REVATURE

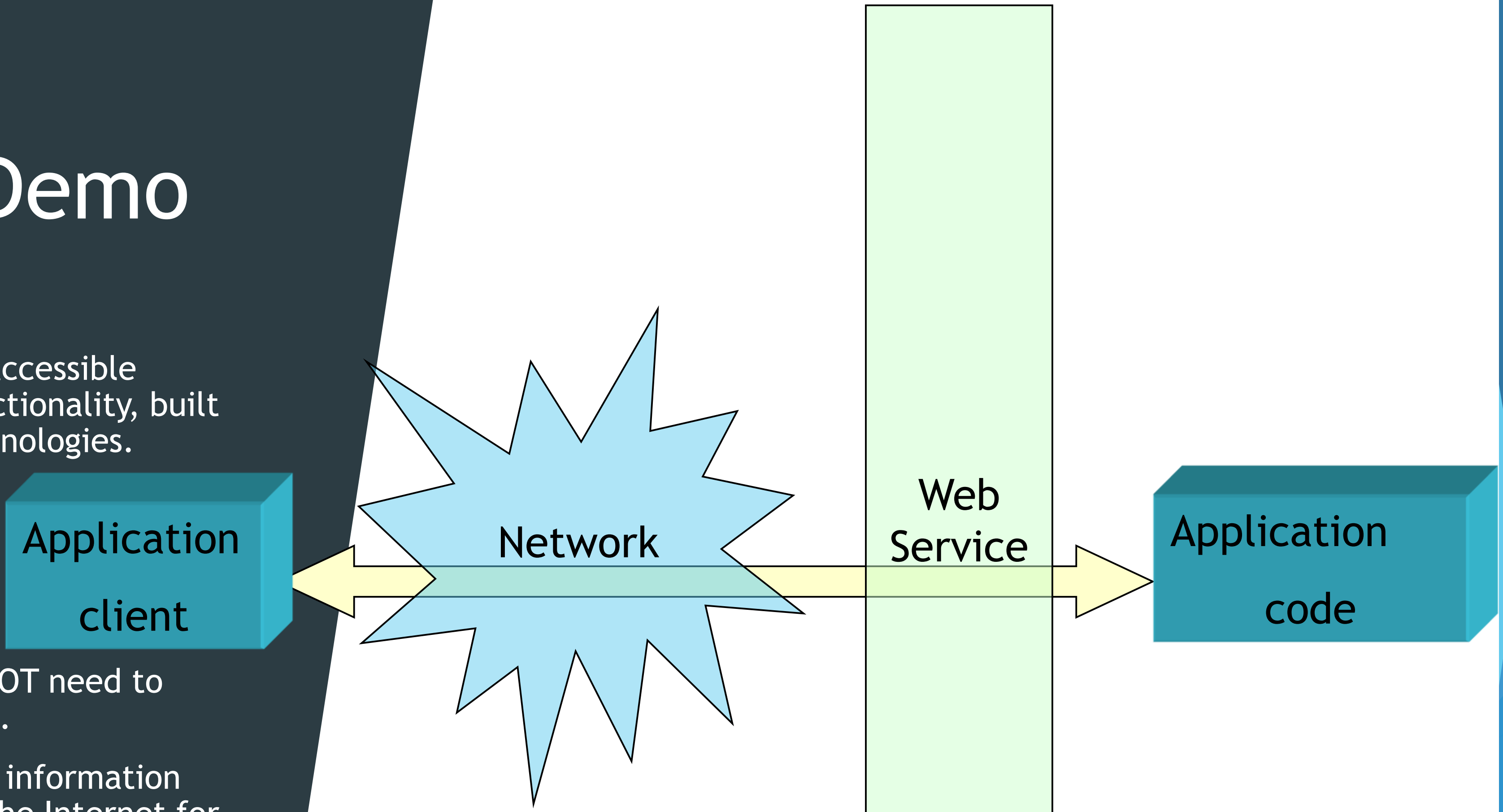
SERVICENOW OVERVIEW

REVATURE

ServiceNow – Webservices

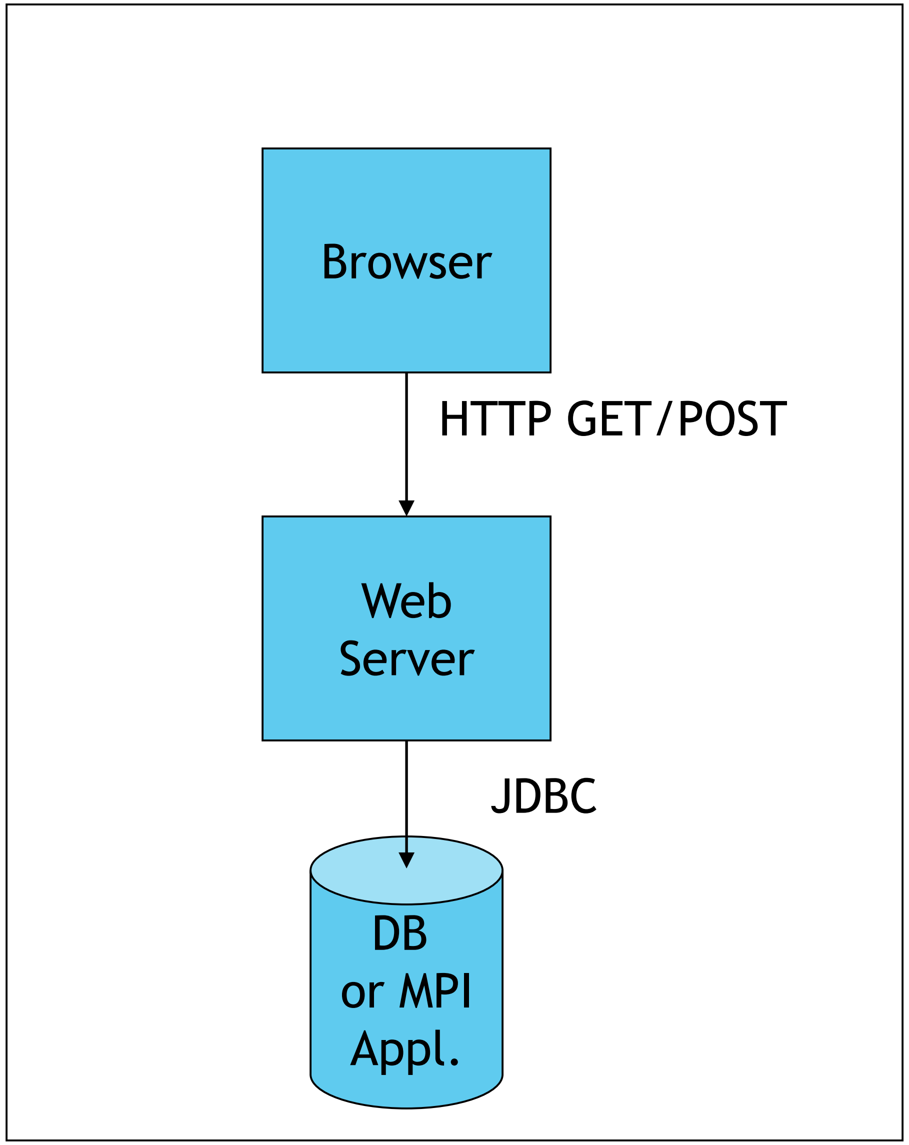
Webservice Demo

- A web service is a network accessible interface to application functionality, built using standard Internet technologies.
- Clients of web services do NOT need to know how it is implemented.
- Web services are XML-based information exchange systems that use the Internet for direct application-to-application interaction. These systems can include programs, objects, messages, or documents. A web service is a collection of open protocols and standards used for exchanging data between applications or systems.



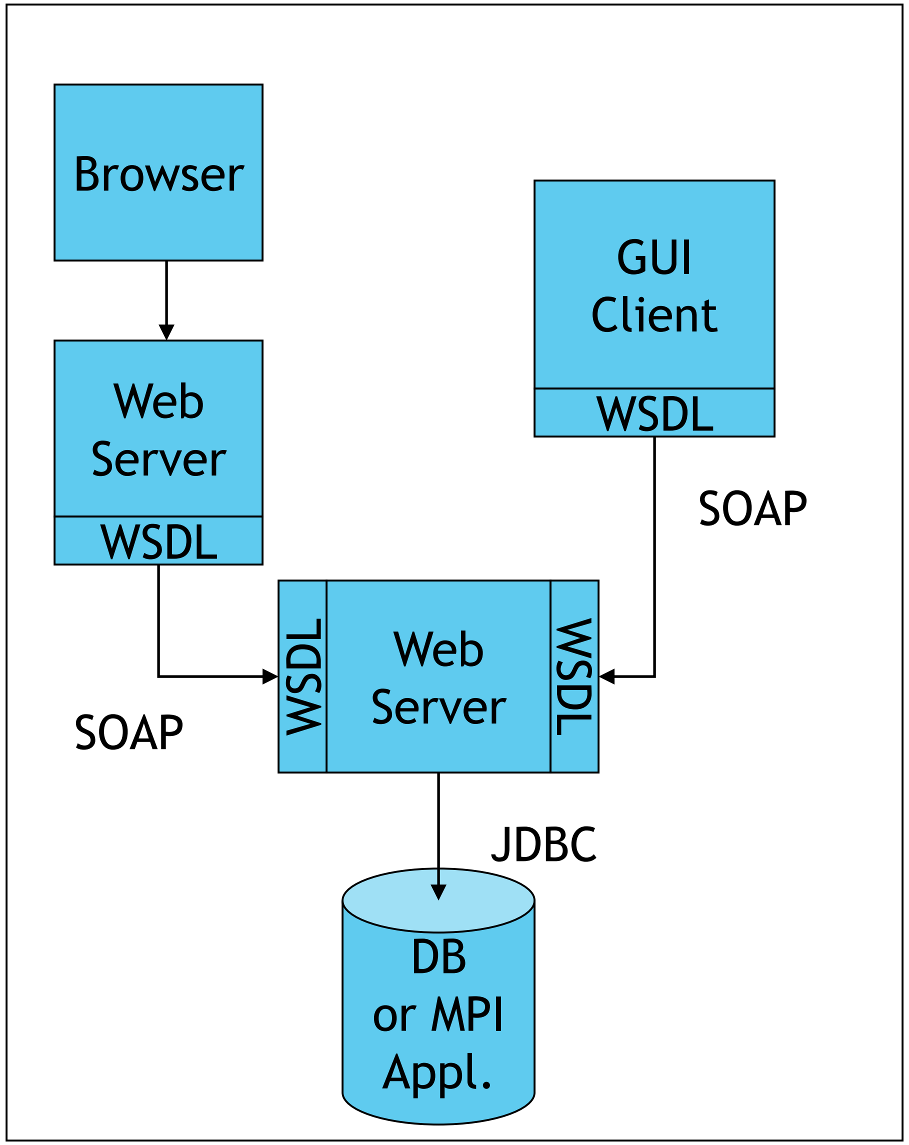
Webservice Demo

- Client/server system - Standard web application.
  - Browsers converse with web servers using HTTP GET/POST methods.
  - Servlets or CGI scripts process the parameters and take action, like connect to a DB.
  - Examples: Google, Amazon



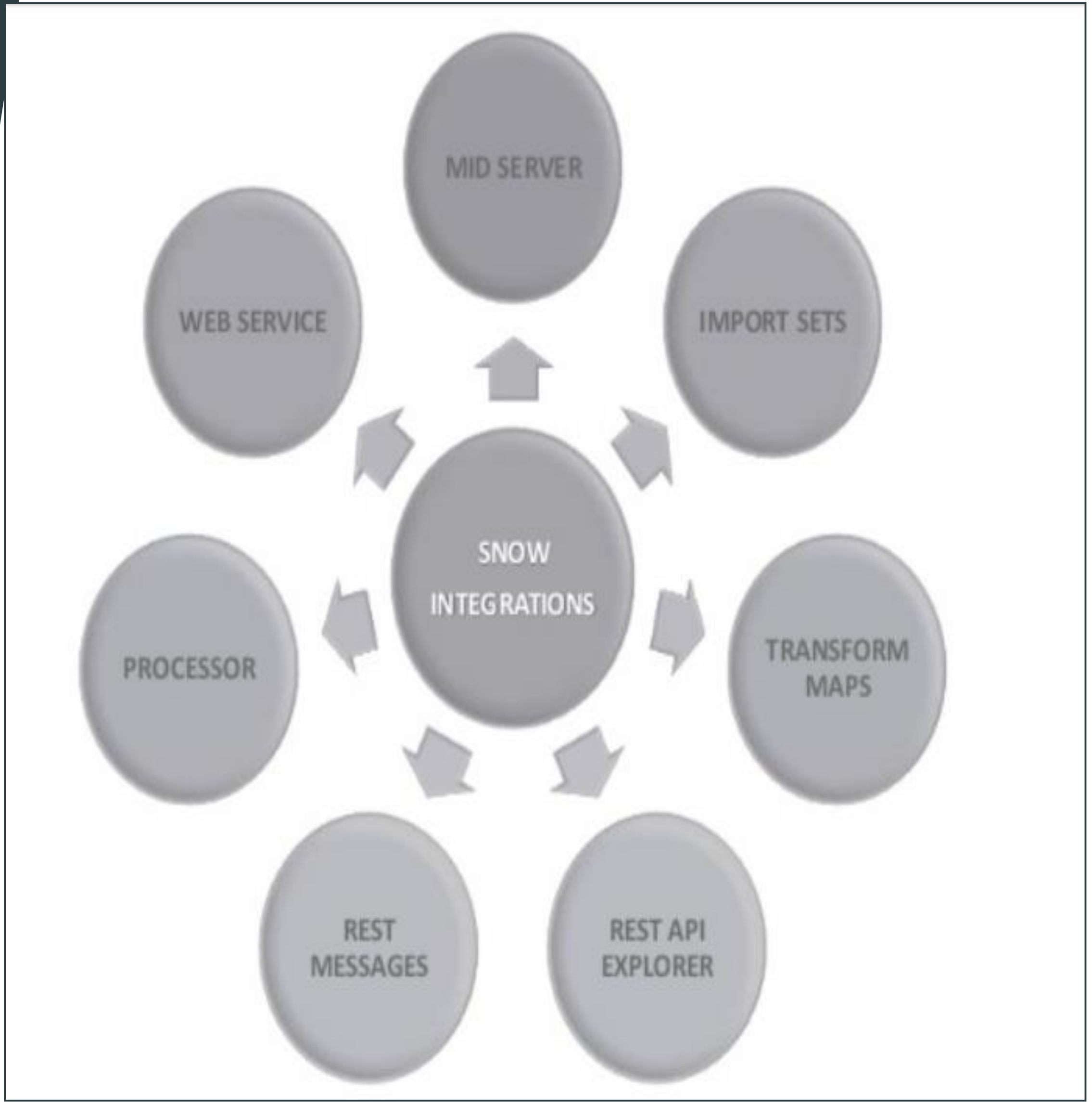
Webservice Demo

- Multi-tiered architecture - Web services system.
  - Interactions may be either through the browser or through a desktop client (ServiceNow,Java Swing, Python, Windows, etc.)
  - Examples: Google, Amazon



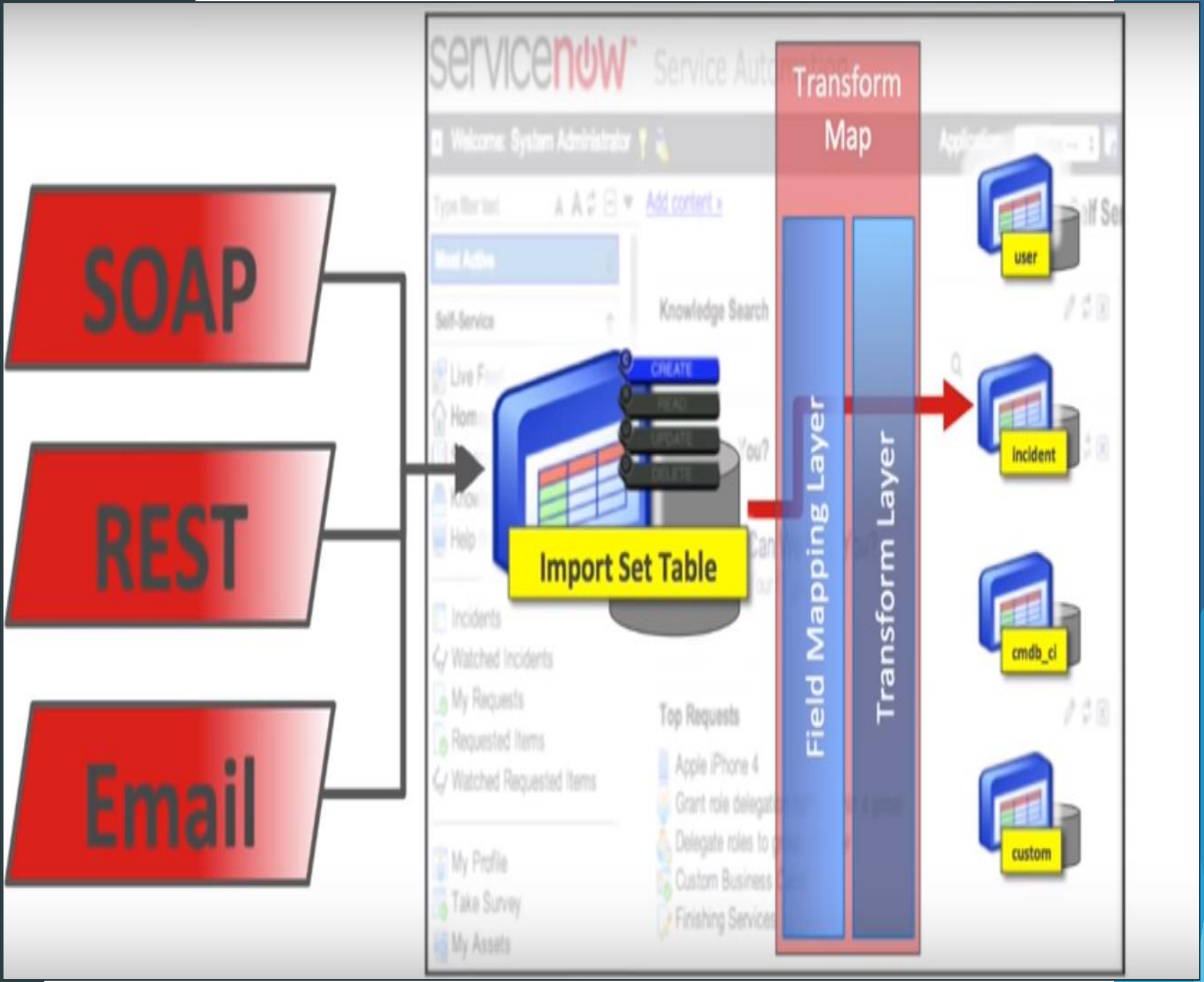
Webservice Demo

- ServiceNow supports standard protocols such as SOAP and REST, which can be understandable by any other technology(C#, Java, Python) and third party applications such as Remedy, HP CSA and JIRA etc. So technology agnostic API facilitates user to integrate their applications seamlessly with ServiceNow.



