodology

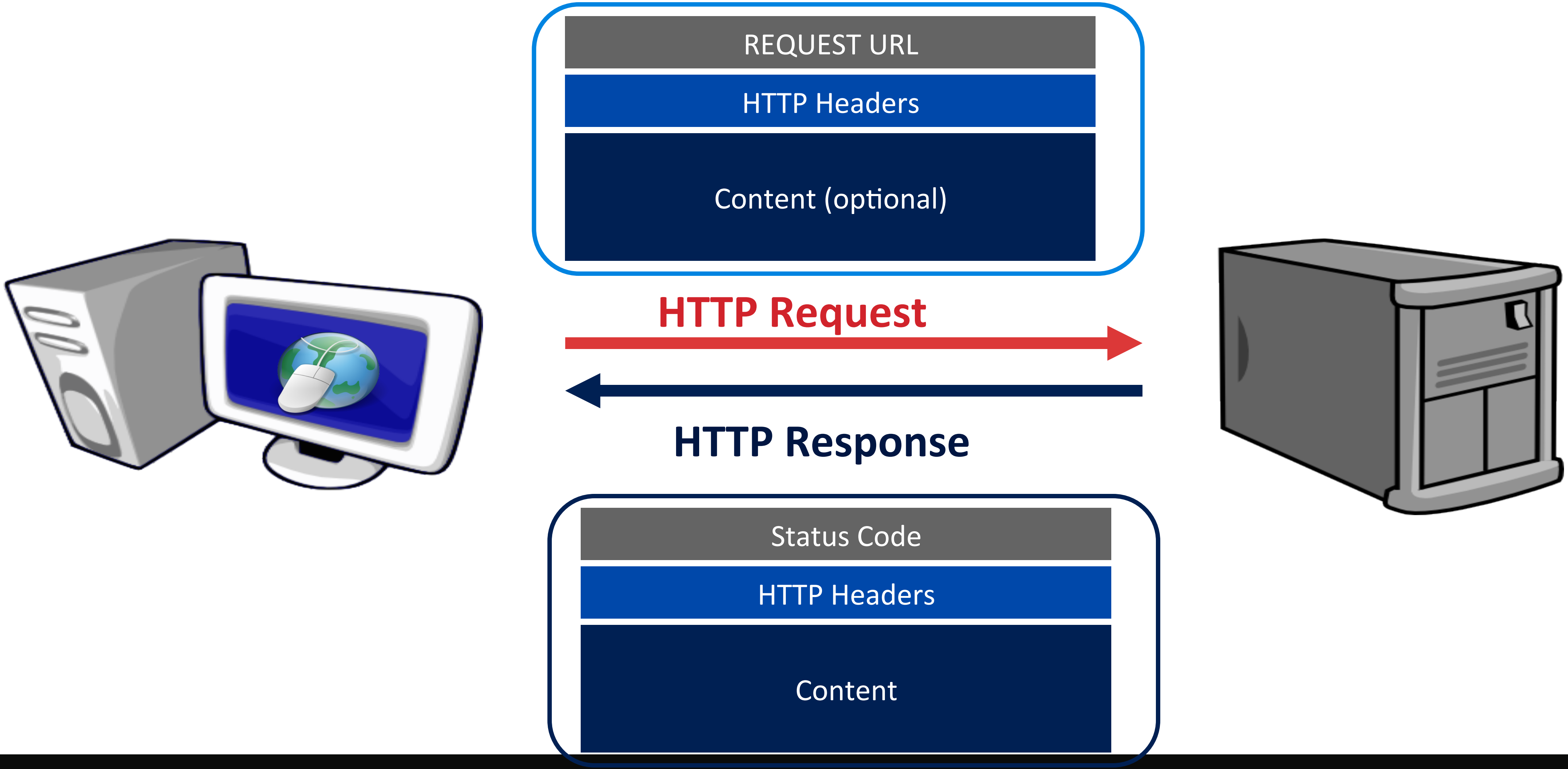
Attempts to decompose an application into smaller components or resources

Utilizes smaller HTTP-based communication for scalable interaction

Major components of a REST Message

- HTTP Function
- Uniform Resource Identifier (URI)
- HTTP Headers
- URL Parameters
- HTTP Body/Content

HTTP Request & Response



REQUEST URL

HTTP Headers

Content (optional)

HTTP Request

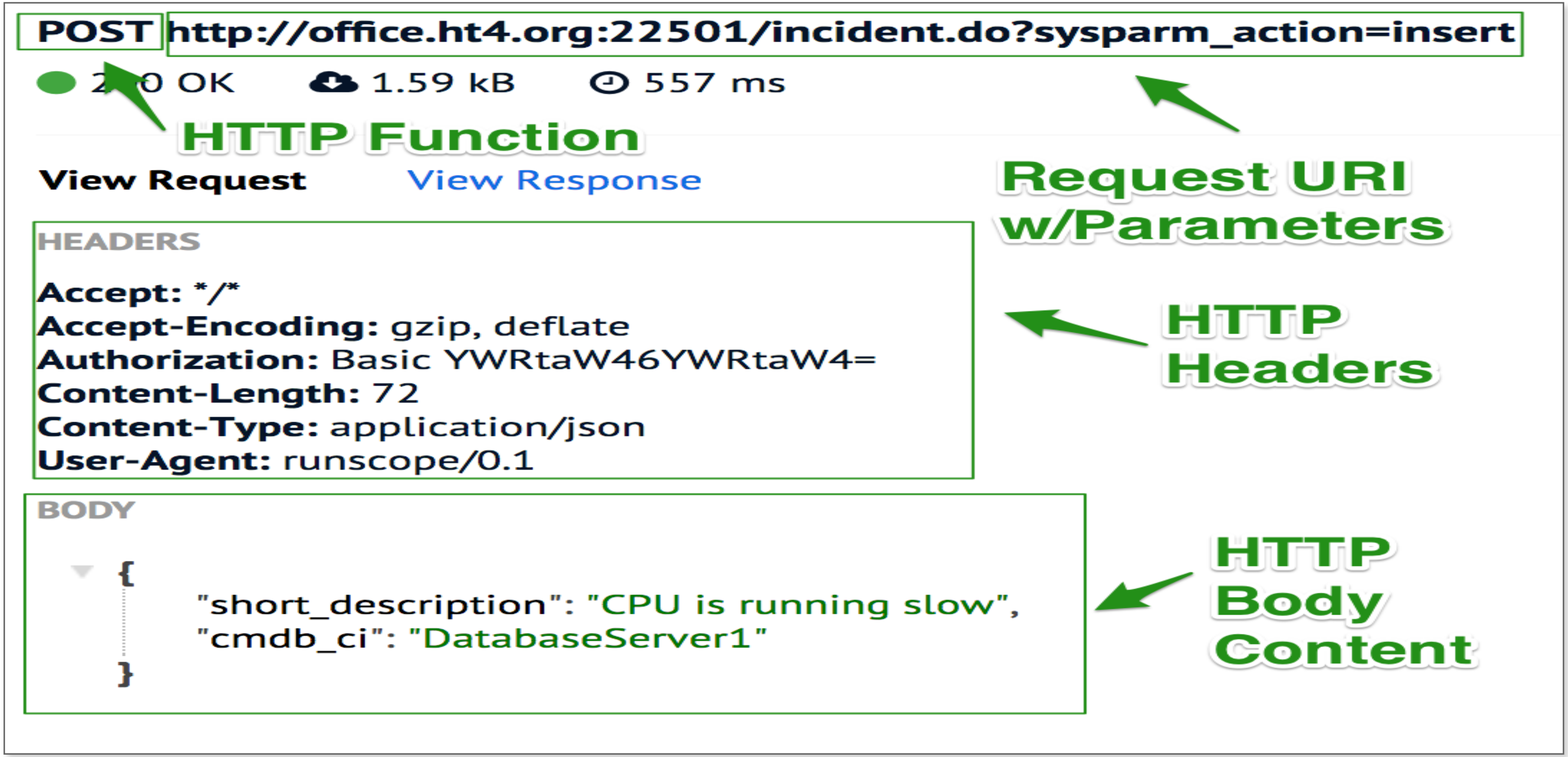
HTTP Response

Status Code

HTTP Headers

Content

Sample HTTP Request



```
POST http://office.ht4.org:22501/incident.do?sysparm_action=insert
```

● 200 OK 1.59 kB 557 ms

HTTP Function

Request URI w/Parameters

HEADERS

```
Accept: */*
Accept-Encoding: gzip, deflate
Authorization: Basic YWRtaW46YWRtaW4=
Content-Length: 72
Content-Type: application/json
User-Agent: runscope/0.1
```

HTTP Headers

BODY

```
{
    "short_description": "CPU is running slow",
    "cmdb_ci": "DatabaseServer1"
}
```

HTTP Body Content

Major REST Functions

- Most Common
  - POST
  - PUT
  - GET
  - DELETE

- Less Common
  - PATCH
  - HEAD
  - OPTIONS

Generally Accepted
REST Collections & Elements

- Collections

Lists or Groups

http://ht4.org/api/users

GET: Return list of users

POST: Creates a user record

PUT: Modifies a filtered list of records

DELETE: Deletes qualifying records

- Elements

Individual Items

http://ht4.org/api/users/john

GET: Returns user record

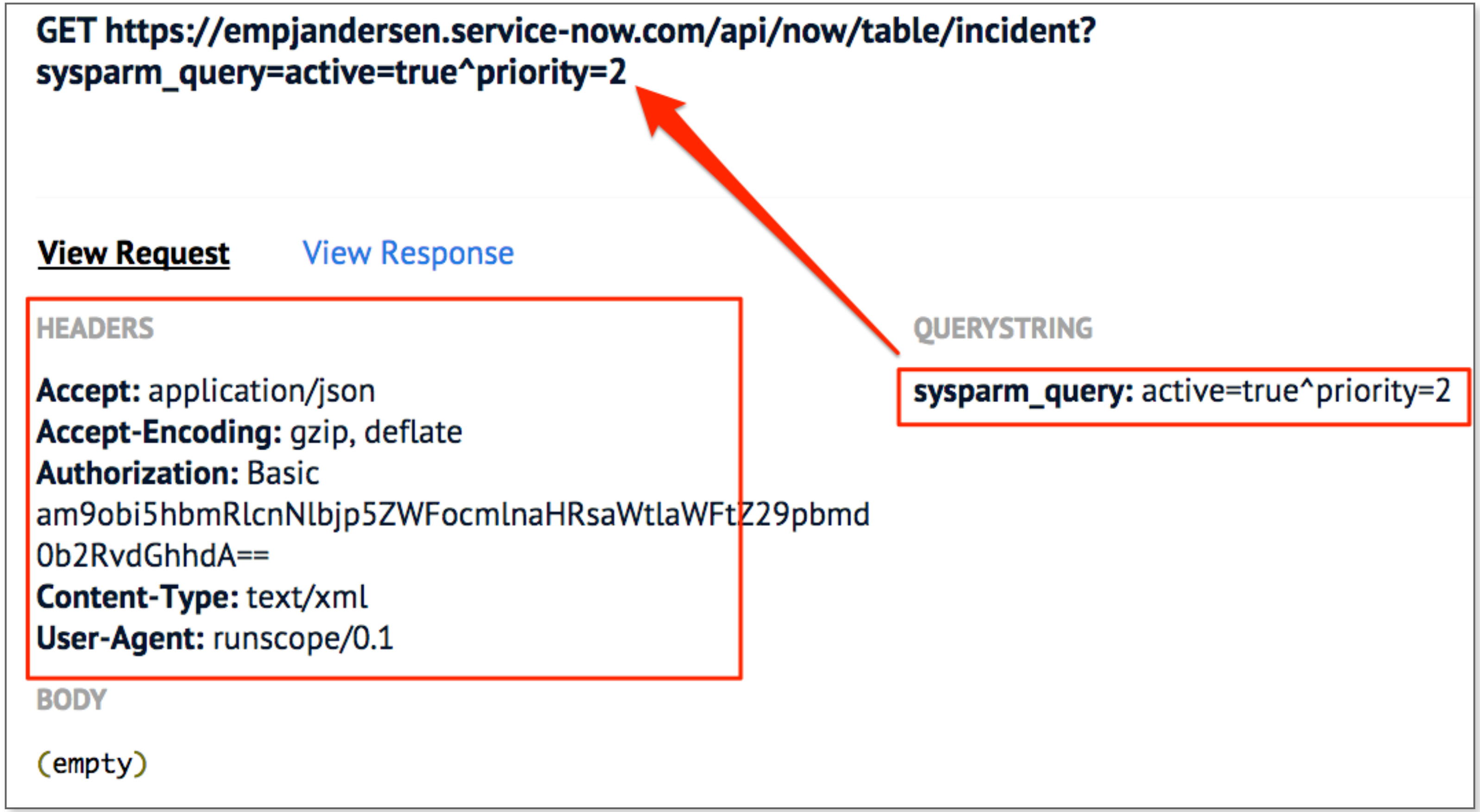
POST: N/A

PUT: Modifies single record

DELETE: Deletes record

HTTP Methods Used with REST

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



GET https://empjandersen.service-now.com/api/now/table/incident?sysparm_query=active=true^priority=2

HEADERS

Accept: application/json
Accept-Encoding: gzip, deflate
Authorization: Basic am9obi5hbmRlcnNlbjp5ZWFocmlnaHRsaWtlaWFtZ29pbmd0b2RvdGhhdA==
Content-Type: text/xml
User-Agent: runscope/0.1