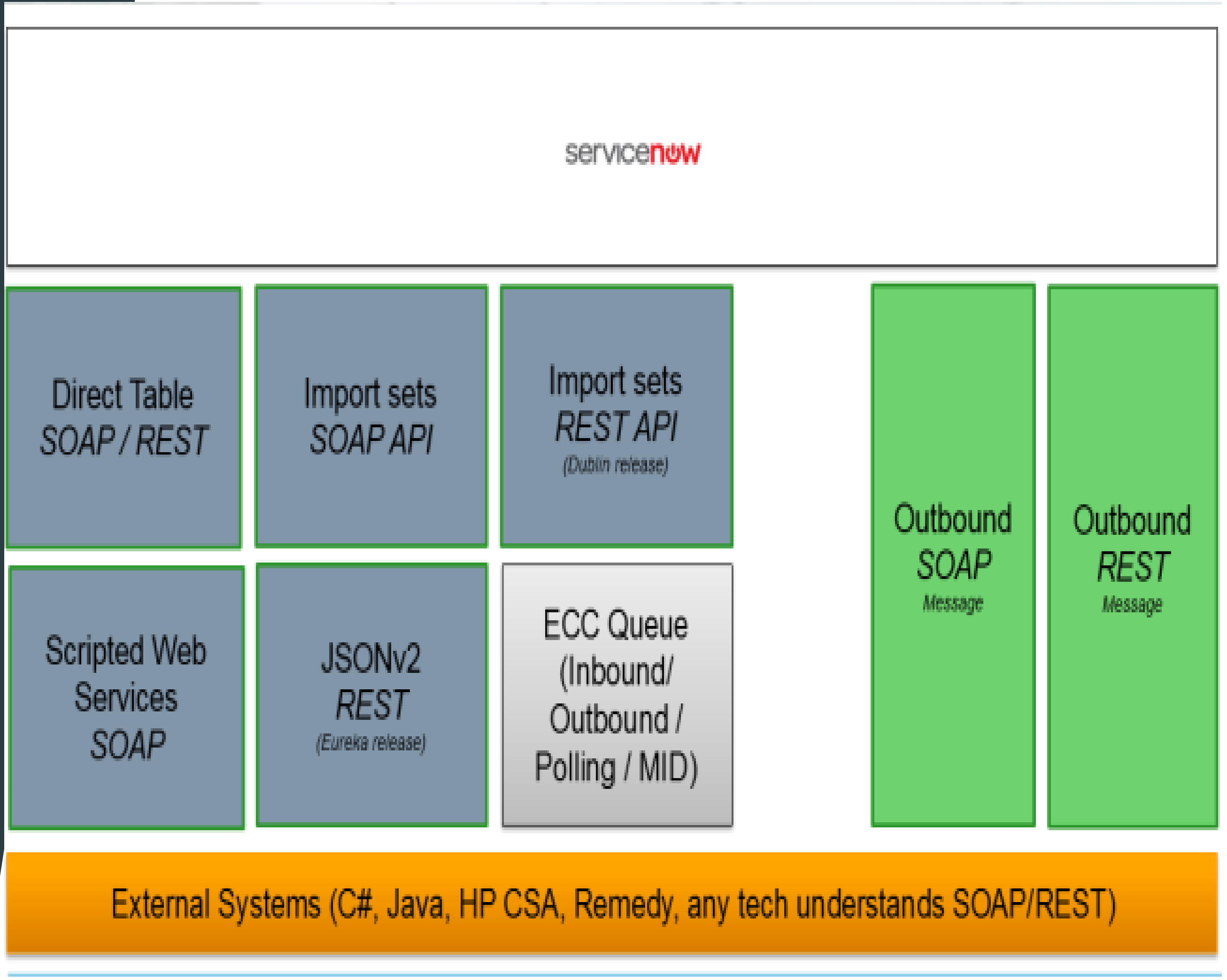
Webservice Demo

- HTTP-based web services allow diverse applications to talk to each other. ServiceNow supports both inbound (provider) and outbound (consumer) web services.
- Inbound webservices – Third party system is quering SNOW tables i.e getting resources from SNOW
- Outbound webservices – SNOW queries third party tables or databases and gets information from third party systems



Webservice Demo

- Inbound web services allow you to access and modify ServiceNow data using a client application.
  - REST API
  - Scripted REST APIs
  - SOAP web service
  - Scripted Web service SOAP
  - Excel web service
  - JSONv2 Web Service
  - Import sets SOAP API
  - ECC Queue(Inbound/Outbound/Polling/MID)
  - XML web service
- Inbound web services are designed to provide third parties with the ability to retrieve (GET) or update (POST) data in ServiceNow, while outbound web services allow ServiceNow to initiate a transaction with a third party (also using either GET or POST, etc.)

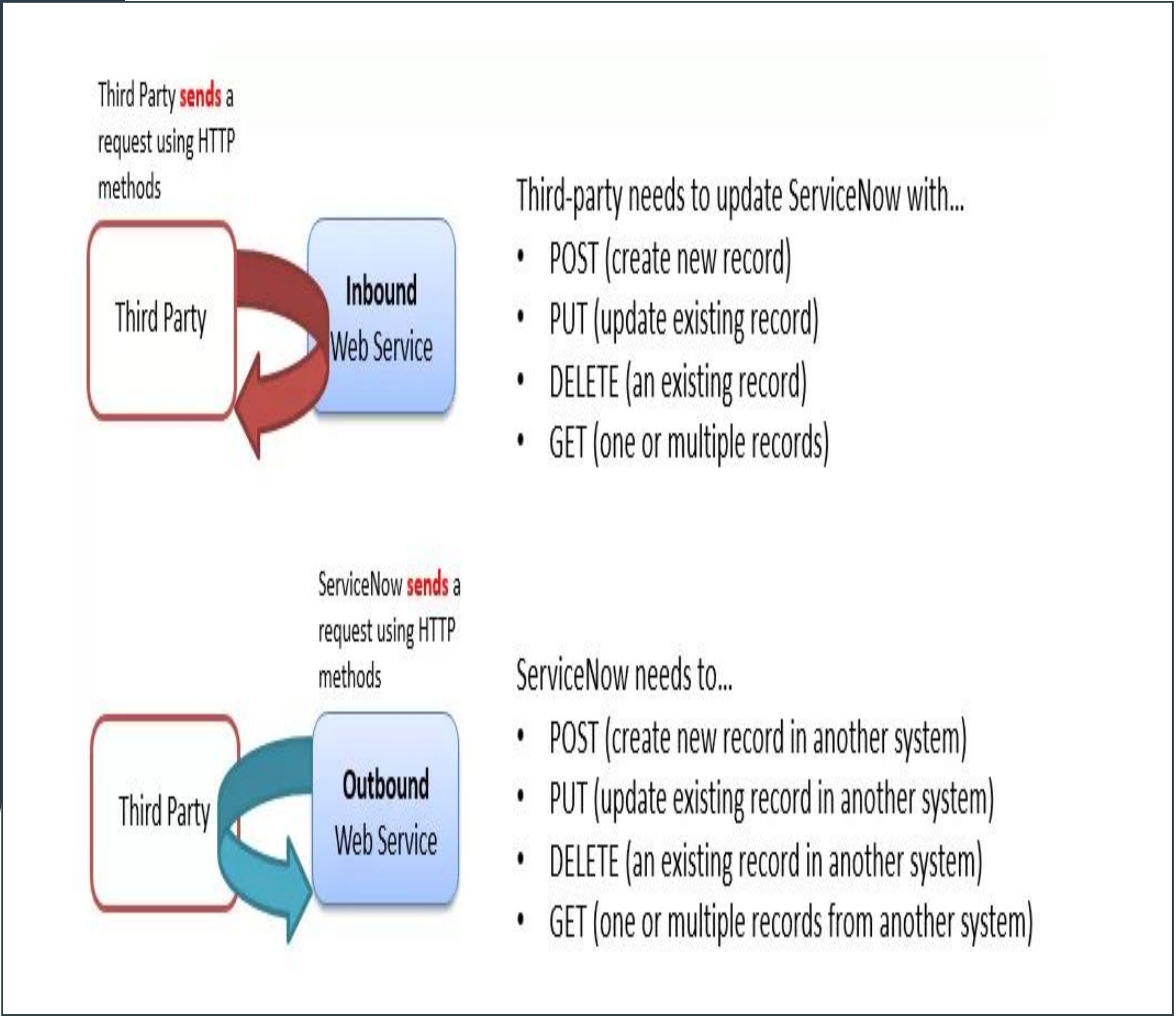

servicenow
Direct Table SOAP / REST
Import sets SOAP API
Import sets REST API (Dublin release)
Scripted Web Services SOAP
JSONv2 REST (Eureka release)
ECC Queue (Inbound/ Outbound / Polling / MID)
Outbound SOAP Message
Outbound REST Message
External Systems (C#, Java, HP CSA, Remedy, any tech understands SOAP/REST)

Webservice Demo

- Outbound web services - Outbound web services allow you to send SOAP and REST messages to external web service providers.
- ServiceNow can communicate to the external systems SOAP or REST end points with API blocks called SOAP Message and REST Messages.
- We can configure these features according to   the client's WSDL and data structures and trigger from Work flows, Business Rules by sending the required parameters to call the client's webservices.
- Types

  - Outbound SOAP web service

  - Outbound REST web service

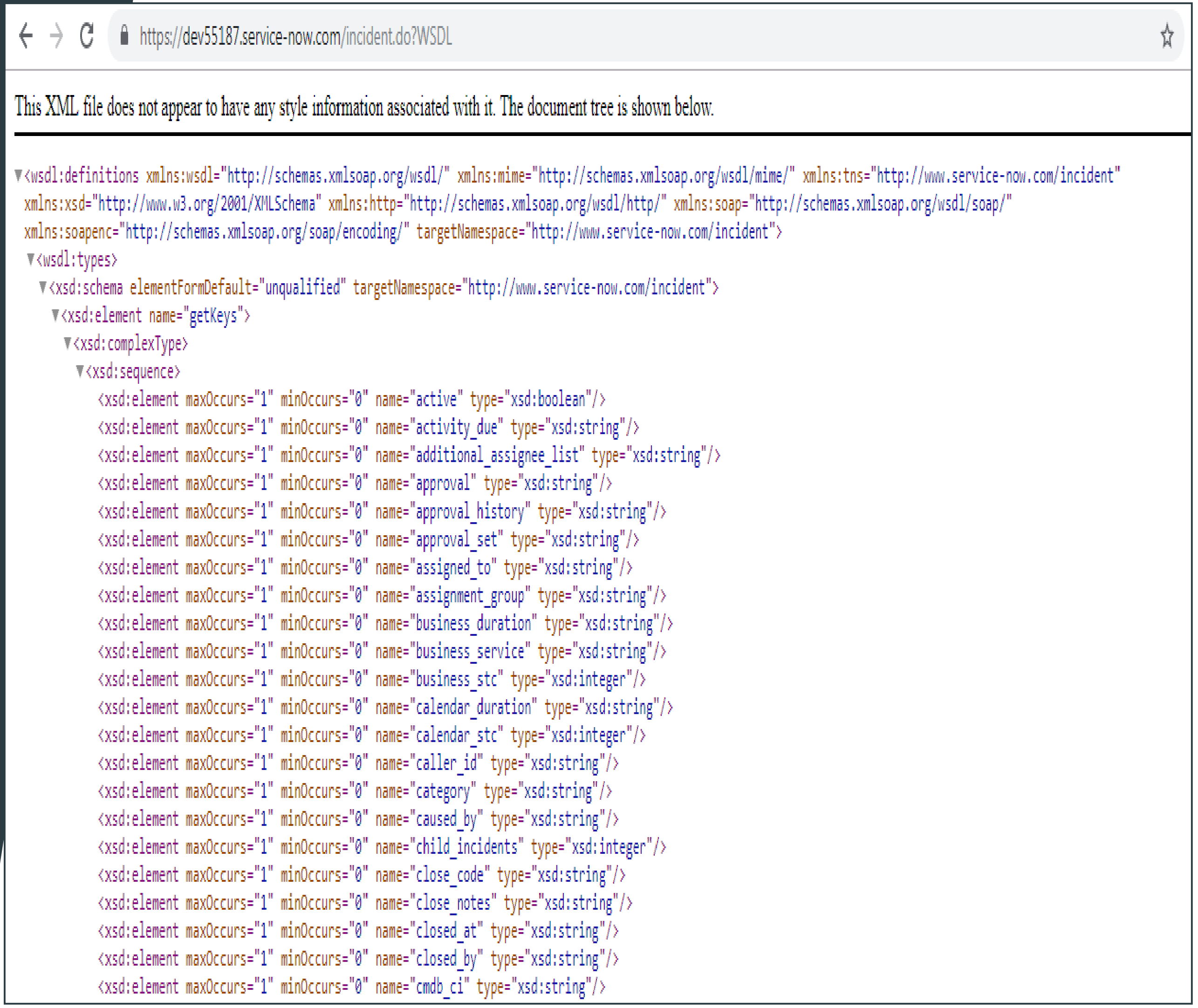

Third Party sends a request using HTTP methods
Third Party
Inbound Web Service
Third-party needs to update ServiceNow with...
• POST (create new record)
• PUT (update existing record)
• DELETE (an existing record)
• GET (one or multiple records)
ServiceNow sends a request using HTTP methods
Third Party
Outbound Web Service
ServiceNow needs to...
• POST (create new record in another system)
• PUT (update existing record in another system)
• DELETE (an existing record in another system)
• GET (one or multiple records from another system)

Webservice Demo

- Direct Webservices – It directly exposes SNOW target tables to third party applications.It allows query of tables and records directly using SOAP,REST, or other web service formats.