QUERYSTRING

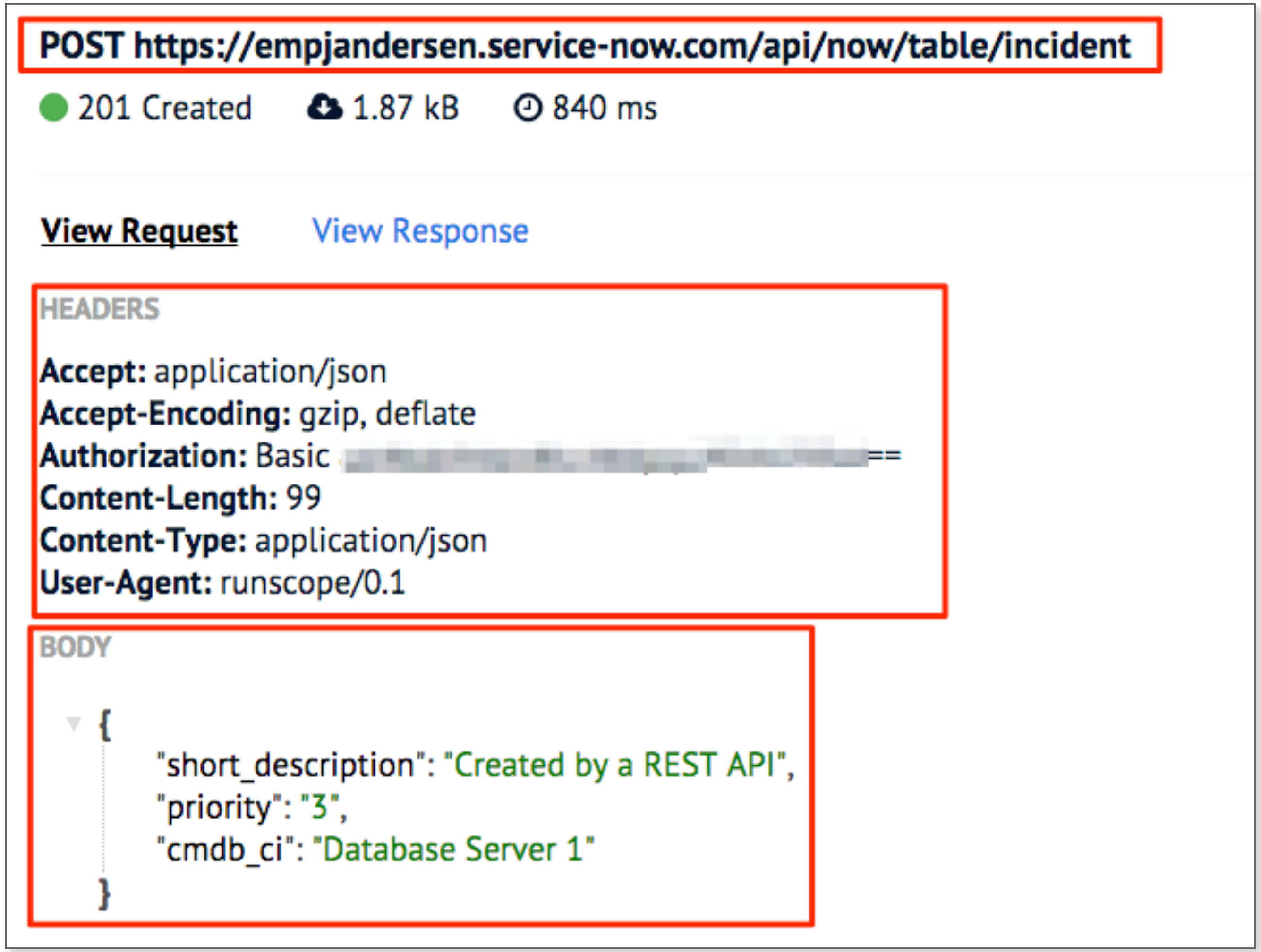
sysparm_query: active=true^priority=2

BODY

(empty)

HTTP Methods Used with REST

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



POST https://empjandersen.service-now.com/api/now/table/incident

● 201 Created   1.87 kB   840 ms

HEADERS

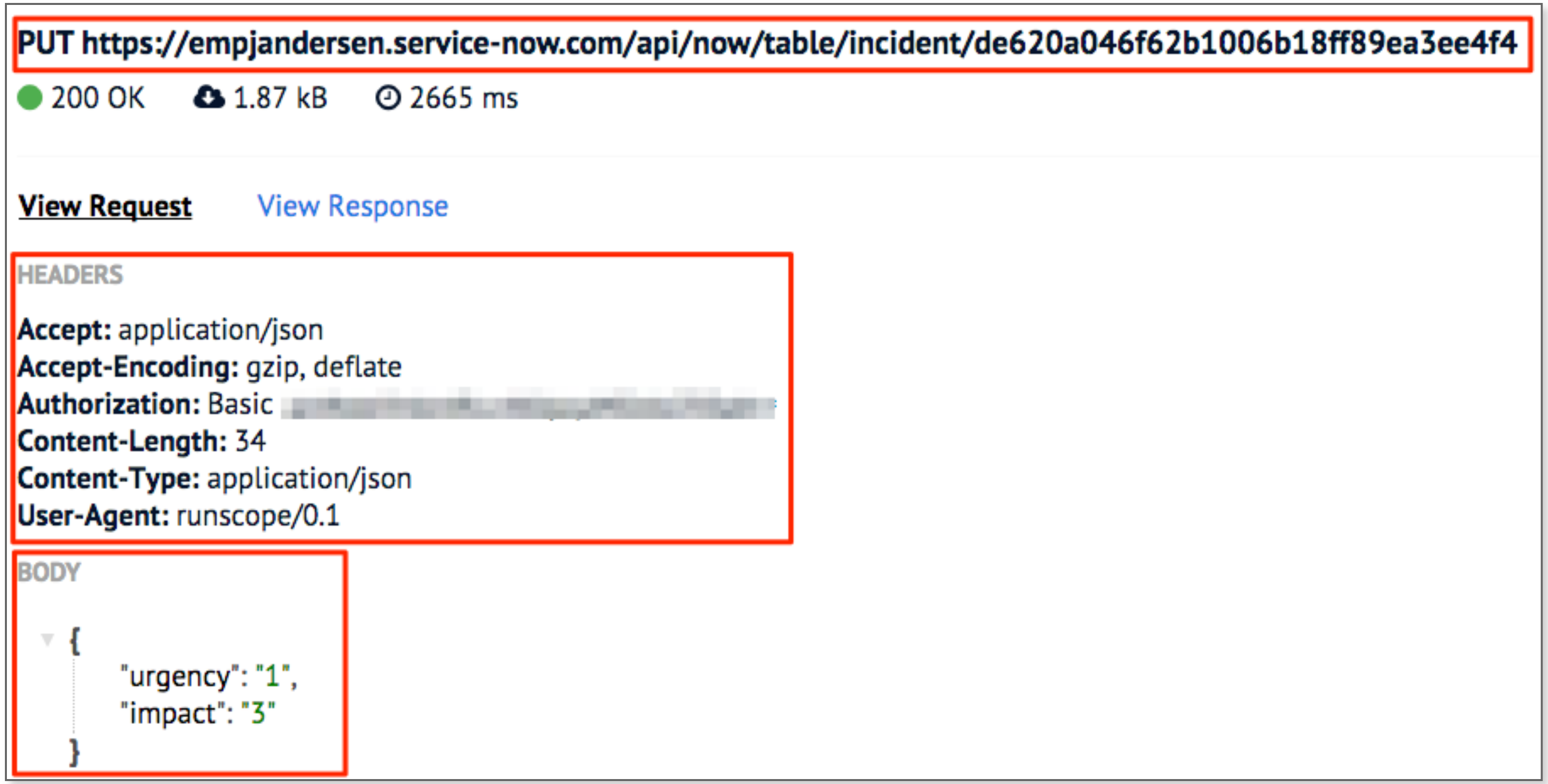
Accept: application/json
Accept-Encoding: gzip, deflate
Authorization: Basic [redacted]==
Content-Length: 99
Content-Type: application/json
User-Agent: runscope/0.1

BODY

```
{
    "short_description": "Created by a REST API",
    "priority": "3",
    "cmdb_ci": "Database Server 1"
}
```

HTTP Methods Used with REST

- PUT
  - For modifications
  - HTTP Components
    - Request URL & Parameters
    - HTTP Headers
    - Content



PUT https://empjandersen.service-now.com/api/now/table/incident/de620a046f62b1006b18ff89ea3ee4f4

HEADERS

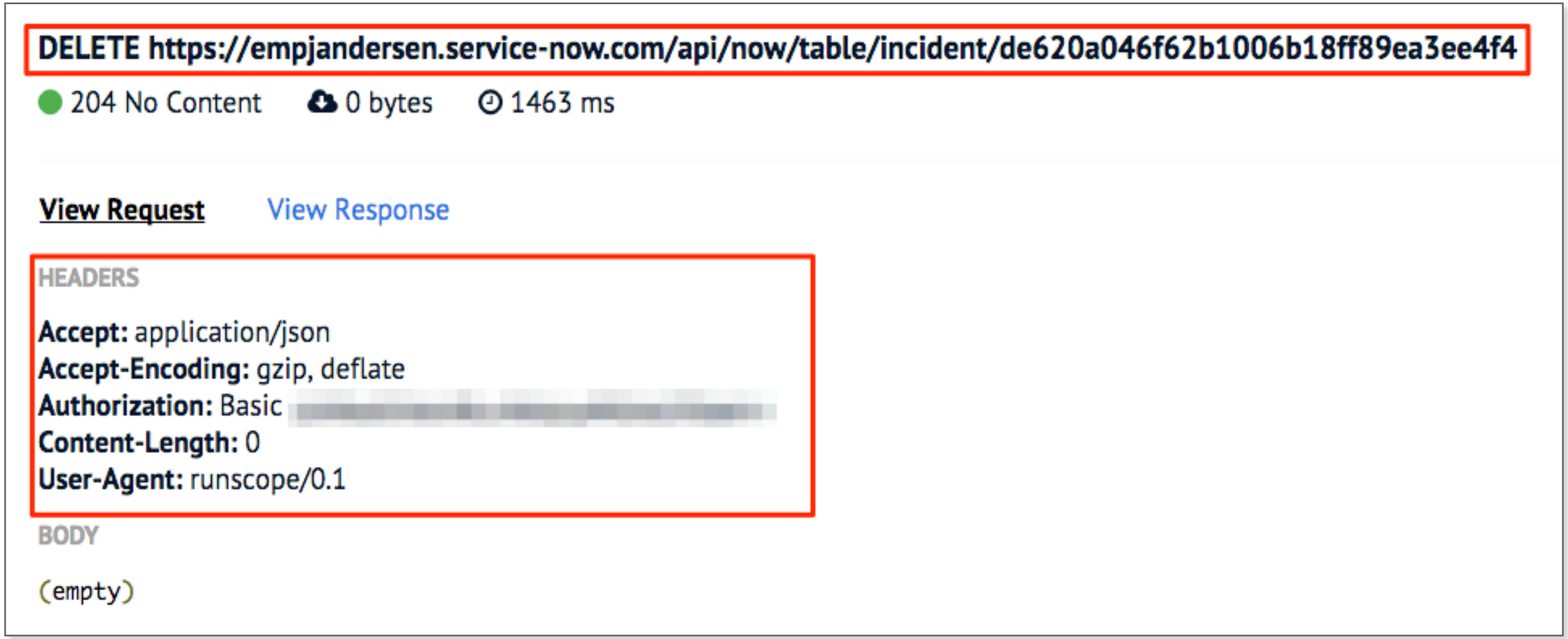
Accept: application/json
Accept-Encoding: gzip, deflate
Authorization: Basic [redacted]
Content-Length: 34
Content-Type: application/json
User-Agent: runscope/0.1

BODY

```
{
    "urgency": "1",
    "impact": "3"
}
```

HTTP Methods Used with REST

- DELETE
  - Delete a resource
  - HTTP Components
    - Request URL & Parameters
    - HTTP Headers



DELETE https://empjandersen.service-now.com/api/now/table/incident/de620a046f62b1006b18ff89ea3ee4f4

● 204 No Content   0 bytes   1463 ms

HEADERS

Accept: application/json
Accept-Encoding: gzip, deflate
Authorization: Basic [redacted]
Content-Length: 0
User-Agent: runscope/0.1

BODY

(empty)

Useful REST Client Tools

- REST Message Client in ServiceNow
- SOAP-UI (yes, it does REST)
- Curl (standard on any real Operating System)
- http://www.Hurl.it
- Browser based extensions
  - Advanced REST Client (Chrome)
  - RestClient.net (Firefox)

ServiceNow’s REST Table API

ServiceNow: A RESTful History Lesson

- JSON Web Service
  - Built in early 2010
  - Originally export processor
  - Optional plugin
  - Added functions through URL parameters
  - Inadvertently became the “REST API”
  - Slow performance on larger tables
  - Security concerns

ServiceNow: A RESTful History Lesson

- JSONv2 Web Service
  - Introduced in Dublin Release
  - Built in Java
  - Based off of JSON Web Service
  - Installed by default
  - Speed and Security Enhancements
  - Primarily same interface as JSON Web Service
  - Built to provide enhancements to current implementations

ServiceNow: A RESTful History Lesson

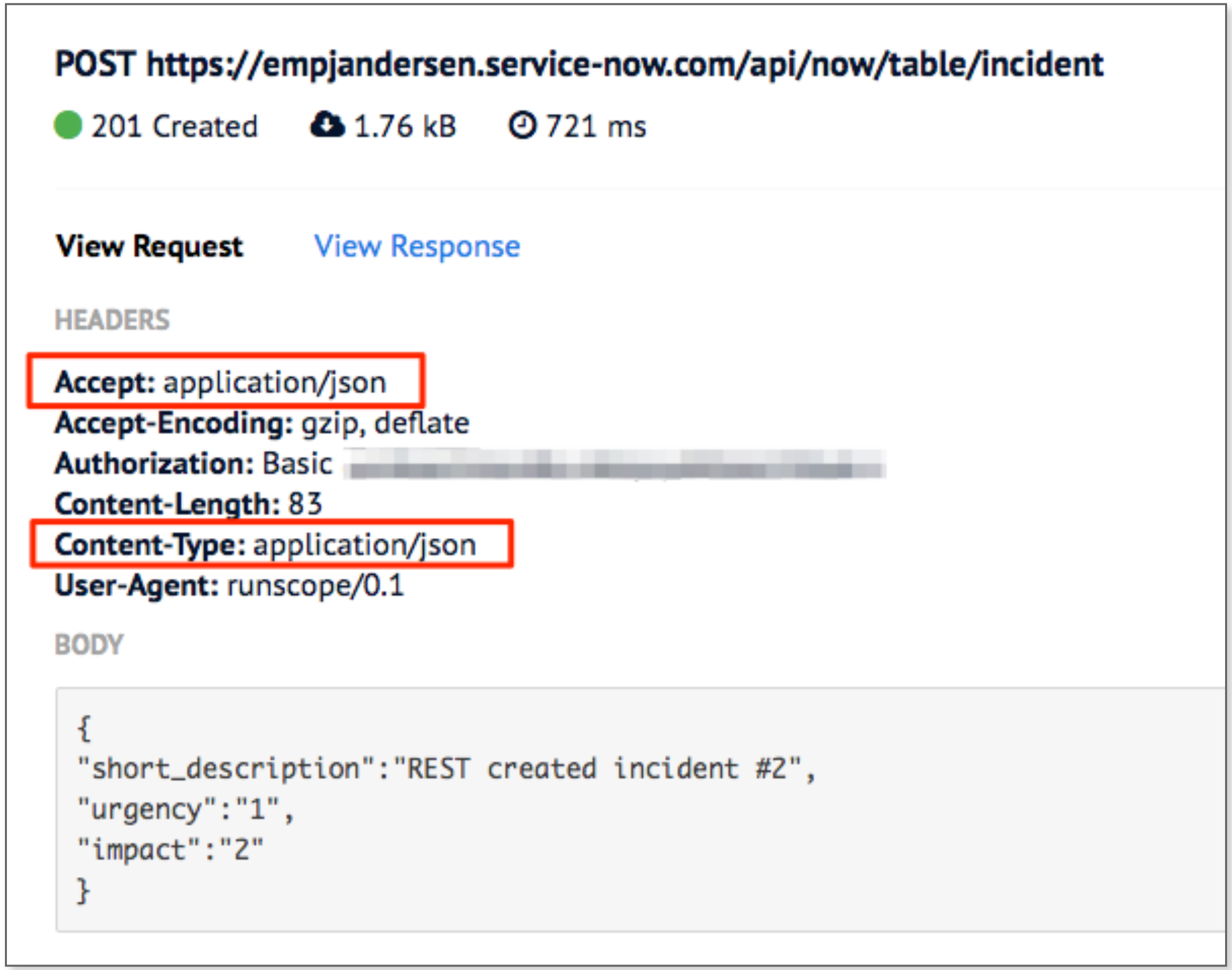
- Official REST API
  - Introduced in Eureka Release
  - Announced at Knowledge 14
  - Enabled by default
  - More traditional REST interface
  - Requires a “rest_service” role

Supported HTTP Methods

- GET - Query
- POST – Insert/Create
- PUT - Modify
- PATCH - Modify
- DELETE - Delete

Supported HTTP Headers

- Accept
  - Format we want back
  - text/xml (XML), application/json (JSON)
- Content-Type
  - Format of data we are sending
  - text/xml (XML), application/json (JSON)