  Ex : https://dev55187.service-now.com/incident.do?WSDL

- This generates Webservices descriptive language for this incident table and gets information and actions available related to Incident



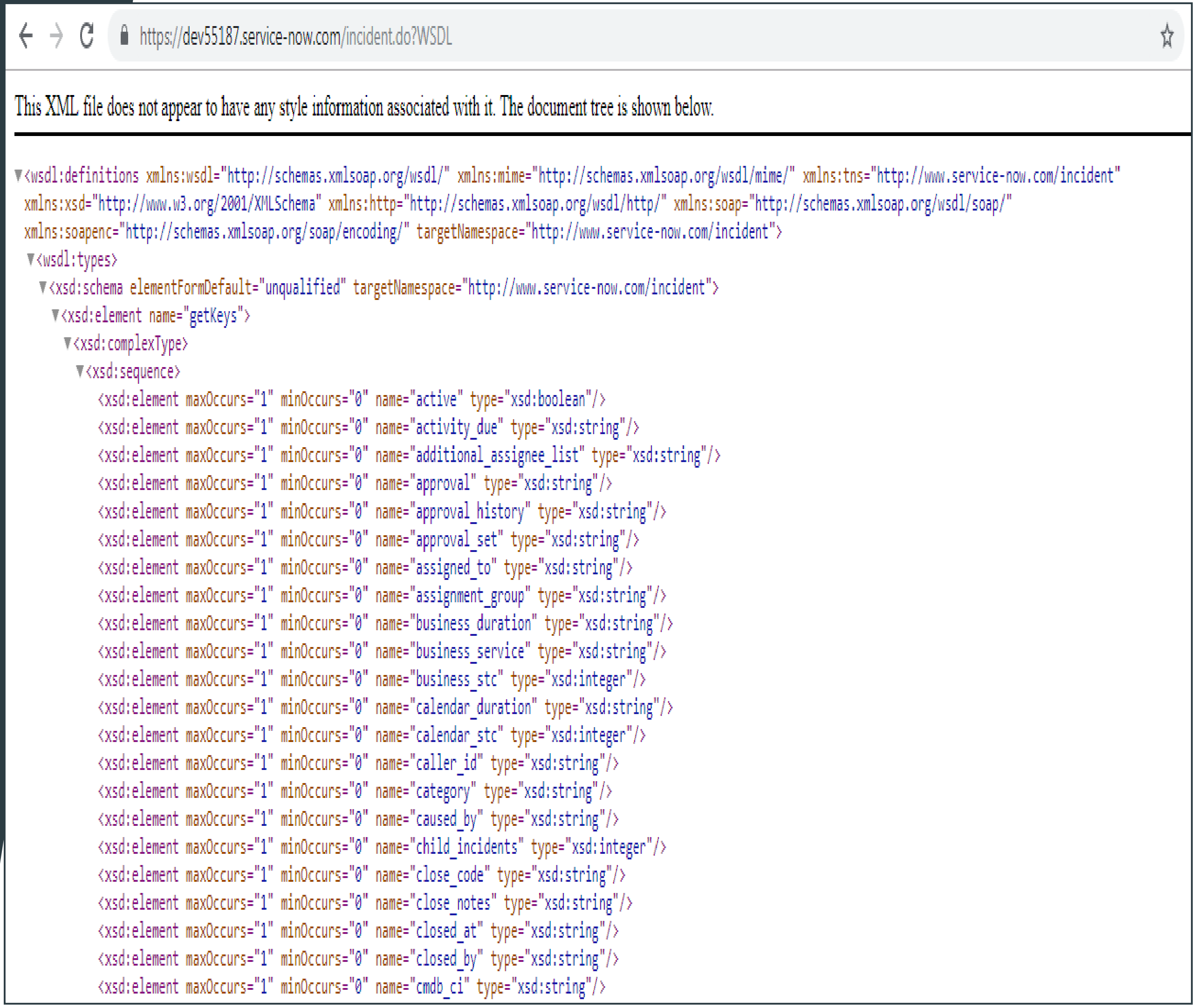
https://dev55187.service-now.com/incident.do?WSDL

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

```
▼<wsdl:definitions xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/" xmlns:tns="http://www.service-now.com/incident"
 xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:http="http://schemas.xmlsoap.org/wsdl/http/" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
 xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/" targetNamespace="http://www.service-now.com/incident">
 ▼<wsdl:types>
  ▼<xsd:schema elementFormDefault="unqualified" targetNamespace="http://www.service-now.com/incident">
   ▼<xsd:element name="getKeys">
    ▼<xsd:complexType>
     ▼<xsd:sequence>
        <xsd:element maxOccurs="1" minOccurs="0" name="active" type="xsd:boolean"/>
        <xsd:element maxOccurs="1" minOccurs="0" name="activity_due" type="xsd:string"/>
        <xsd:element maxOccurs="1" minOccurs="0" name="additional_assignee_list" type="xsd:string"/>
        <xsd:element maxOccurs="1" minOccurs="0" name="approval" type="xsd:string"/>
        <xsd:element maxOccurs="1" minOccurs="0" name="approval_history" type="xsd:string"/>
        <xsd:element maxOccurs="1" minOccurs="0" name="approval_set" type="xsd:string"/>
        <xsd:element maxOccurs="1" minOccurs="0" name="assigned_to" type="xsd:string"/>
        <xsd:element maxOccurs="1" minOccurs="0" name="assignment_group" type="xsd:string"/>
        <xsd:element maxOccurs="1" minOccurs="0" name="business_duration" type="xsd:string"/>
        <xsd:element maxOccurs="1" minOccurs="0" name="business_service" type="xsd:string"/>
        <xsd:element maxOccurs="1" minOccurs="0" name="business_stc" type="xsd:integer"/>
        <xsd:element maxOccurs="1" minOccurs="0" name="calendar_duration" type="xsd:string"/>
        <xsd:element maxOccurs="1" minOccurs="0" name="calendar_stc" type="xsd:integer"/>
        <xsd:element maxOccurs="1" minOccurs="0" name="caller_id" type="xsd:string"/>
        <xsd:element maxOccurs="1" minOccurs="0" name="category" type="xsd:string"/>
        <xsd:element maxOccurs="1" minOccurs="0" name="caused_by" type="xsd:string"/>
        <xsd:element maxOccurs="1" minOccurs="0" name="child_incidents" type="xsd:integer"/>
        <xsd:element maxOccurs="1" minOccurs="0" name="close_code" type="xsd:string"/>
        <xsd:element maxOccurs="1" minOccurs="0" name="close_notes" type="xsd:string"/>
        <xsd:element maxOccurs="1" minOccurs="0" name="closed_at" type="xsd:string"/>
        <xsd:element maxOccurs="1" minOccurs="0" name="closed_by" type="xsd:string"/>
        <xsd:element maxOccurs="1" minOccurs="0" name="cmdb_ci" type="xsd:string"/>
```

Webservice Demo

- Actions include get Method,getResponse,getRecord,get RecordsResponse,update, updateResponse, insert, insertResponse, deleteMultiple.
- It also exposes fields of Incident table through this methods.
- Thus we are exposing these fields and webmethods and third party tools can consume these tables data.
- Thus these type of webservices where we are directly exposing tables are Direct Webservices
- http://wiki.servicenow.com/index.php?title=Web_Service_Import_Sets

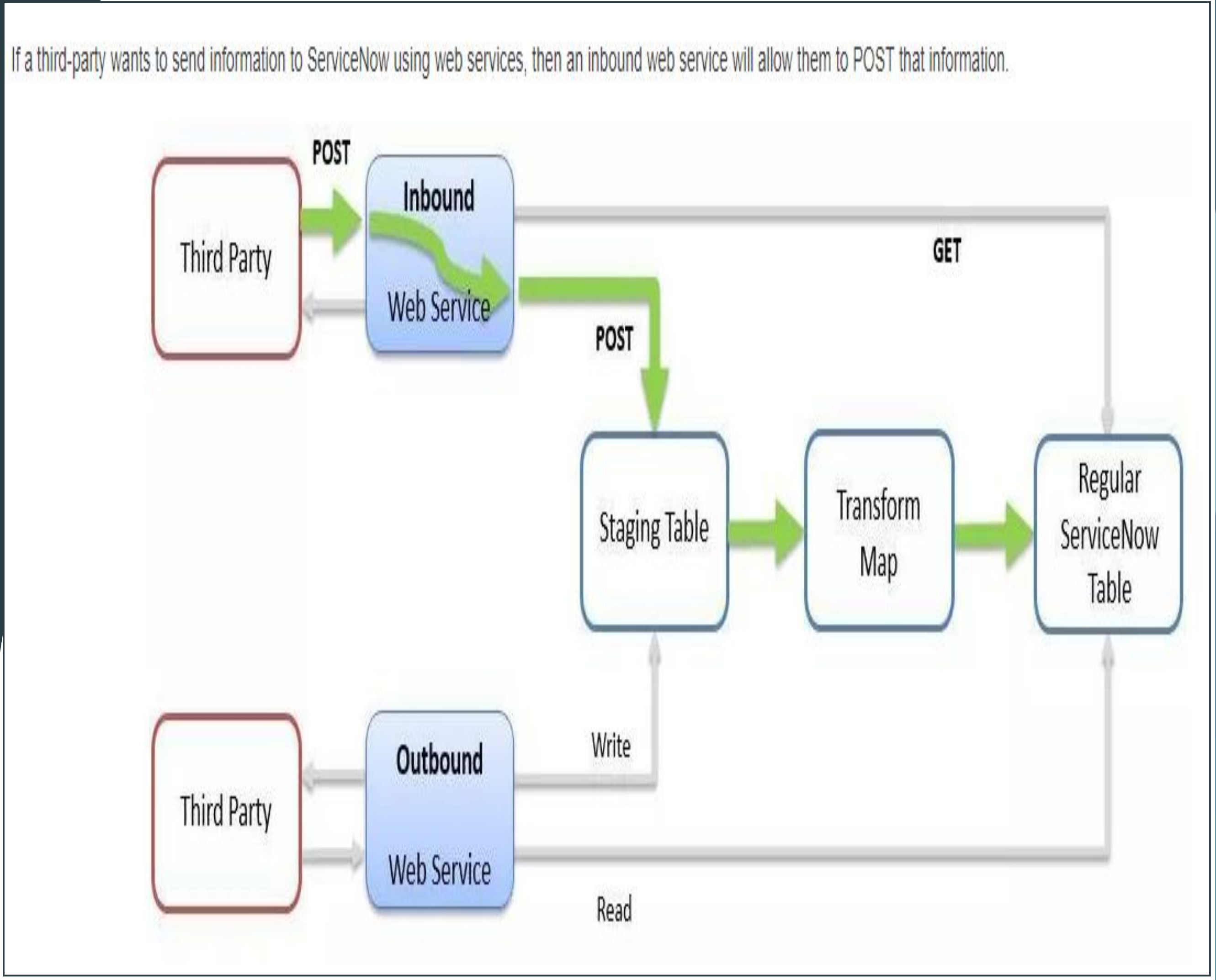

https://dev55187.service-now.com/incident.do?WSDL

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

```
▼<wsdl:definitions xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/" xmlns:tns="http://www.service-now.com/incident"
 xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:http="http://schemas.xmlsoap.org/wsdl/http/" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
 xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/" targetNamespace="http://www.service-now.com/incident">
 ▼<wsdl:types>
  ▼<xsd:schema elementFormDefault="unqualified" targetNamespace="http://www.service-now.com/incident">
   ▼<xsd:element name="getKeys">
    ▼<xsd:complexType>
     ▼<xsd:sequence>
        <xsd:element maxOccurs="1" minOccurs="0" name="active" type="xsd:boolean"/>
        <xsd:element maxOccurs="1" minOccurs="0" name="activity_due" type="xsd:string"/>
        <xsd:element maxOccurs="1" minOccurs="0" name="additional_assignee_list" type="xsd:string"/>
        <xsd:element maxOccurs="1" minOccurs="0" name="approval" type="xsd:string"/>
        <xsd:element maxOccurs="1" minOccurs="0" name="approval_history" type="xsd:string"/>
        <xsd:element maxOccurs="1" minOccurs="0" name="approval_set" type="xsd:string"/>
        <xsd:element maxOccurs="1" minOccurs="0" name="assigned_to" type="xsd:string"/>
        <xsd:element maxOccurs="1" minOccurs="0" name="assignment_group" type="xsd:string"/>
        <xsd:element maxOccurs="1" minOccurs="0" name="business_duration" type="xsd:string"/>
        <xsd:element maxOccurs="1" minOccurs="0" name="business_service" type="xsd:string"/>
        <xsd:element maxOccurs="1" minOccurs="0" name="business_stc" type="xsd:integer"/>
        <xsd:element maxOccurs="1" minOccurs="0" name="calendar_duration" type="xsd:string"/>
        <xsd:element maxOccurs="1" minOccurs="0" name="calendar_stc" type="xsd:integer"/>
        <xsd:element maxOccurs="1" minOccurs="0" name="caller_id" type="xsd:string"/>
        <xsd:element maxOccurs="1" minOccurs="0" name="category" type="xsd:string"/>
        <xsd:element maxOccurs="1" minOccurs="0" name="caused_by" type="xsd:string"/>
        <xsd:element maxOccurs="1" minOccurs="0" name="child_incidents" type="xsd:integer"/>
        <xsd:element maxOccurs="1" minOccurs="0" name="close_code" type="xsd:string"/>
        <xsd:element maxOccurs="1" minOccurs="0" name="close_notes" type="xsd:string"/>
        <xsd:element maxOccurs="1" minOccurs="0" name="closed_at" type="xsd:string"/>
        <xsd:element maxOccurs="1" minOccurs="0" name="closed_by" type="xsd:string"/>
        <xsd:element maxOccurs="1" minOccurs="0" name="cmdb_ci" type="xsd:string"/>
```

Webservice Demo

- Indirect /Import Set webservices - Lets take use case where instead of directly exposing SNOW tables we would exposing staging table or import set table where there would be transformation map which will pull data and push into target SNOW tables.
- It can be both SOAP type or REST type and Supports JSON, CSV, Excel, and XML as input formats.
- http://wiki.servicenow.com/index.php?title=Web_Service_Import_Sets

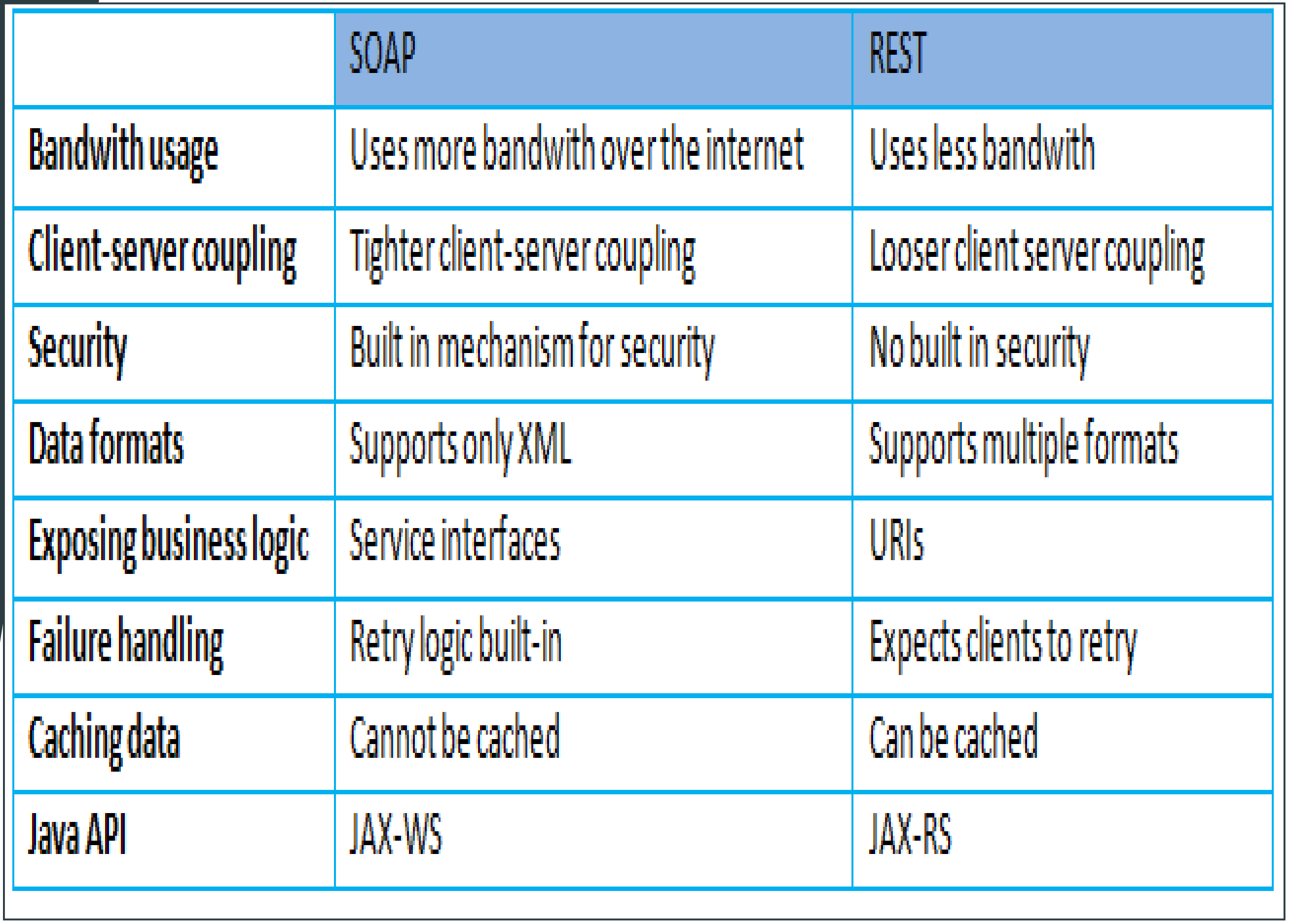

If a third-party wants to send information to ServiceNow using web services, then an inbound web service will allow them to POST that information.

Webservice Demo

- SOAP Webservices - SOAP are simple object access Protocol and is older
- REST - Representative State Transfer is rather new
- Advantages of SOAP over REST

  - SOAP is language,platform and transport independent while REST relies on HTTP

  - SOAP can communicate on Distributed Enterprise Environments REST is more of direct point to point communication



	SOAP	REST
Bandwith usage	Uses more bandwith over the internet	Uses less bandwith
Client-server coupling	Tighter client-server coupling	Looser client server coupling
Security	Built in mechanism for security	No built in security
Data formats	Supports only XML	Supports multiple formats
Exposing business logic	Service interfaces	URIs
Failure handling	Retry logic built-in	Expects clients to retry
Caching data	Cannot be cached	Can be cached
Java API	JAX-WS	JAX-RS

Webservice

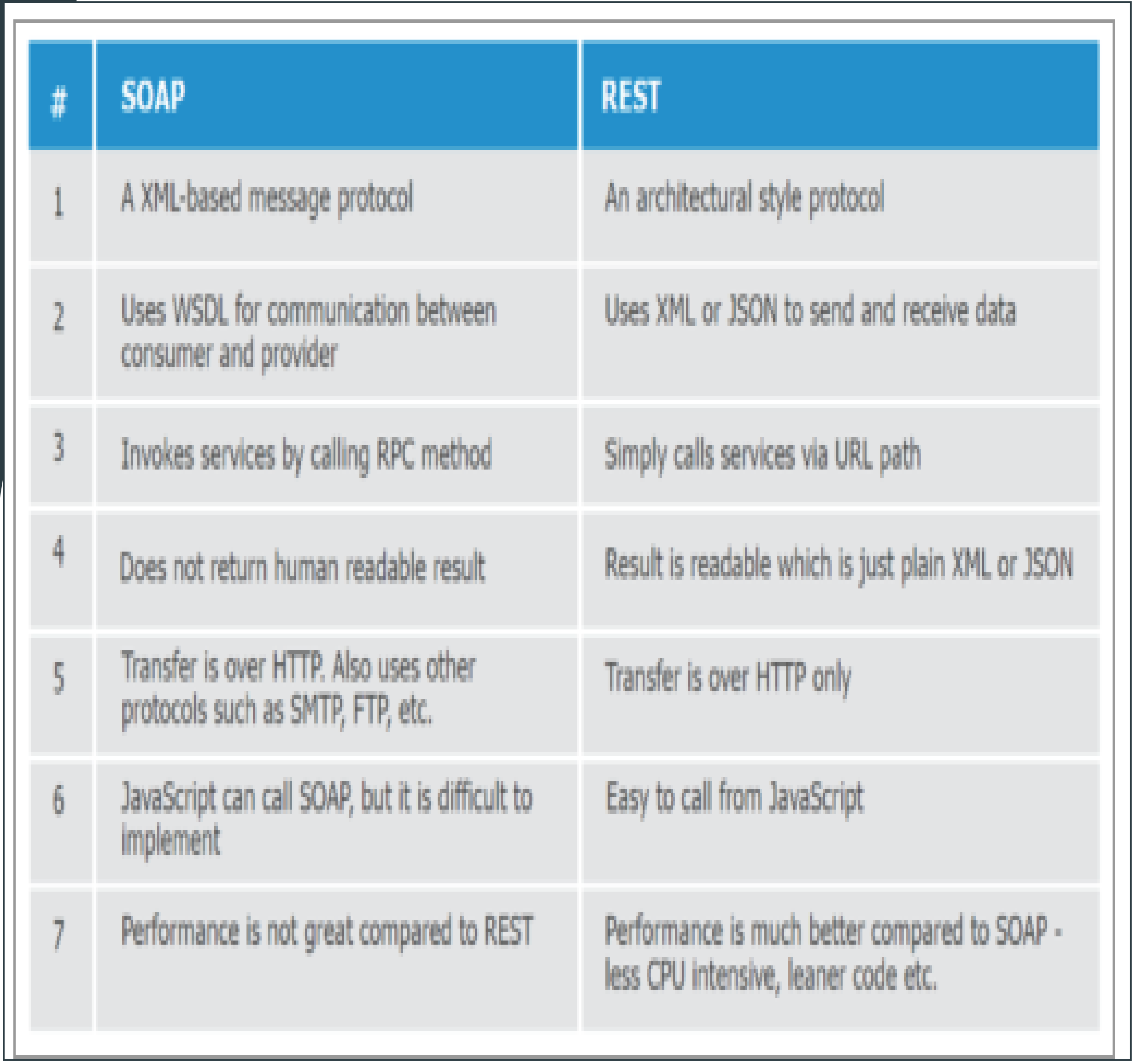
- SOAP is more standardized and provides pre-build extensibility,built in Error handling and Automation in case of certain language products.
- Advantages of REST over SOAP

  - It is more easier and flexible

  - No expensive tools required to interact with web services

  - Smaller learning curve

  - Efficient(SOAP uses XML and REST uses smaller message formats



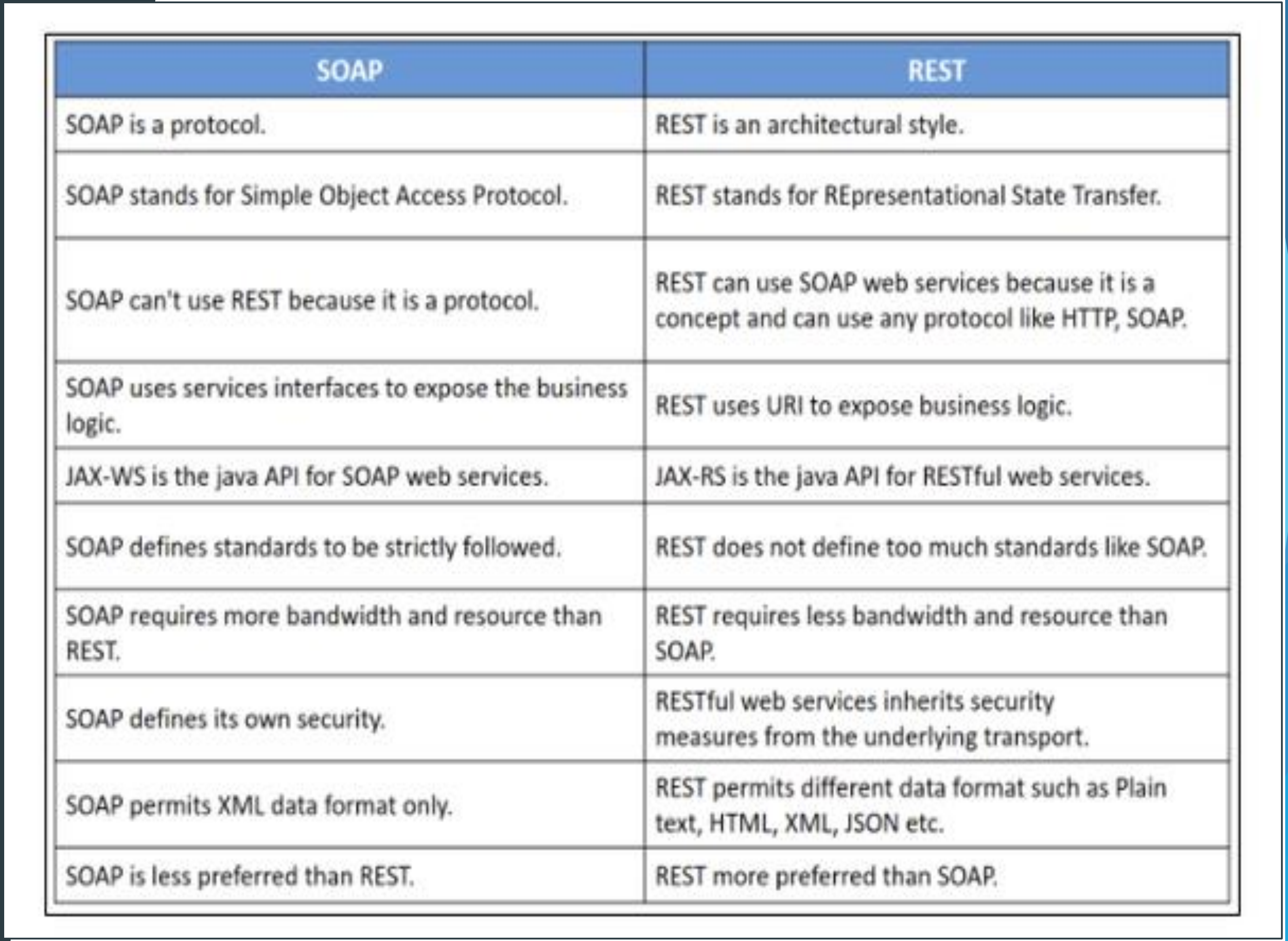
#	SOAP	REST
1	A XML-based message protocol	An architectural style protocol
2	Uses WSDL for communication between consumer and provider	Uses XML or JSON to send and receive data
3	Invokes services by calling RPC method	Simply calls services via URL path
4	Does not return human readable result	Result is readable which is just plain XML or JSON
5	Transfer is over HTTP. Also uses other protocols such as SMTP, FTP, etc.	Transfer is over HTTP only
6	JavaScript can call SOAP, but it is difficult to implement	Easy to call from JavaScript
7	Performance is not great compared to REST	Performance is much better compared to SOAP - less CPU intensive, leaner code etc.

Webservice Demo

- It is fast and does not require extensive processing required
- Closer to other web technologies

- REST - http://wiki.servicenow.com/index.php?title=REST_API_Explorer
- SOAP

  http://wiki.servicenow.com/index.php?title=SOAP_Web_Service

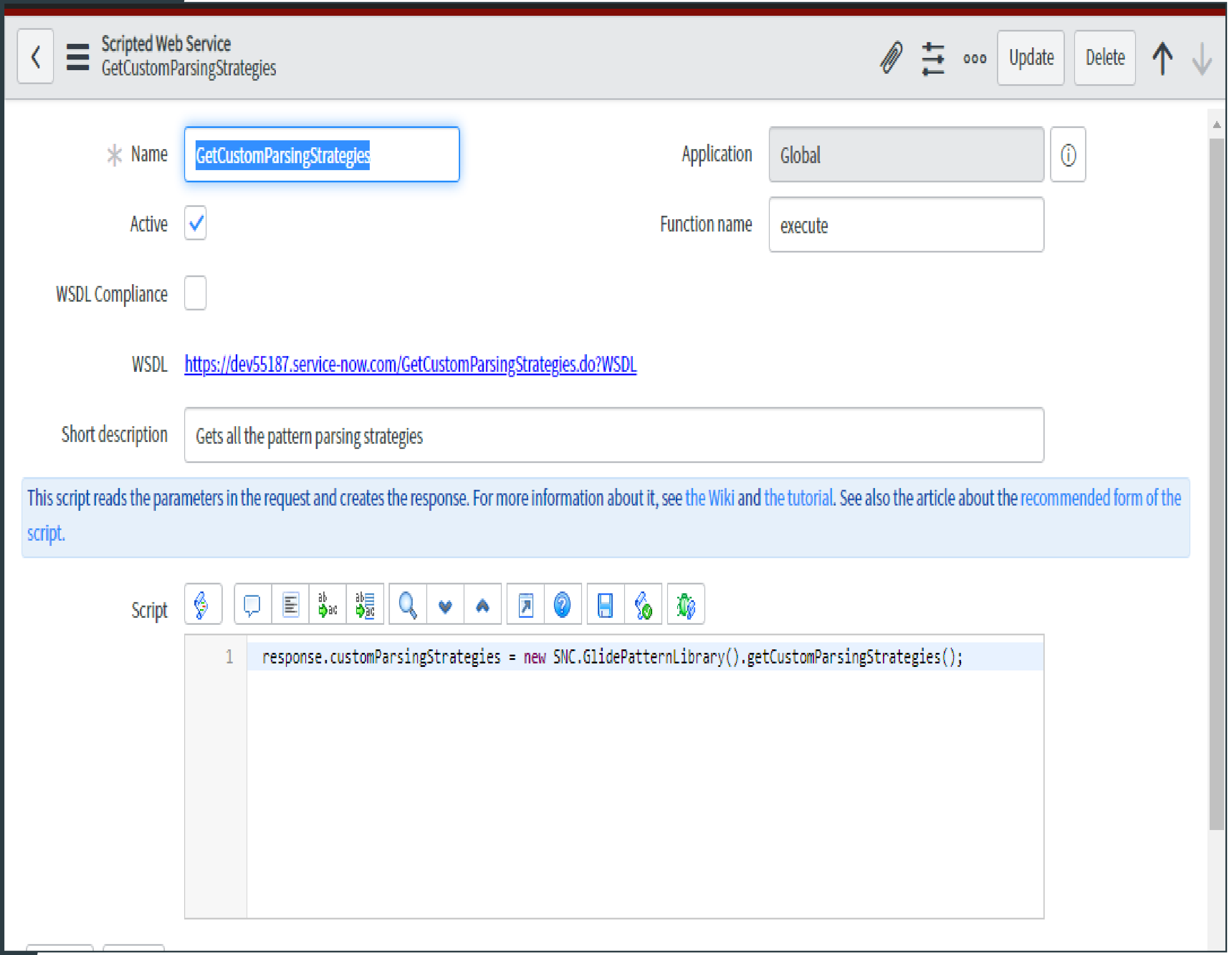


SOAP	REST
SOAP is a protocol.	REST is an architectural style.
SOAP stands for Simple Object Access Protocol.	REST stands for REpresentational State Transfer.
SOAP can't use REST because it is a protocol.	REST can use SOAP web services because it is a concept and can use any protocol like HTTP, SOAP.
SOAP uses services interfaces to expose the business logic.	REST uses URI to expose business logic.
JAX-WS is the java API for SOAP web services.	JAX-RS is the java API for RESTful web services.
SOAP defines standards to be strictly followed.	REST does not define too much standards like SOAP.
SOAP requires more bandwidth and resource than REST.	REST requires less bandwidth and resource than SOAP.
SOAP defines its own security.	RESTful web services inherits security measures from the underlying transport.
SOAP permits XML data format only.	REST permits different data format such as Plain text, HTML, XML, JSON etc.
SOAP is less preferred than REST.	REST more preferred than SOAP.

Webservice Demo

Types of Webservices

- Scripted Webservices - When creating webservices we can apply our custom scripts where we can define parameters to webservice through Java scripts
- It can be both REST as well as SOAP Scripted web service
- http://wiki.servicenow.com/index.php?title=Scripted_Web_Services



Name	GetCustomParsingStrategies	Application	Global
Active	✓	Function name	execute
WSDL Compliance			
WSDL	https://dev55187.service-now.com/GetCustomParsingStrategies.do?WSDL		
Short description	Gets all the pattern parsing strategies		

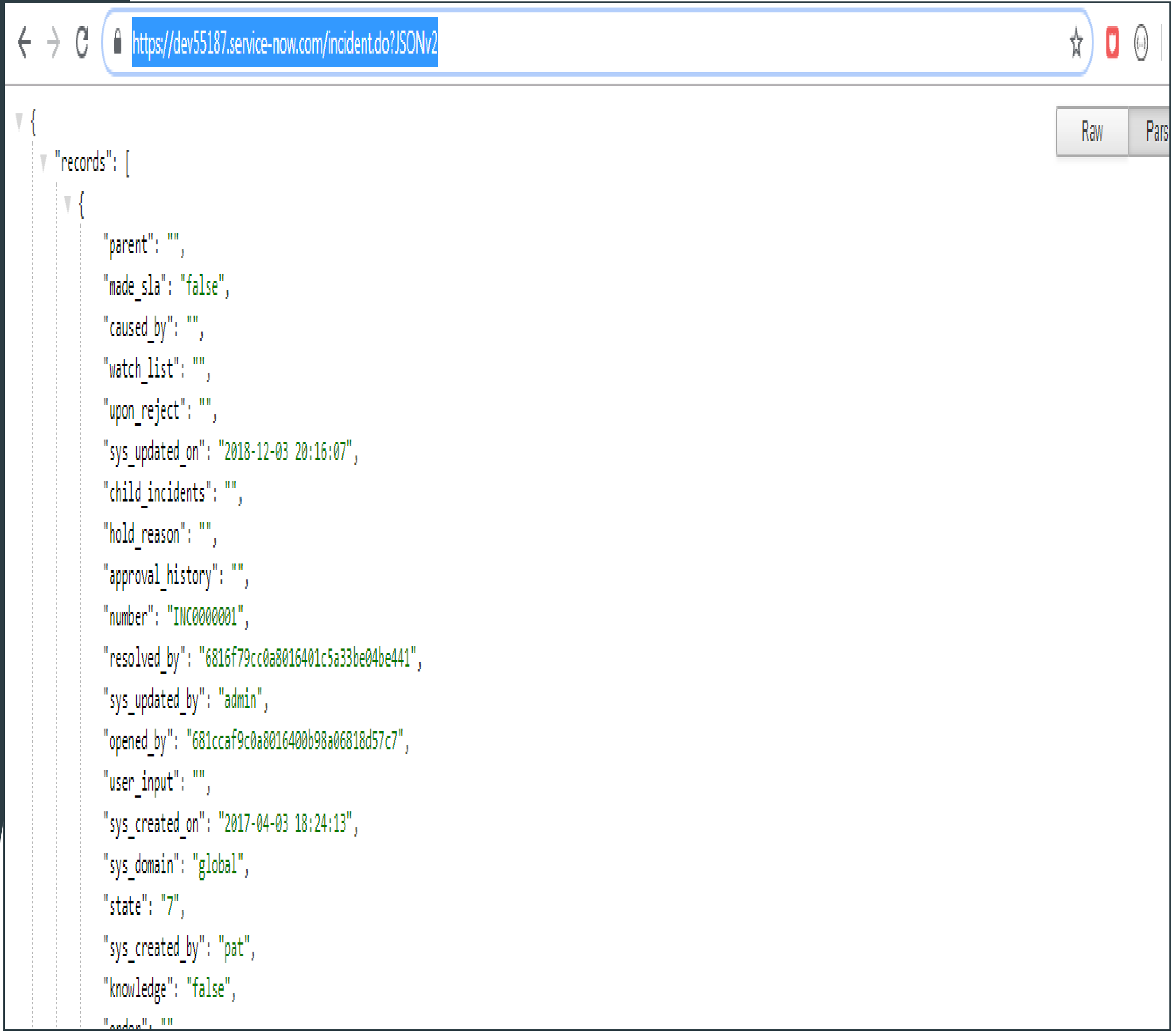
This script reads the parameters in the request and creates the response. For more information about it, see the Wiki and the tutorial. See also the article about the recommended form of the script.