POST https://empjandersen.service-now.com/api/now/table/incident

● 201 Created 1.76 kB 721 ms

HEADERS

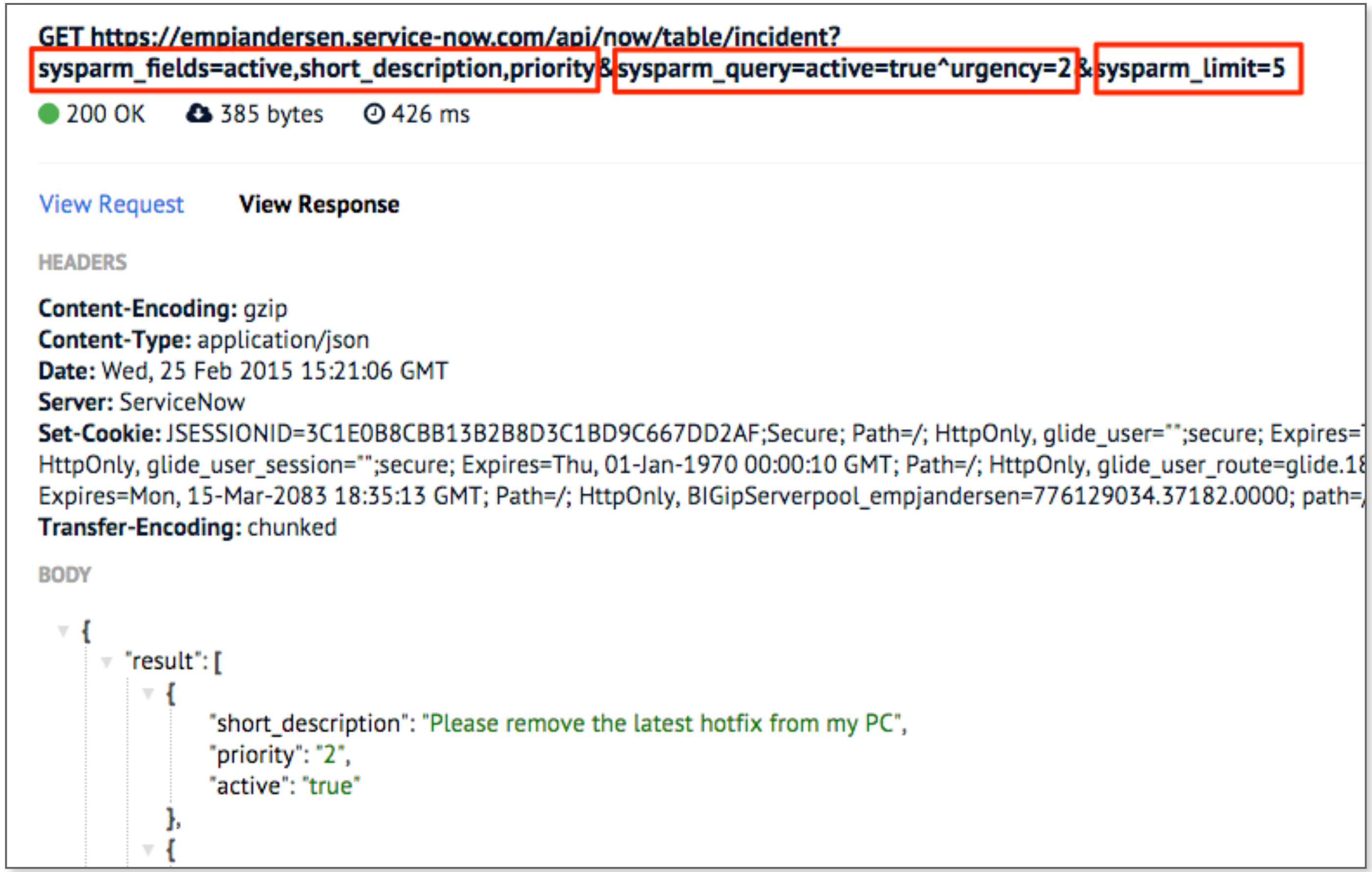
Accept: application/json
Accept-Encoding: gzip, deflate
Authorization: Basic [redacted]
Content-Length: 83
Content-Type: application/json
User-Agent: runscope/0.1

BODY

```
{
"short_description":"REST created incident #2",
"urgency":"1",
"impact":"2"
}
```

Supported URL/Query Parameters

- sysparm_query
- sysparm_display_value
- sysparm_fields
- sysparm_view
- sysparm_limit
- sysparm_exclude_reference_link
- sysparm_read_replica_category
- sysparm_input_display_value
- Key-Value pairs (eg. &active=true)



GET https://empjandersen.service-now.com/api/now/table/incident?sysparm_fields=active,short_description,priority&sysparm_query=active=true^urgency=2&sysparm_limit=5

● 200 OK   385 bytes   426 ms

HEADERS

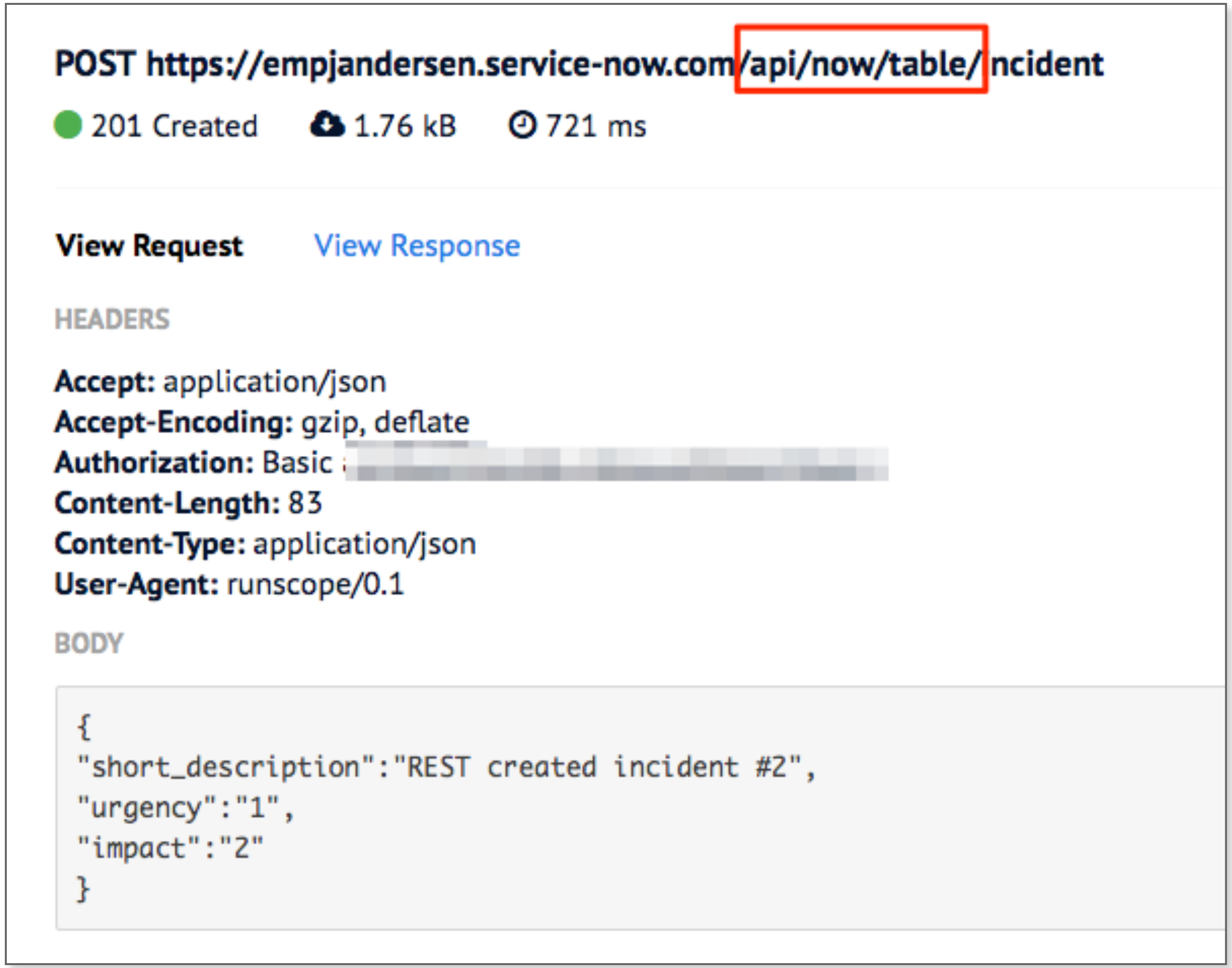
Content-Encoding: gzip
Content-Type: application/json
Date: Wed, 25 Feb 2015 15:21:06 GMT
Server: ServiceNow
Set-Cookie: JSESSIONID=3C1E0B8CBB13B2B8D3C1BD9C667DD2AF;Secure; Path=/; HttpOnly, glide_user="";secure; Expires=
HttpOnly, glide_user_session="";secure; Expires=Thu, 01-Jan-1970 00:00:10 GMT; Path=/; HttpOnly, glide_user_route=glide.1
Expires=Mon, 15-Mar-2083 18:35:13 GMT; Path=/; HttpOnly, BIGipServerpool_empjandersen=776129034.37182.0000; path=
Transfer-Encoding: chunked