Script

```
response.customParsingStrategies = new SNC.GlidePatternLibrary().getCustomParsingStrategies();
```

Webservice Demo

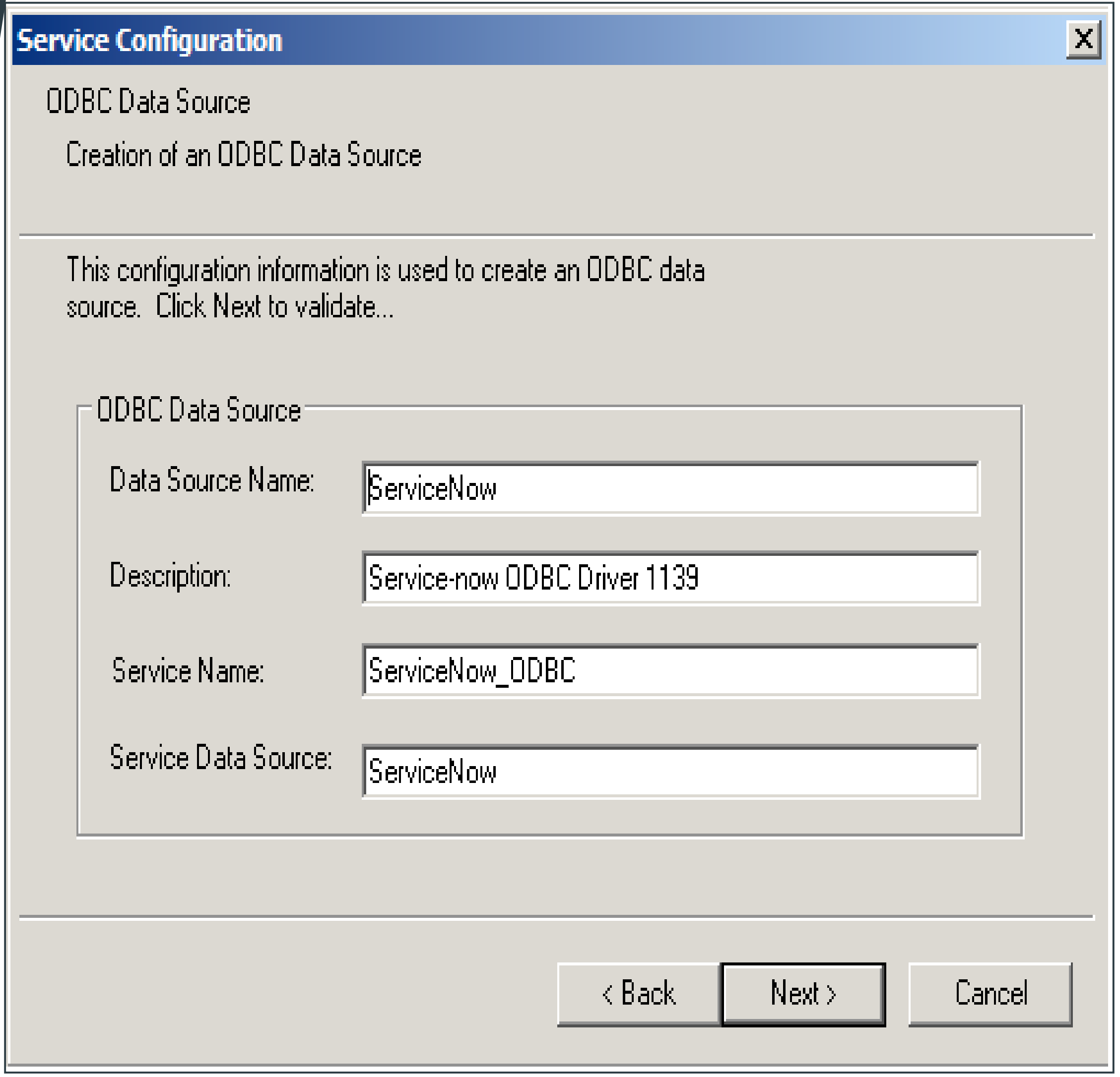
- **JSONV2 - Inbound Webservices This exposes SNOW table data in JSON format**
- https://dev55187.service-now.com/incident.do?JSONv2 Install JSON formatter extension for chrome
- This data can be parsed and used with third party application providing Action Parameters and Basic Authentication.
- http://wiki.servicenow.com/index.php?title=JSONv2_Web_Service



```
{
  "records": [
    {
      "parent": "",
      "made_sla": "false",
      "caused_by": "",
      "watch_list": "",
      "upon_reject": "",
      "sys_updated_on": "2018-12-03 20:16:07",
      "child_incidents": "",
      "hold_reason": "",
      "approval_history": "",
      "number": "INC0000001",
      "resolved_by": "6816f79cc0a8016401c5a33be04be441",
      "sys_updated_by": "admin",
      "opened_by": "681ccaf9c0a8016400b98a06818d57c7",
      "user_input": "",
      "sys_created_on": "2017-04-03 18:24:13",
      "sys_domain": "global",
      "state": "7",
      "sys_created_by": "pat",
      "knowledge": "false",
      "order": ""
```

Webservice Demo

- ODBC driver Webservice – This is another type of Inbound web service which helps in connecting to databases like Microsoft SQL server using DSN and ODBC driver set in system environment.
- We can create direct link by creating link server as SNOW table in Microsoft SQL server DB
- https://docs.servicenow.com/bundle/london-application-development/page/integrate/odbc-driver/task/t_DownloadAndInstallTheODBCDriver.html



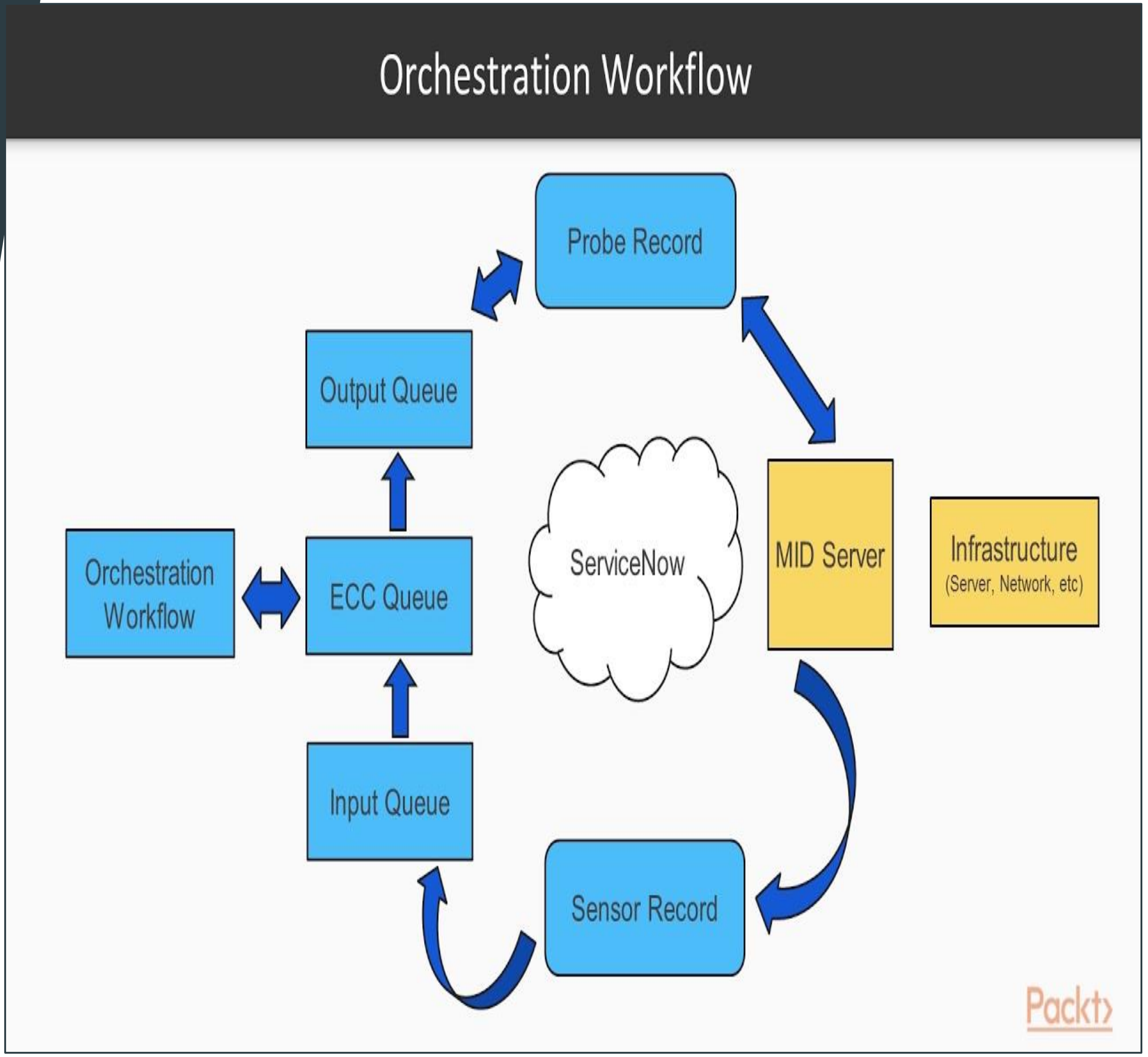
This configuration information is used to create an ODBC data source. Click Next to validate...

ODBC Data Source

Data Source Name:	ServiceNow
Description:	Service-now ODBC Driver 1139
Service Name:	ServiceNow_ODBC
Service Data Source:	ServiceNow

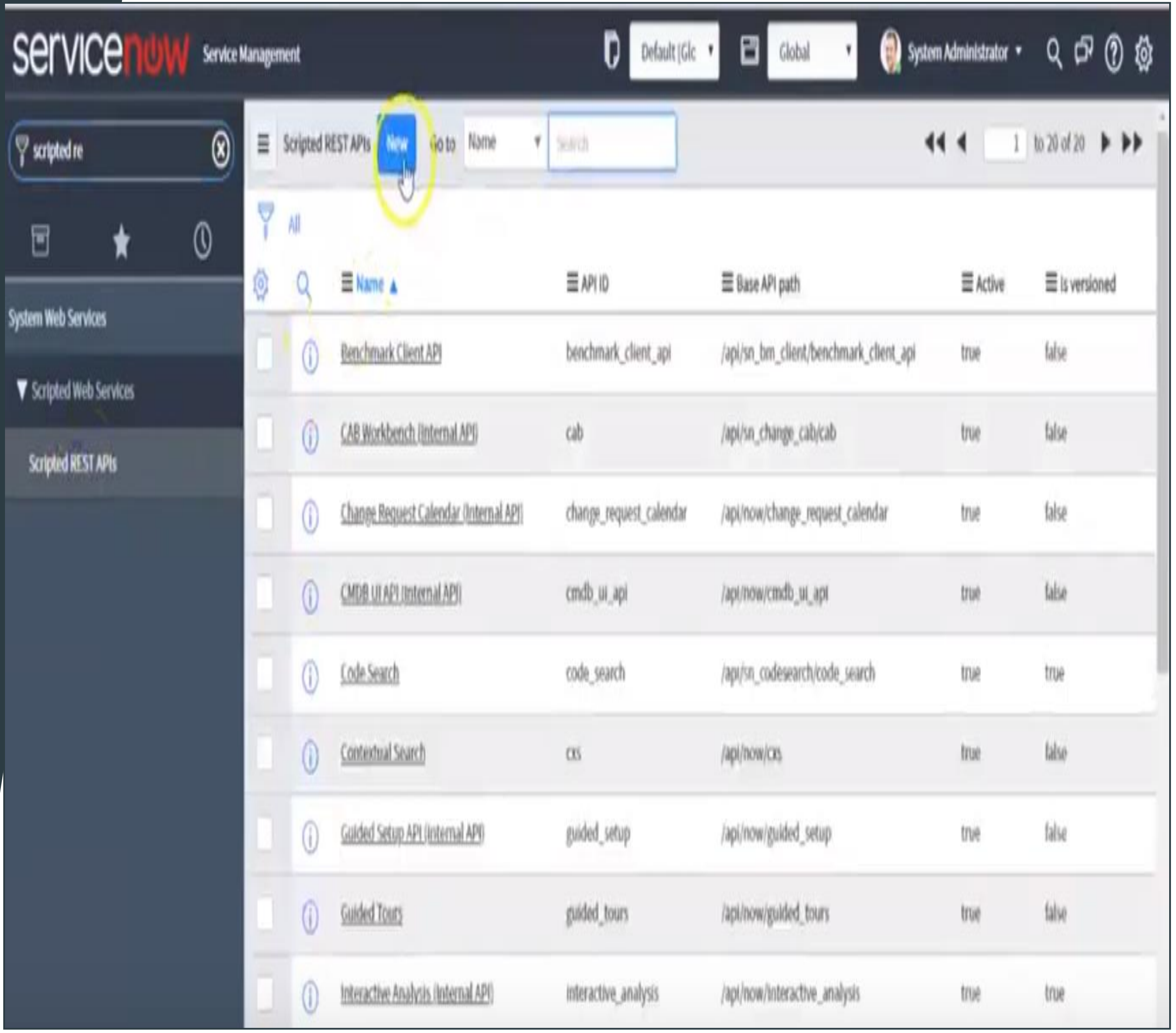
Webservice Demo

- **ECC Queue** - It's a External Communication Channel and is queue based (actually a table acts like a queue in SNOW).
- Both Inbound and outbound but Asynchronous
- Supported with SOAP, REST and JSONv2 APIs
- Business Rules can trigger on this queue to fire the events or SOAP or REST Messages (outbound) to update client systems
- MID Server (small java agents) can be used if you have the external system behind the firewalls.
- Payload can be arbitrary (JSON, XML, CSV, text)



Webservice – Scripted REST API

- Scripted Webservices - When creating webservices we can apply our custom scripts where we can define parameters to webservice through Java scripts
- Go to Navigator and type scripted REST APIs
- Create new Scripted REST API

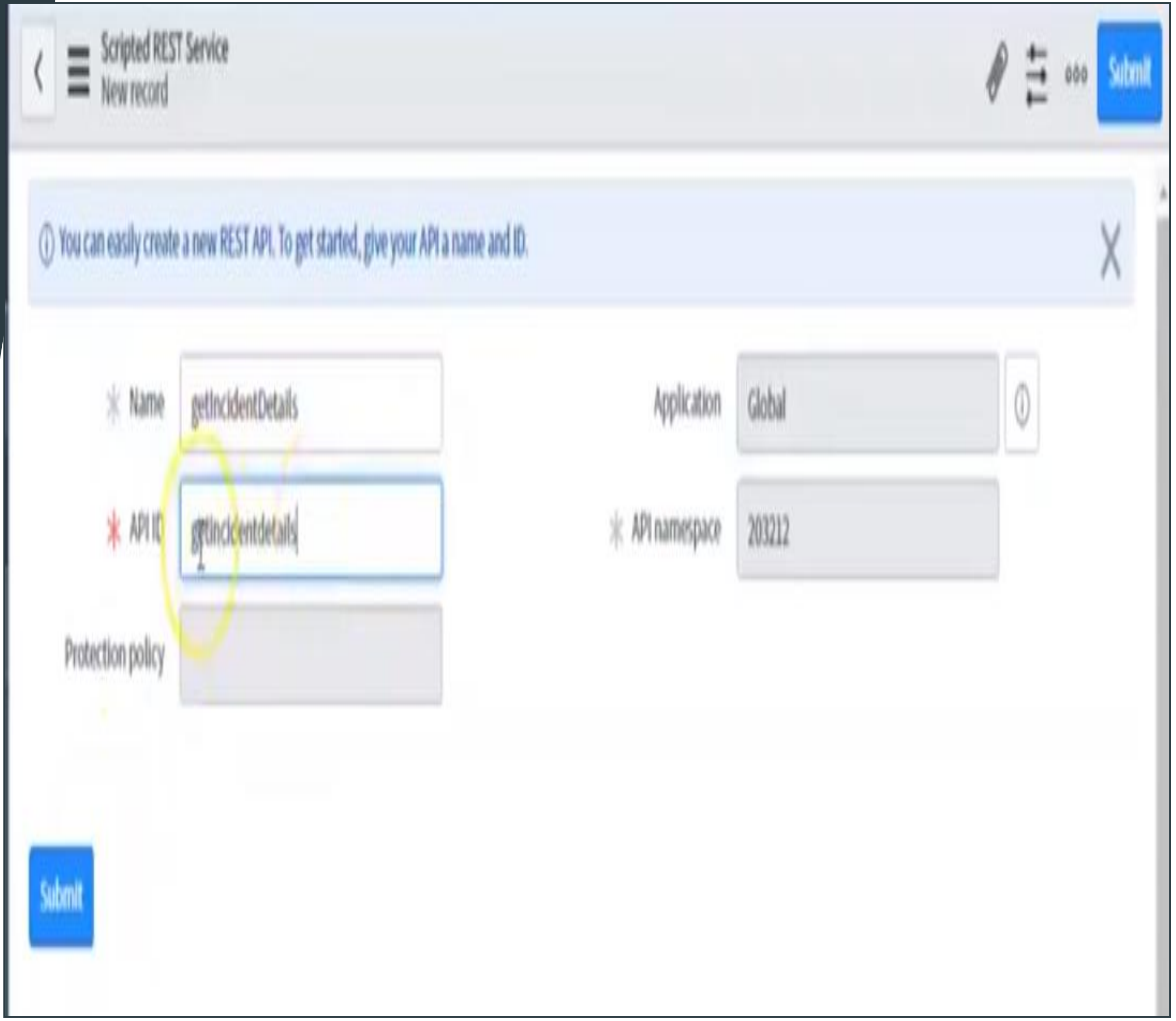


Name ▲	API ID	Base API path	Active	Is versioned
Benchmark Client API	benchmark_client_api	/api/sn_bm_client/benchmark_client_api	true	false
CAB Workbench (Internal API)	cab	/api/sn_change_cab/cab	true	false
Change Request Calendar (Internal API)	change_request_calendar	/api/now/change_request_calendar	true	false
CMDB UI API (Internal API)	cmdb_ui_api	/api/now/cmdb_ui_api	true	false
Code Search	code_search	/api/sn_codesearch/code_search	true	true
Contextual Search	cxs	/api/now/cxs	true	false
Guided Setup API (Internal API)	guided_setup	/api/now/guided_setup	true	false
Guided Tours	guided_tours	/api/now/guided_tours	true	false
Interactive Analysis (Internal API)	interactive_analysis	/api/now/interactive_analysis	true	true

Webservice – Scripted REST API

- Use case requirement is to retrieve Incident tickets details from SNOW instance.Hence we would create custom scripted rest API
- Name : getIncidentDetails

  API id can be modified as per requirement
- Click Submit button

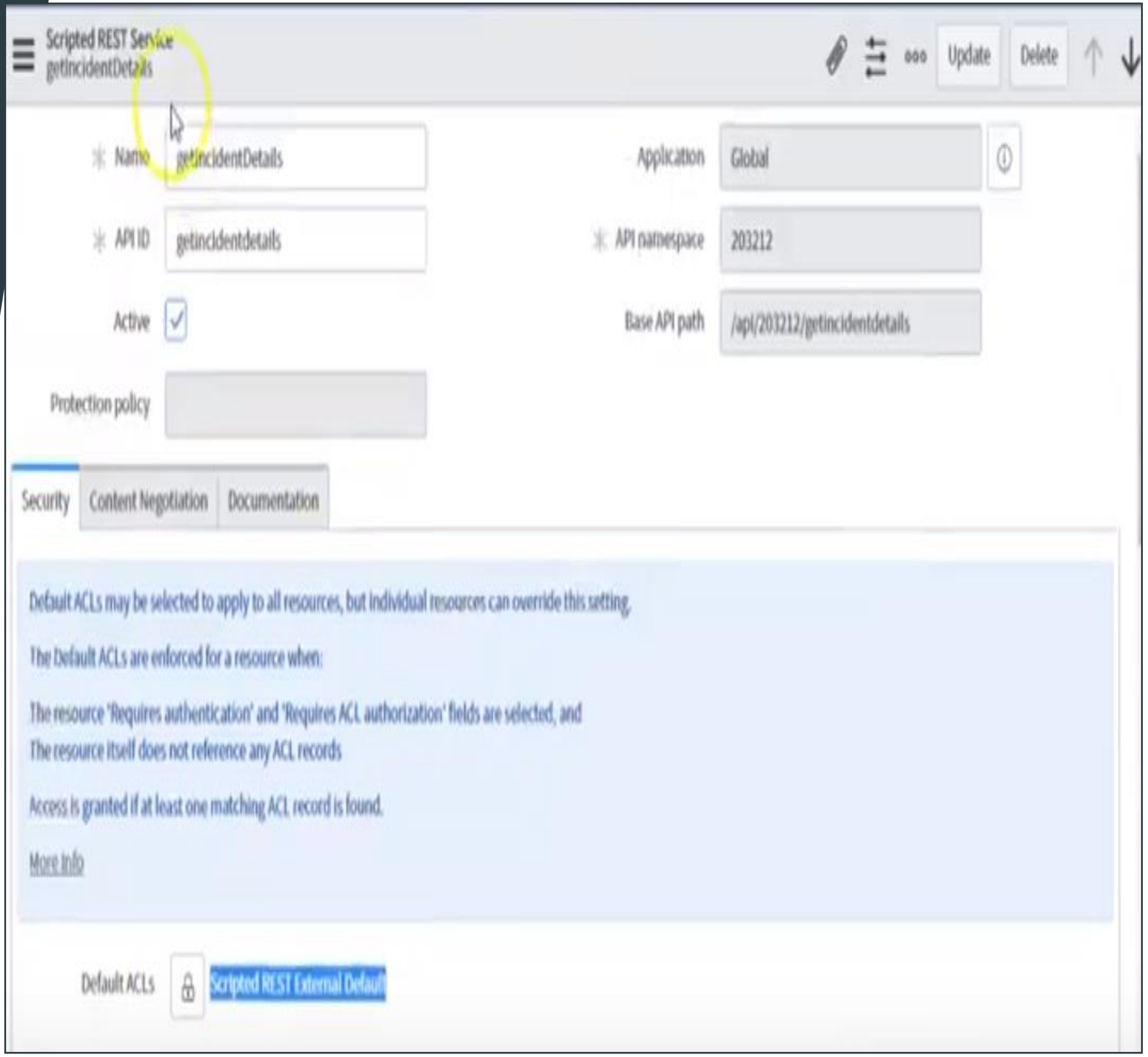

Scripted REST Service
New record

You can easily create a new REST API. To get started, give your API a name and ID.

* Name	getIncidentDetails	Application	Global
* API ID	getIncidentdetails	* API namespace	203212
Protection policy			

Webservice – Scripted REST API

- We can observe default ACL are created for scripted REST API service



Scripted REST Service
getincidentDetails

Name	getincidentDetails	Application	Global
API ID	getincidentdetails	API namespace	203212
Active	✓	Base API path	/api/203212/getincidentdetails
Protection policy			

Default ACLs may be selected to apply to all resources, but individual resources can override this setting.

The Default ACLs are enforced for a resource when:

The resource 'Requires authentication' and 'Requires ACL authorization' fields are selected, and
The resource itself does not reference any ACL records

Access is granted if at least one matching ACL record is found.

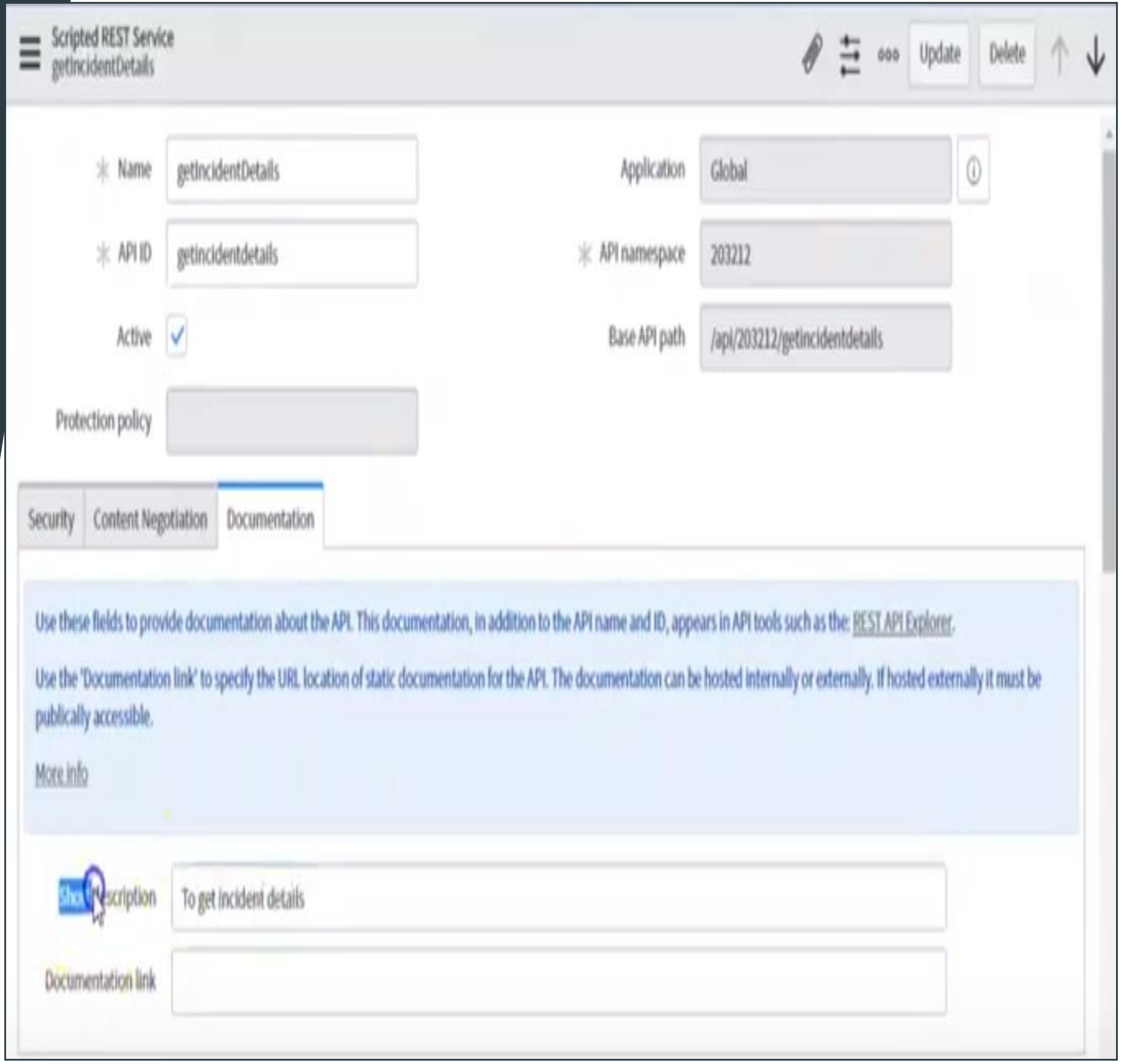
More info

Default ACLs: Scripted REST External Default

Webservice – Scripted REST API

- On Documentation tab

  - Short Description : To get Incident details
- Update the record



Scripted REST Service
getIncidentDetails

Name	getIncidentDetails	Application	Global
API ID	getincidentdetails	API namespace	203212
Active	✓	Base API path	/api/203212/getincidentdetails
Protection policy			

Use these fields to provide documentation about the API. This documentation, in addition to the API name and ID, appears in API tools such as the: REST API Explorer.

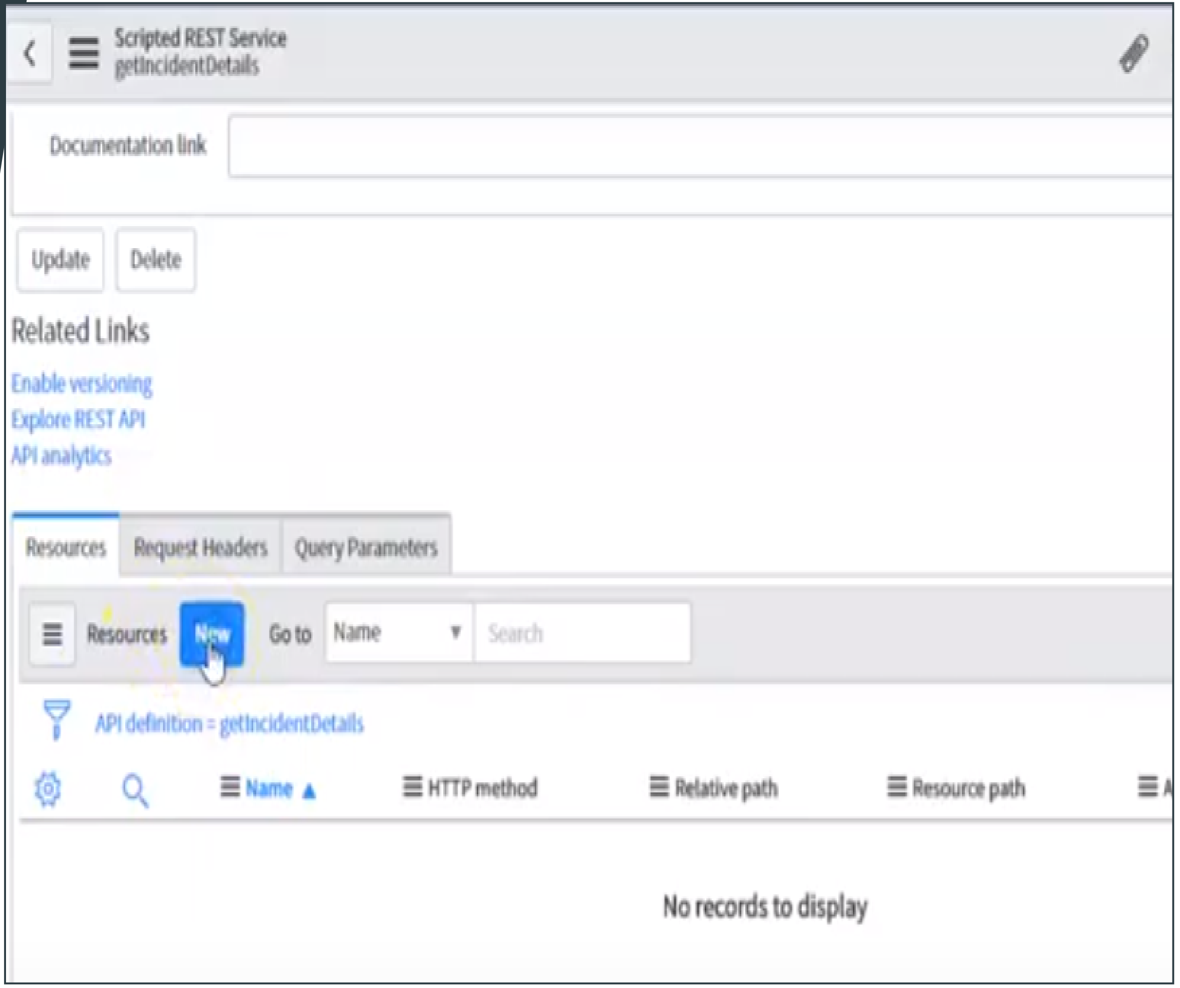
Use the 'Documentation link' to specify the URL location of static documentation for the API. The documentation can be hosted internally or externally. If hosted externally it must be publically accessible.