BODY

```
{
  "result": [
    {
      "short_description": "Please remove the latest hotfix from my PC",
      "priority": "2",
      "active": "true"
    },
    {
```

Built in API Versioning

- URL Based Versioning
  - /api/now/v1/table/incident
- Reference the latest version
  - /api/now/table/incident



POST https://empjandersen.service-now.com/api/now/table/incident

● 201 Created 1.76 kB 721 ms

HEADERS

Accept: application/json
Accept-Encoding: gzip, deflate
Authorization: Basic [redacted]
Content-Length: 83
Content-Type: application/json
User-Agent: runscope/0.1

BODY

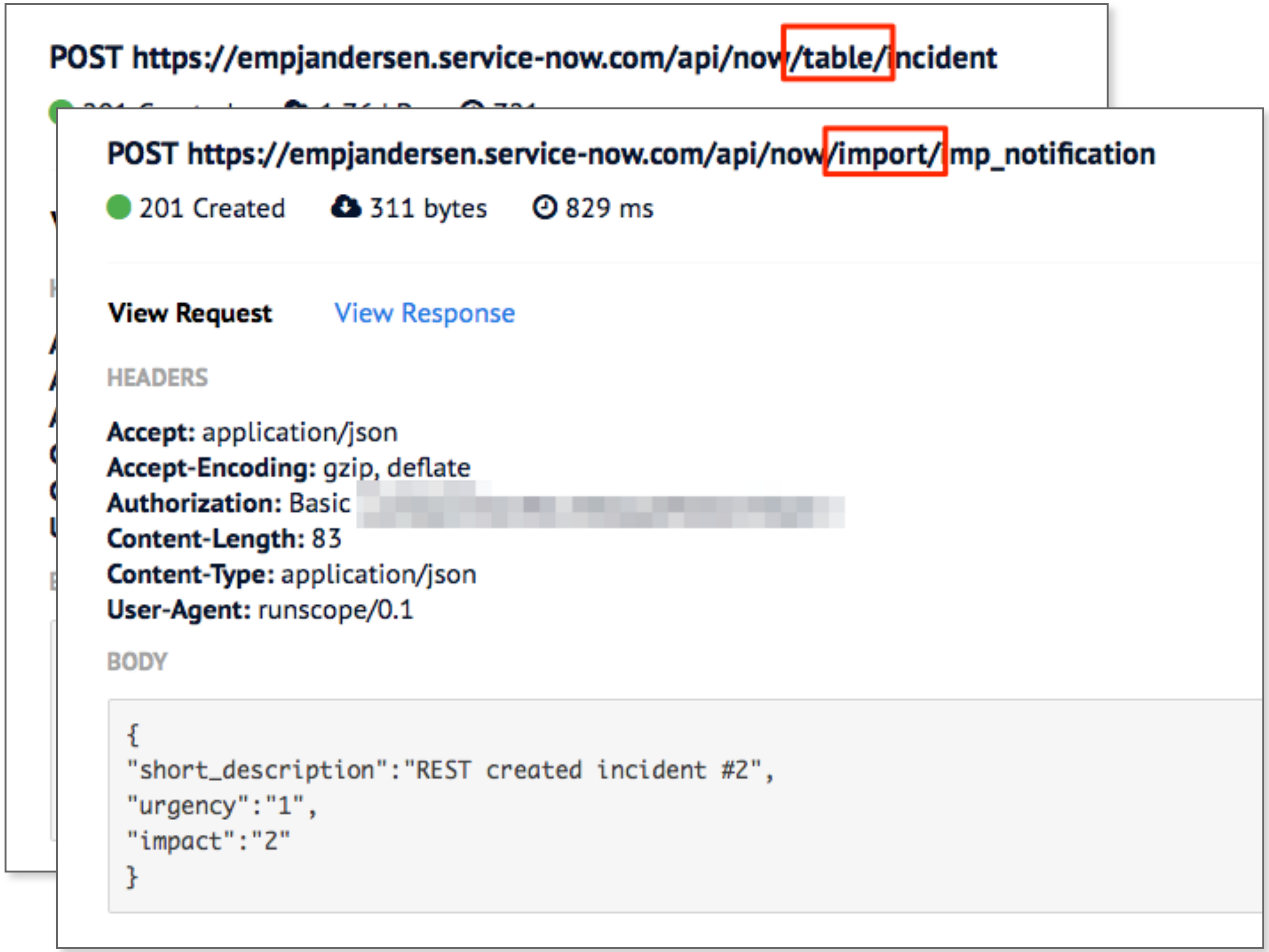
```
{
"short_description":"REST created incident #2",
"urgency":"1",
"impact":"2"
}
```

Misc Features & Requirements

- Security
  - User ACLs still apply
  - rest_service role is required for authenticating user
- Debug Option
  - Inbound requests can be written to the system log
  - “glide.rest.debug = true” – performance warning
- Descriptive HTTP response codes and messages
  - 8 supported codes with multiple possible messages per code

Two Main APIs

- Table
  - Direct Table Web Service API
  - /api/now/table/{table_name}
  - POST, GET, PUT, PATCH, DELETE
- Import
  - Web Service Import Set support
  - /api/now/import/{ws_import_set}
  - POST, GET



POST https://empjandersen.service-now.com/api/now/table/incident

POST https://empjandersen.service-now.com/api/now/import/imp_notification

● 201 Created 311 bytes 829 ms

View Request View Response

HEADERS

Accept: application/json
Accept-Encoding: gzip, deflate
Authorization: Basic [redacted]
Content-Length: 83
Content-Type: application/json
User-Agent: runscope/0.1

BODY

```
{
"short_description":"REST created incident #2",
"urgency":"1",
"impact":"2"
}
```

Table API Method Summary

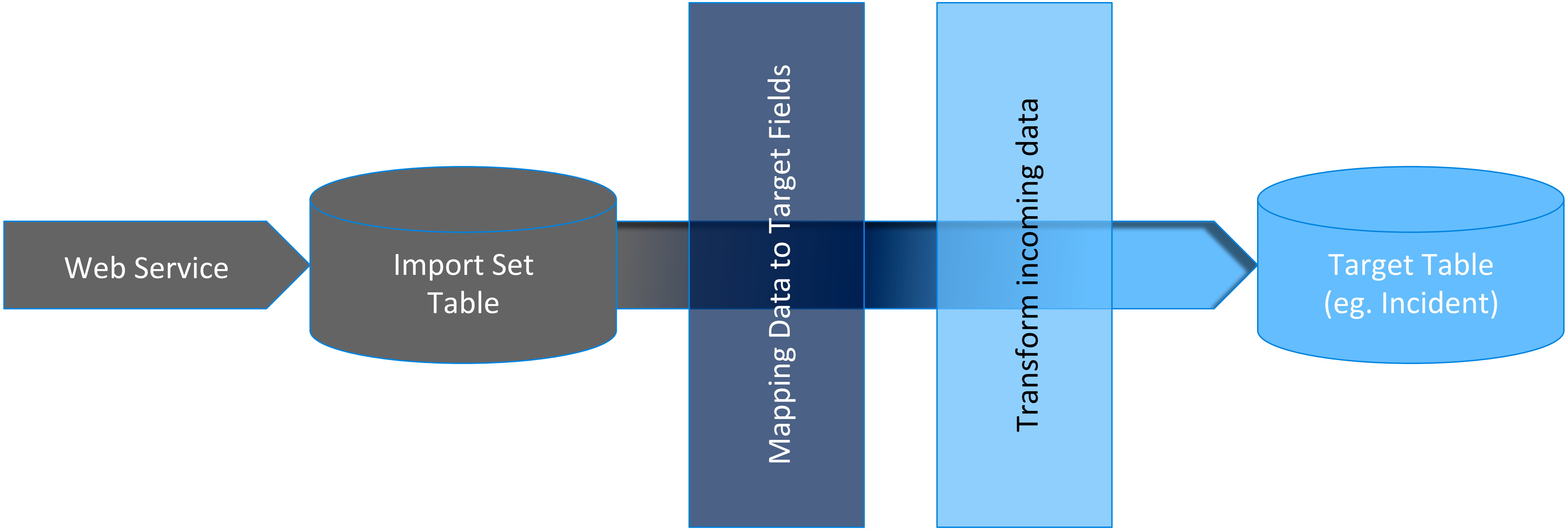
- **GET**
  - /api/now/table/{table_name} - **gets filtered list of records**
  - /api/now/table/{table_name}/{sys_id} – **single record**
- **PUT**
  - /api/now/table/{table_name}/{sys_id} – **updates specific record**
- **POST**
  - /api/now/table/{table_name} – **creates record in the table**
- **DELETE**
  - /api/now/table/{table_name}/{sys_id} – **deletes specific record**

Labs 2-5

REST Web Service Table API Exercises

ServiceNow’s REST Import API

Web Service Import Set Flow

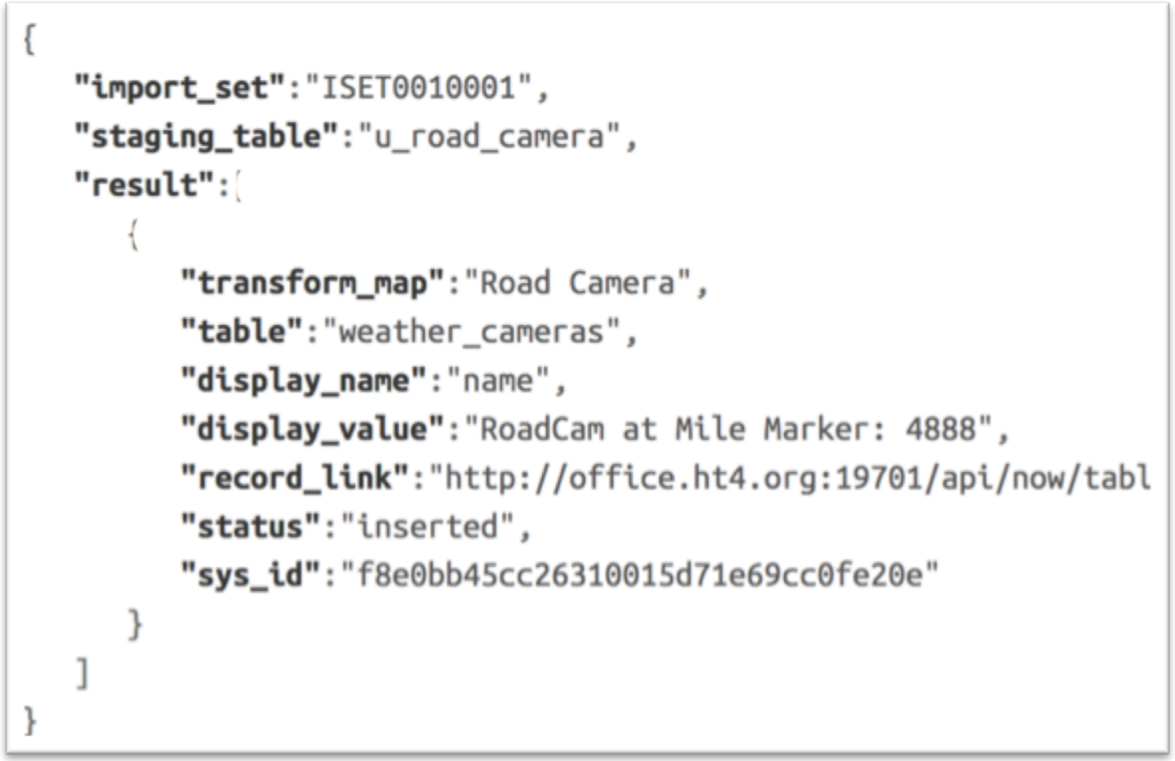


Import Web Services

- Fancy UI rendering of a standard Import Set
- Enable scripting and mapping around incoming data
- Can view raw data that came in
- Can decide whether or not to accept the data
- Can blank out fields
- Automatically determine if CREATE or UPDATE

Web Service Import Sets

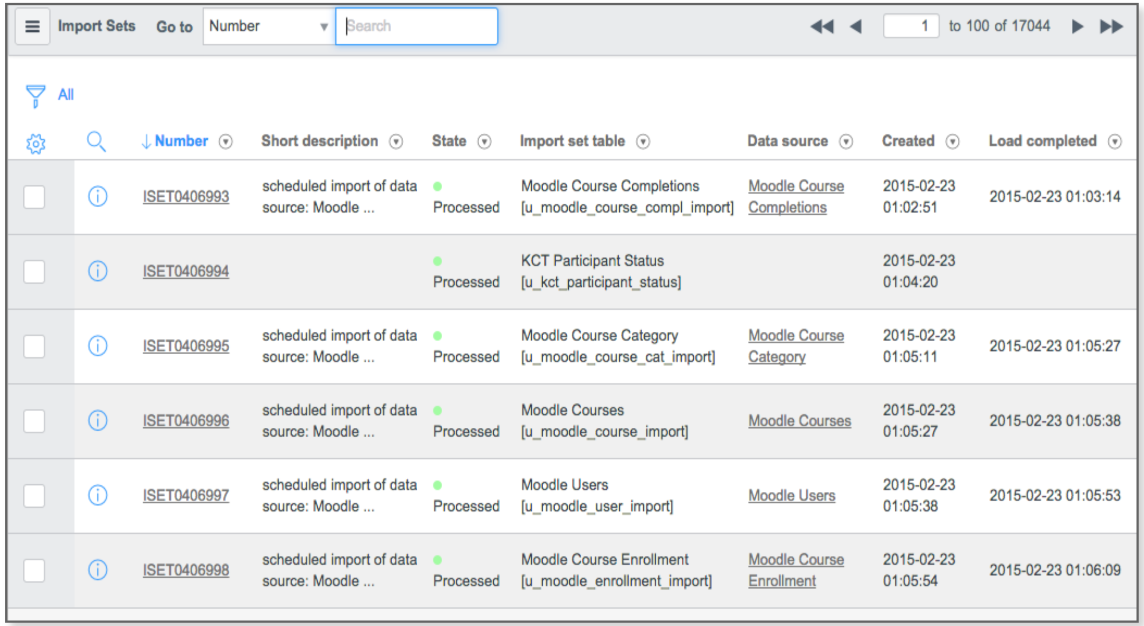
- Transform results returned to third party
  - Import action that took place (eg. Insert)
  - Table where the record ended up
  - Sys_id of the new record
  - Display value of the new record
  - Can send additional data if desired
- Recommend using Correlation ID & Display



```
{
   "import_set":"ISET0010001",
   "staging_table":"u_road_camera",
   "result":[
      {
         "transform_map":"Road Camera",
         "table":"weather_cameras",
         "display_name":"name",
         "display_value":"RoadCam at Mile Marker: 4888",
         "record_link":"http://office.ht4.org:19701/api/now/tabl
         "status":"inserted",
         "sys_id":"f8e0bb45cc26310015d71e69cc0fe20e"
      }
   ]
}
```

Operations Available

- Really just another table
- All direct web service functions available
  - SOAP
  - REST Table API
- Primary Interfaces
  - SOAP: Insert
  - REST: Post, Get



Number	Short description	State	Import set table	Data source	Created	Load completed
ISET0406993	scheduled import of data source: Moodle ...	● Processed	Moodle Course Completions [u_moodle_course_compl_import]	Moodle Course Completions	2015-02-23 01:02:51	2015-02-23 01:03:14
ISET0406994		● Processed	KCT Participant Status [u_kct_participant_status]		2015-02-23 01:04:20	
ISET0406995	scheduled import of data source: Moodle ...	● Processed	Moodle Course Category [u_moodle_course_cat_import]	Moodle Course Category	2015-02-23 01:05:11	2015-02-23 01:05:27
ISET0406996	scheduled import of data source: Moodle ...	● Processed	Moodle Courses [u_moodle_course_import]	Moodle Courses	2015-02-23 01:05:27	2015-02-23 01:05:38
ISET0406997	scheduled import of data source: Moodle ...	● Processed	Moodle Users [u_moodle_user_import]	Moodle Users	2015-02-23 01:05:38	2015-02-23 01:05:53
ISET0406998	scheduled import of data source: Moodle ...	● Processed	Moodle Course Enrollment [u_moodle_enrollment_import]	Moodle Course Enrollment	2015-02-23 01:05:54	2015-02-23 01:06:09