More info

Short Description	To get incident details
Documentation link	

Webservice – Scripted REST API

- **Create resources under related links.Click on new button**



Scripted REST Service
getIncidentDetails

Documentation link

Update | Delete

Related Links

Enable versioning
Explore REST API
API analytics

Resources | Request Headers | Query Parameters

Resources | New | Go to | Name | Search

API definition = getIncidentDetails

Name ▲	HTTP method	Relative path	Resource path

No records to display

Webservice – Scripted REST API

- Name : incidentDetails

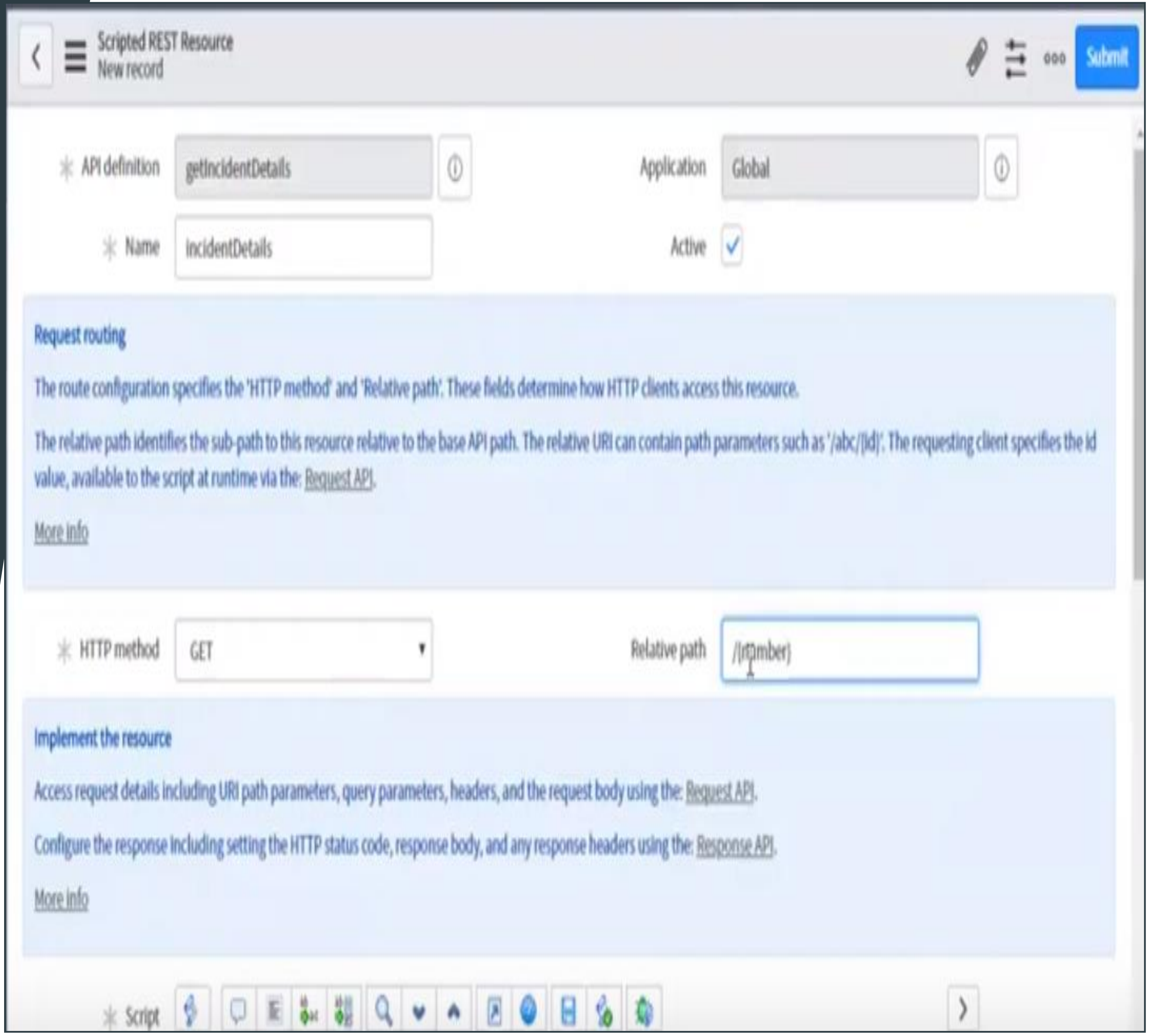
  HTTP method:GET(As per reqt. We need to retreive information from SNOW instance)

  POST – to create a record

  PUT/PATCH – Update a record

  DELETE – to remove a record
- Select method as GET

  Relative path:/{number} //on the basis of this number we will fetch all incident details


Scripted REST Resource
New record

Submit

API definition	getIncidentDetails	Application	Global
Name	incidentDetails	Active	✓

Request routing

The route configuration specifies the 'HTTP method' and 'Relative path'. These fields determine how HTTP clients access this resource.

The relative path identifies the sub-path to this resource relative to the base API path. The relative URI can contain path parameters such as '/abc/{id}'. The requesting client specifies the id value, available to the script at runtime via the: Request API.

More info

HTTP method	GET	Relative path	/{number}

Implement the resource

Access request details including URI path parameters, query parameters, headers, and the request body using the: Request API.

Configure the response including setting the HTTP status code, response body, and any response headers using the: Response API.

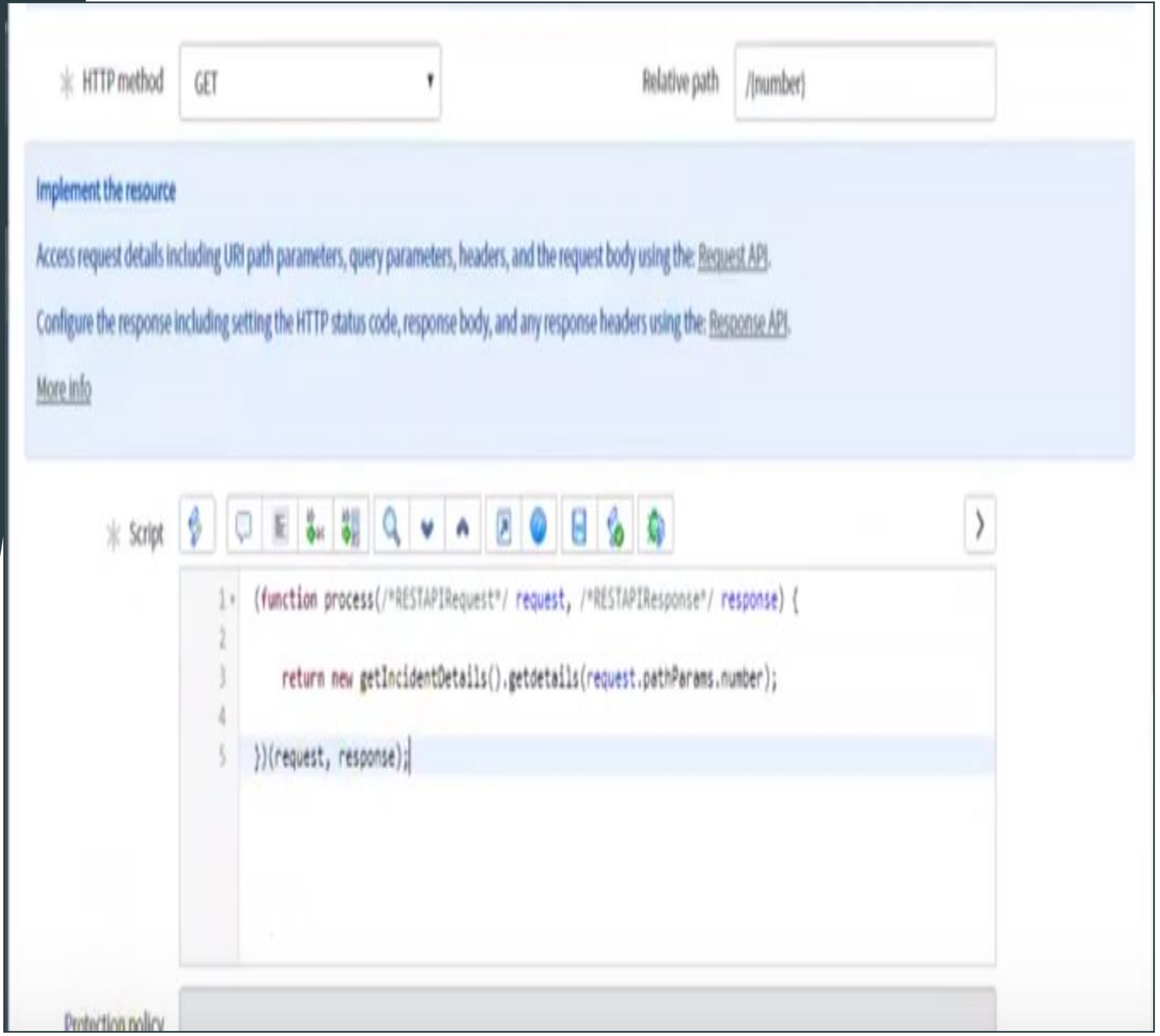
More info

Script

Webservice – Scripted REST API

- Request and Response are objects to retrieve data from instance
- We will write script include to fetch incident details on the basis of number mentioned in relative path

  return new getIncidentDetails().getdetails(request.pathParams.number);



HTTP method: GET

Relative path: /{number}

Implement the resource

Access request details including URI path parameters, query parameters, headers, and the request body using the: Request API.

Configure the response including setting the HTTP status code, response body, and any response headers using the: Response API.

More info

Script

```
(function process(/*RESTAPIRequest*/ request, /*RESTAPIResponse*/ response) {

    return new getIncidentDetails().getdetails(request.pathParams.number);

})(request, response);
```

Protection policy

Webservice – Scripted REST API

- When we are passing parameters for the function getdetails() we can pass different variables like

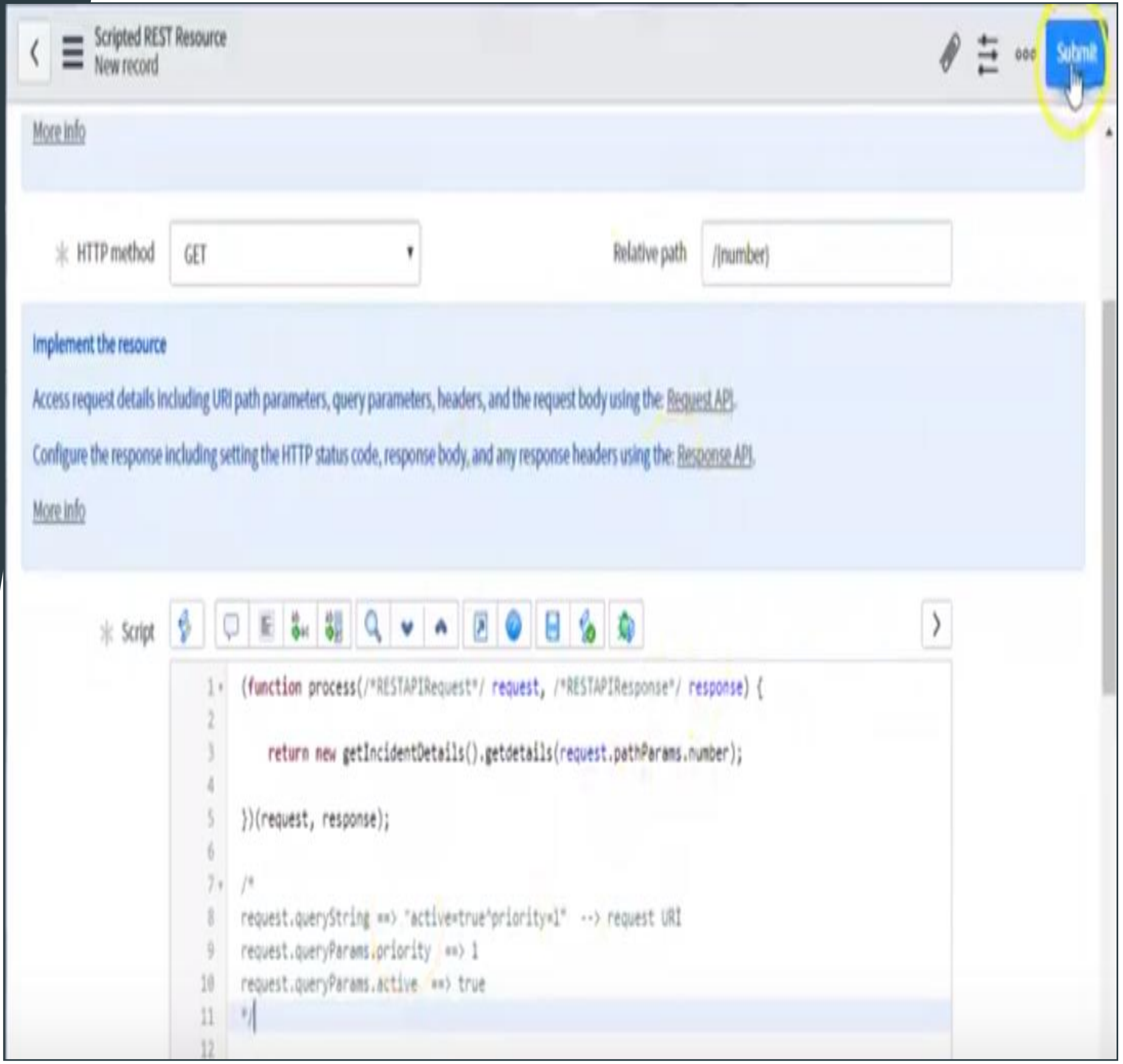
  request.querystring == "active=true^priority=1"

  //querystring uses entire string which is coming from requested URI

  request.queryParams.active

  request.queryParams.priority

- Click Submit



HTTP method: GET
Relative path: /{number}

Implement the resource

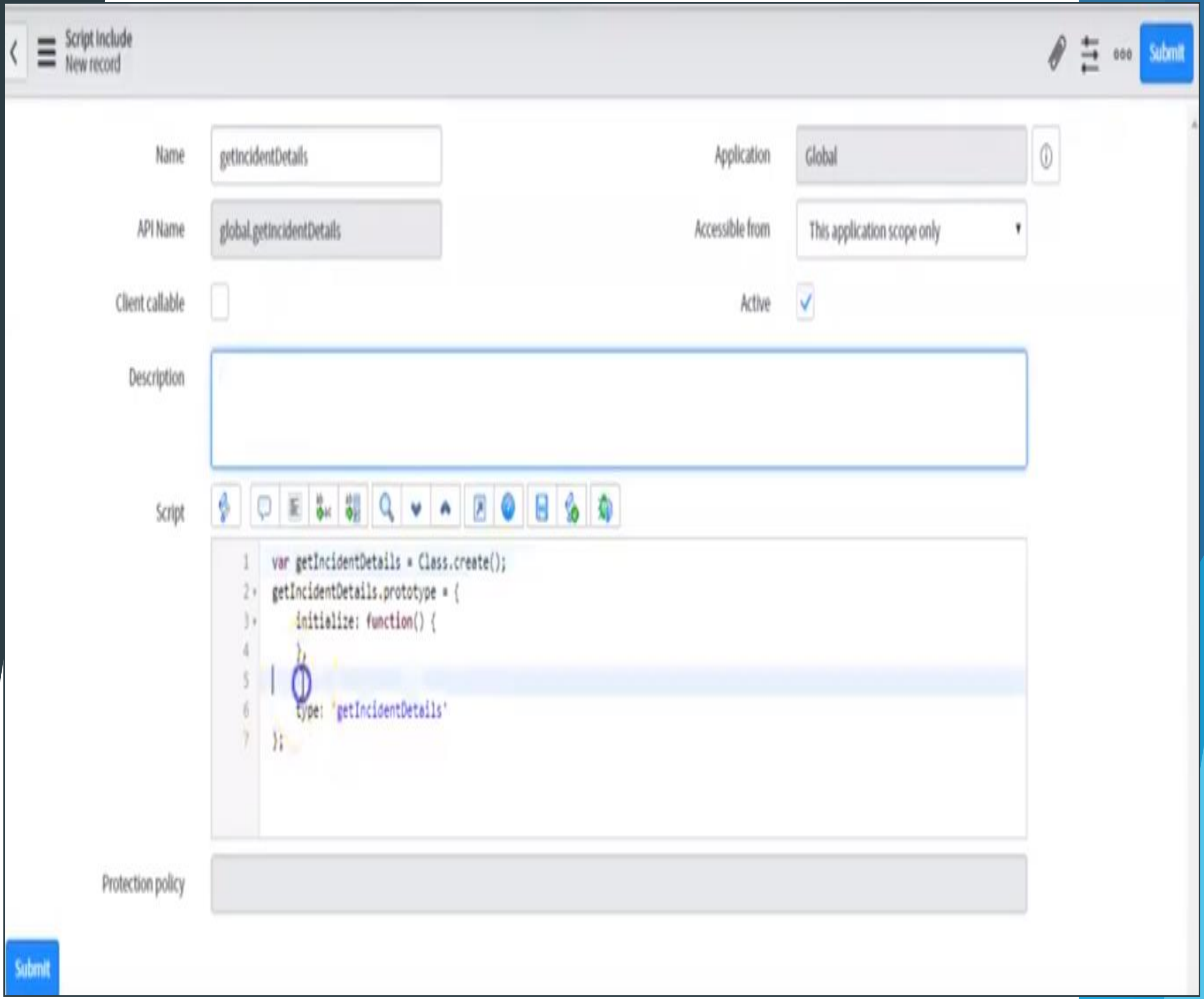
Access request details including URI path parameters, query parameters, headers, and the request body using the: Request API.