REST Import API Method Summary

- **GET**
  - /api/now/import/{ws_name}/{sys_id} – **import set record info**
- **POST**
  - /api/now/import/{ws_name} – **submits record to import set**

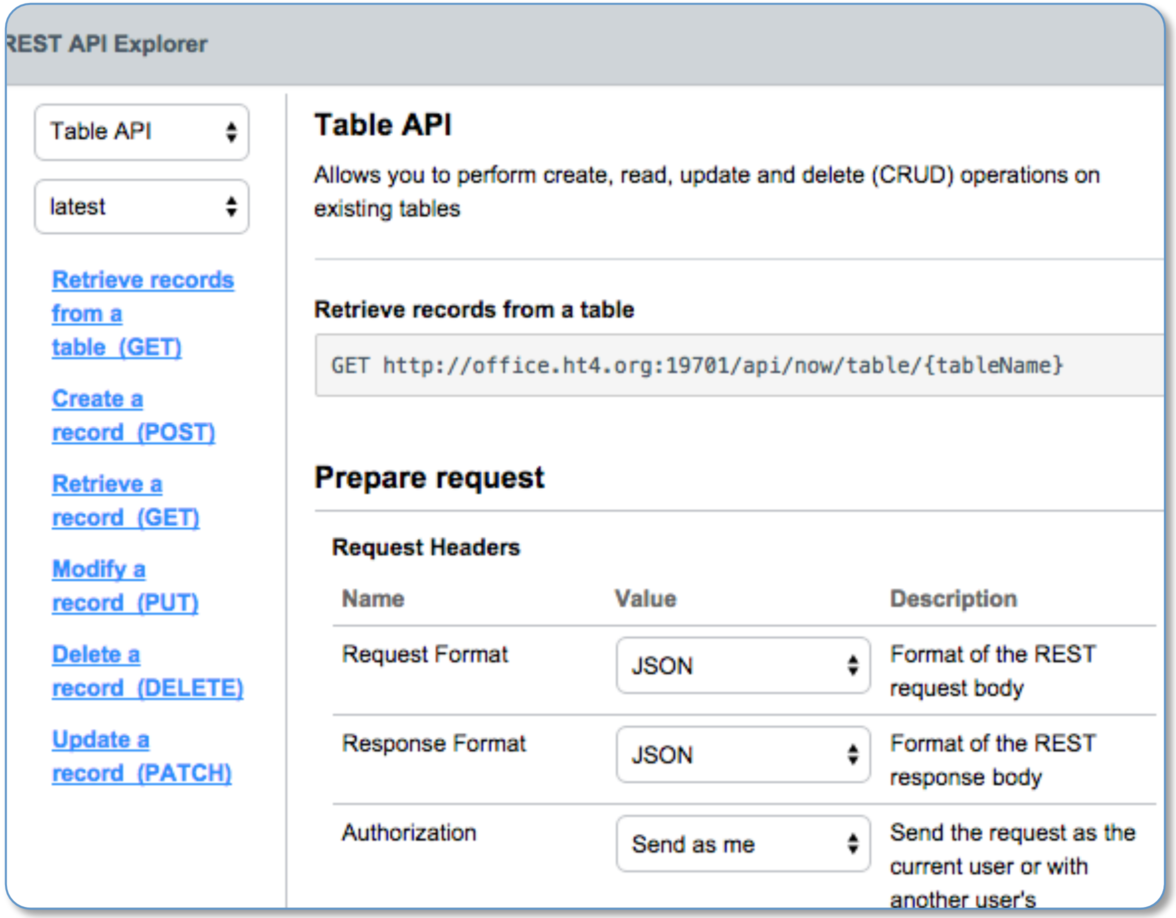
Lab 6

REST Web Service Import API Exercises

REST API Explorer

REST API Explorer

- Covers all supported REST APIs within the platform
- Test interfaces in real time
- Auto generate script examples
- Available in the Fuji release



REST API Explorer

Table API

latest

Retrieve records from a table (GET)

Create a record (POST)

Retrieve a record (GET)

Modify a record (PUT)

Delete a record (DELETE)

Update a record (PATCH)

Table API

Allows you to perform create, read, update and delete (CRUD) operations on existing tables

Retrieve records from a table

```
GET http://office.ht4.org:19701/api/now/table/{tableName}
```

Prepare request

Request Headers

Name	Value	Description
Request Format	JSON	Format of the REST request body
Response Format	JSON	Format of the REST response body
Authorization	Send as me	Send the request as the current user or with another user's

Quick Demo

REST API Explorer

Other Web Service Interfaces

JSON Web Service

- Commonly used as ServiceNow's REST API
- Two Versions
  - JSON (Deprecated and requires "JSON Web Service plugin")
  - JSONv2 (Available since Calgary Patch 2)
- CRUD operations on all tables
- Future development focus will be on the official REST API

CSV Web Service

- URL Parameter: CSV
- Formatted Data
- Large data exports
- Utilize views to control exported fields

XML Web Service

- URL Parameter: XML
- Raw data
- Semi-large exports
- Paging capabilities
- Can show Display Values (optional)

Other Data Export Web Services

- Excel Web Service
- PDF Web Service
- RSS Web Service

AttachmentCreator Web Service

- Third parties can send files via SOAP, REST
- We attach them to a specified record
- Insert only
- Wiki Search String: AttachmentCreator

Lab 7-9

Misc Web Services

Top Takeaways

1 Tables vs. Import Sets for API access

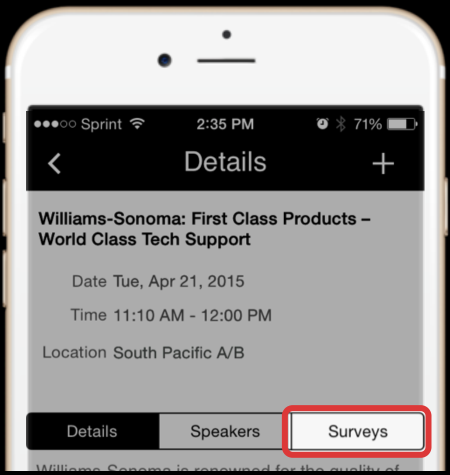
2 SOAP & REST APIs provide complete access to instance data

3 Export based web services allow for fast, easy pull of data

How Did We Do?

Your feedback on this session helps us deliver great content.

Please take a moment to complete a session survey in the Knowledge15 app or use the survey forms at the back of the room.



●●●○○ Sprint 2:35 PM 71%

Details

Williams-Sonoma: First Class Products – World Class Tech Support

Date Tue, Apr 21, 2015

Time 11:10 AM - 12:00 PM

Location South Pacific A/B

Details | Speakers | Surveys

Williams-Sonoma is renowned for the quality of

Everything as a Service

Thank You

John Andersen

Sr. Solution Architect

ServiceNow

john.andersen@servicenow.com

Chris Collazo

Solution Architect

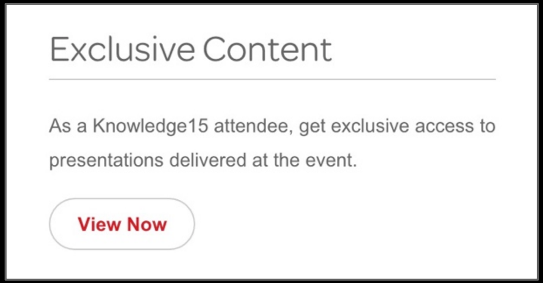
ServiceNow

chris.collazo@servicenow.com

Get Presentations

As a Knowledge15 attendee, you have exclusive access to breakout and lab session content from the event.

1. Go to knowledge.servicenow.com
2. Log into the community
3. Click on **View Now** button



Exclusive Content

As a Knowledge15 attendee, get exclusive access to presentations delivered at the event.

View Now

knowledge.servicenow.com

@arbonboy | #Know15