Configure the response including setting the HTTP status code, response body, and any response headers using the: Response API.

```
(function process(/*RESTAPIRequest*/ request, /*RESTAPIResponse*/ response) {

    return new getIncidentDetails().getdetails(request.pathParams.number);

})(request, response);

/*
request.queryString ==> "active=true^priority=1" --> request URI
request.queryParams.priority ==> 1
request.queryParams.active ==> true
*/
```

Webservice – Scripted REST API

- Lets now create the scriptInclude with the same name copied from scripted REST API i.e getIncidentDetails
- Now write the function getdetails we defined in scripted REST API

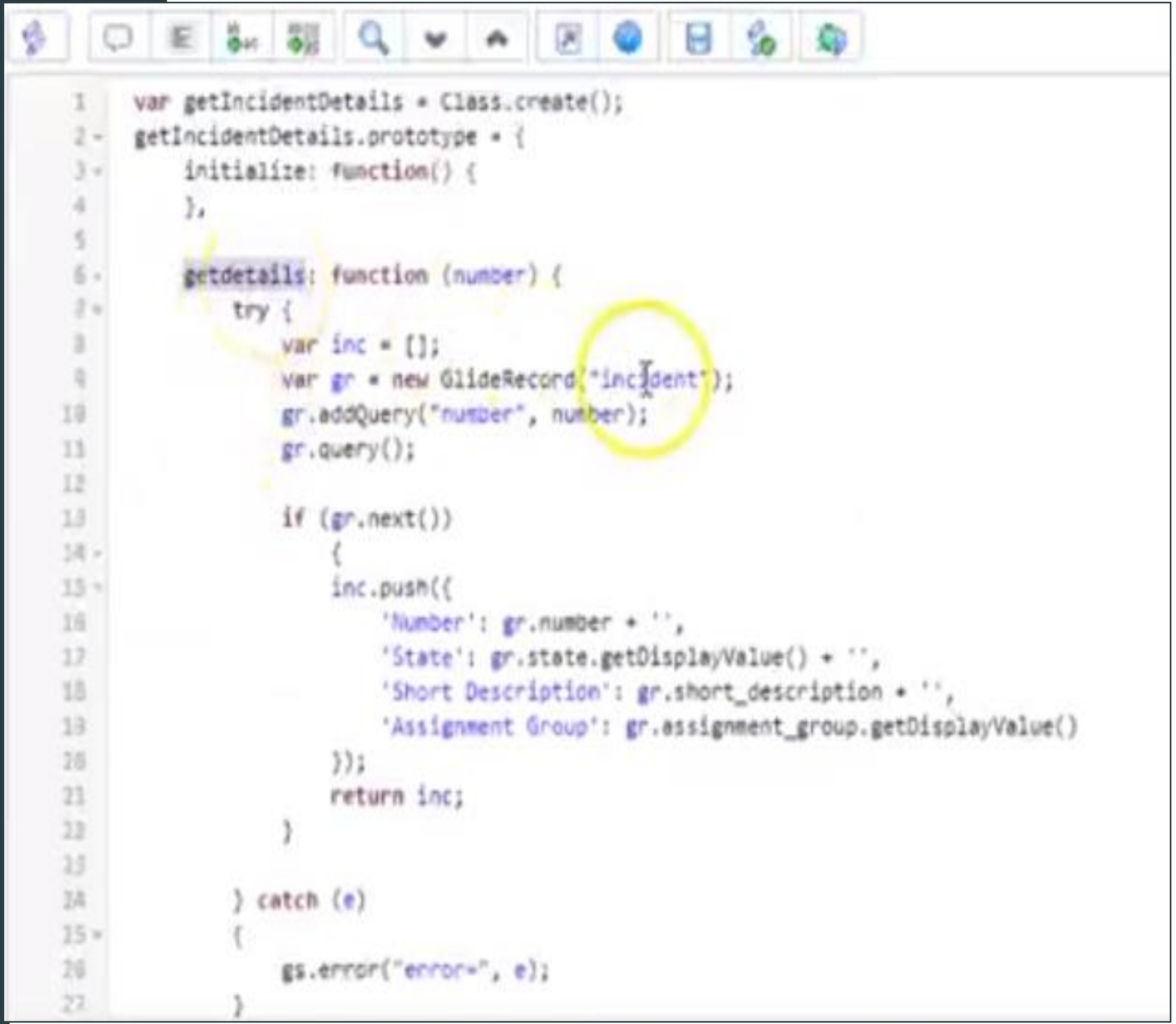


Name	getIncidentDetails	Application	Global
API Name	global.getIncidentDetails	Accessible from	This application scope only
Client callable		Active	✓
Description			
Script			
Protection policy			

```
var getIncidentDetails = Class.create();
getIncidentDetails.prototype = {
    initialize: function() {
    },

    type: 'getIncidentDetails'
};
```

Webservice – Scripted REST API

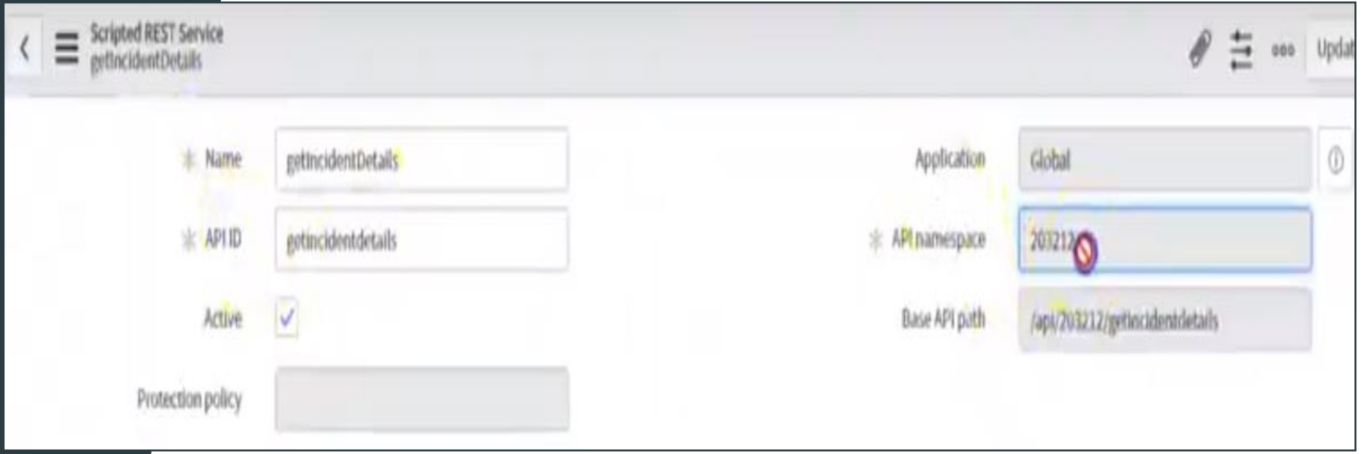
```
getdetails; function (number) {
        try {
                var inc = [];
                var gr = new GlideRecord("incident");
gr.addQuery("number",number);
gr.query();
If (gr.next())
        {
        inc.push((
        'Number' : gr.number + '',
        'State' : gr.state.getDisplayValue() + '',
'Short Description': gr.short_description + '',
'Assignment Group' : gr.assignment_group.getDisplayValue()
});
Return inc;
}
} catch (e)
{
gs.error("error=",e);
}
```



```
var getIncidentDetails = Class.create();
getIncidentDetails.prototype = {
    initialize: function() {
    },

    getdetails: function (number) {
        try {
            var inc = [];
            var gr = new GlideRecord("incident");
            gr.addQuery("number", number);
            gr.query();

            if (gr.next())
            {
                inc.push({
                    'Number': gr.number + '',
                    'State': gr.state.getDisplayValue() + '',
                    'Short Description': gr.short_description + '',
                    'Assignment Group': gr.assignment_group.getDisplayValue()
                });
                return inc;
            }

        } catch (e)
        {
            gs.error("error=", e);
        }
```

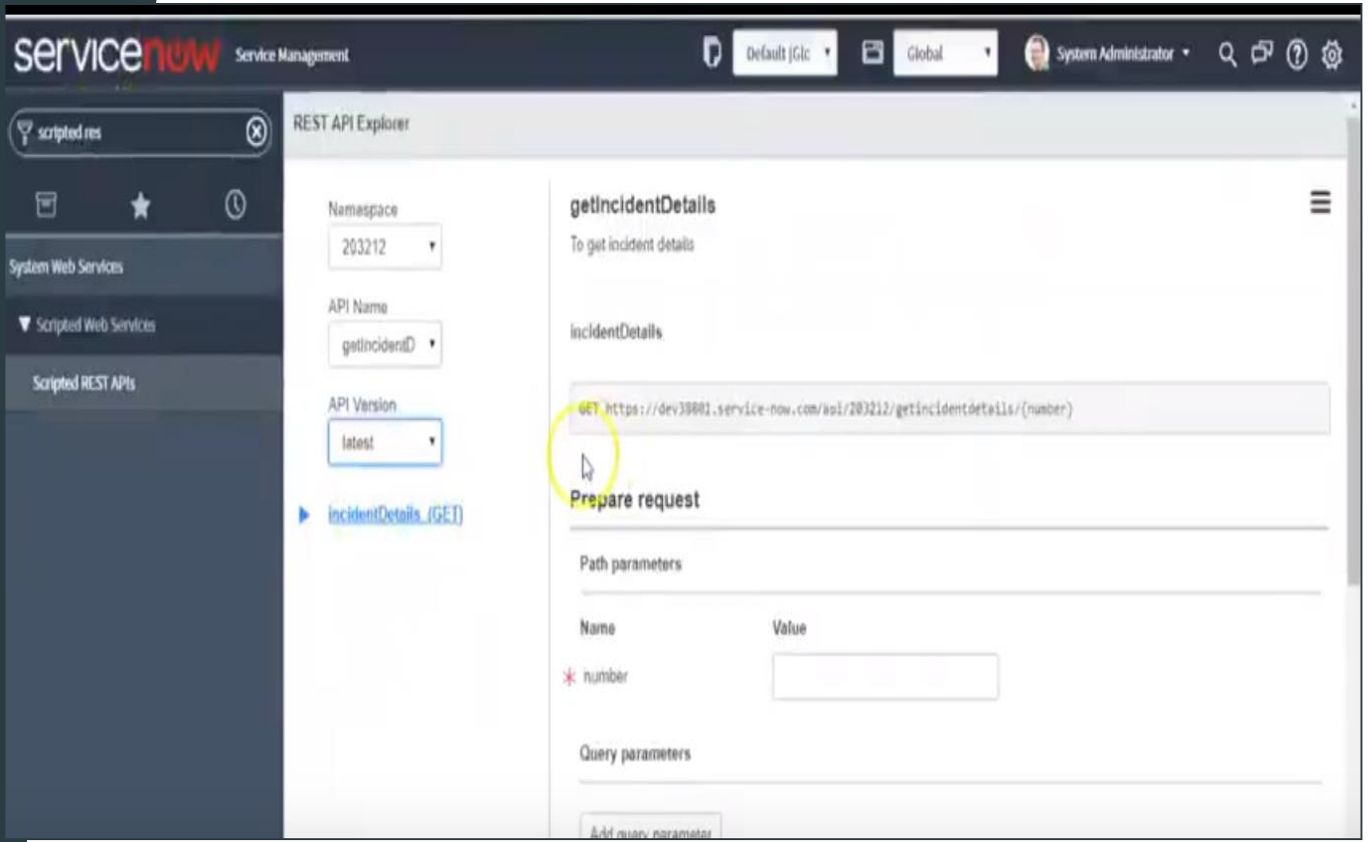
Webservice – Scripted REST API

- Lets use now REST API Explorer
- Navigate to System Web Services →REST API Explorer
- Use the same Namespace we used in Scripted REST Service



Scripted REST Service
getIncidentDetails

Name	getIncidentDetails	Application	Global
API ID	getincidentdetails	API namespace	203212
Active	✓	Base API path	/api/203212/getincidentdetails
Protection policy			



Namespace: 203212

API Name: getIncidentD

API Version: latest

incidentDetails (GET)

getIncidentDetails

To get incident details

incidentDetails

```
GET https://dev39001.service-now.com/api/203212/getincidentdetails/{number}
```

Prepare request

Path parameters

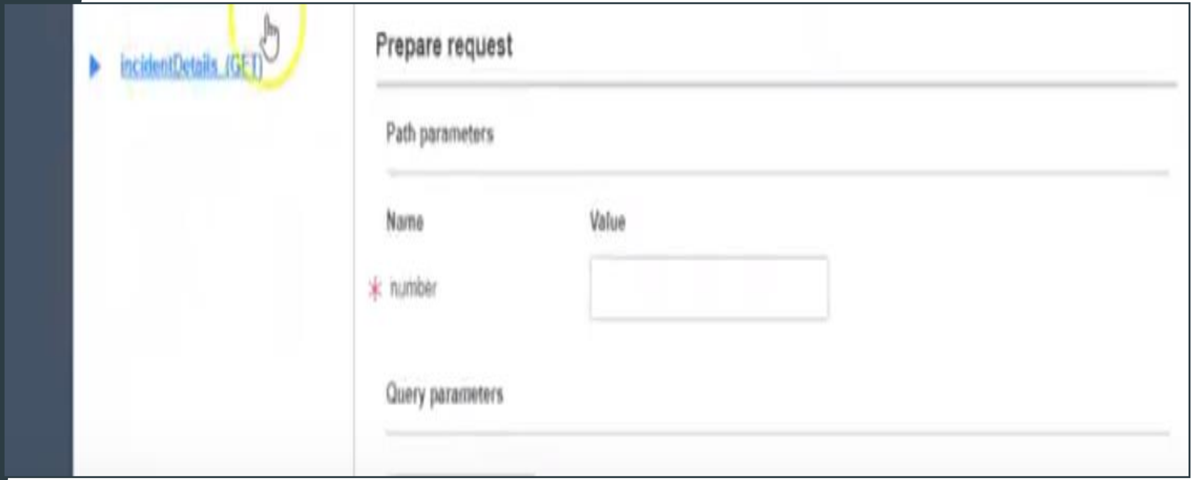
Name	Value
* number	

Query parameters

Add query parameter

Webservice Demo

- We can also observe the resource we defined i.e GET method of incidentDetails.
- Lets pass the incident number we referred in relative path
- Open Incident table and copy any number and pass the number in REST API Explorer



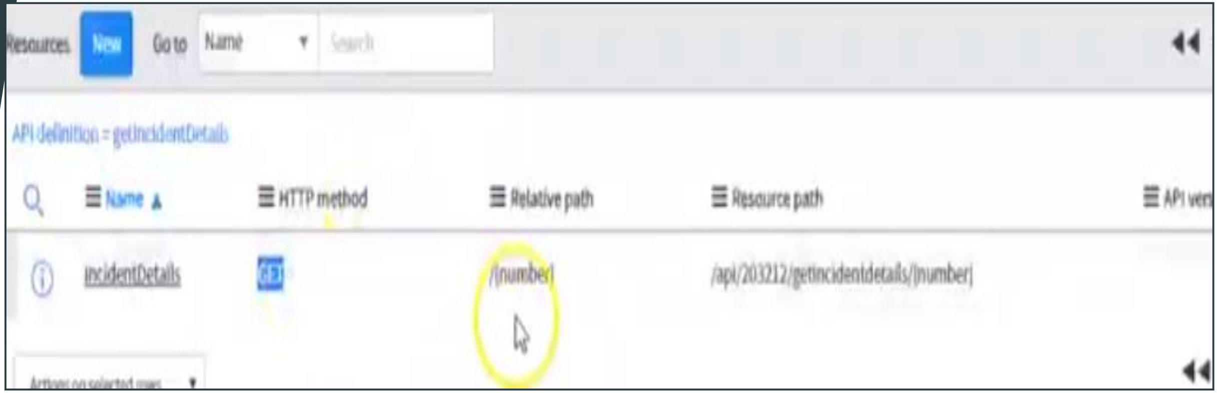
incidentDetails (GET)

Prepare request

Path parameters

Name	Value
* number	

Query parameters

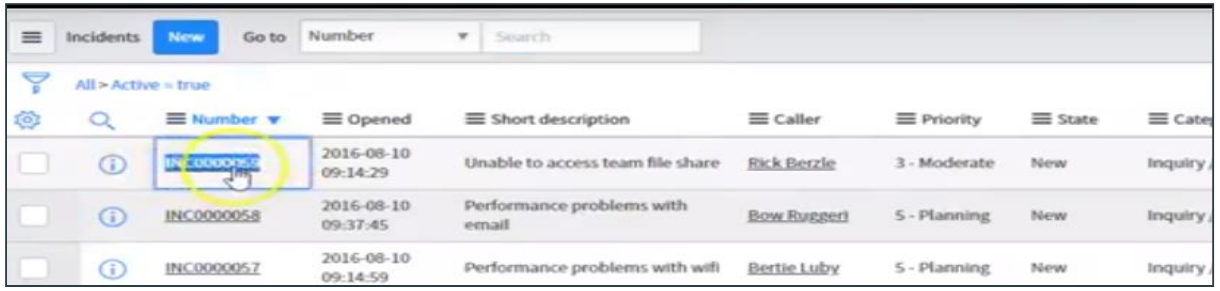


Resources New Go to Name Search

API definition = getIncidentDetails

Name	HTTP method	Relative path	Resource path	API ver
incidentDetails	GET	/{number}	/api/203212/getincidentdetails/{number}	

Actions on selected rows



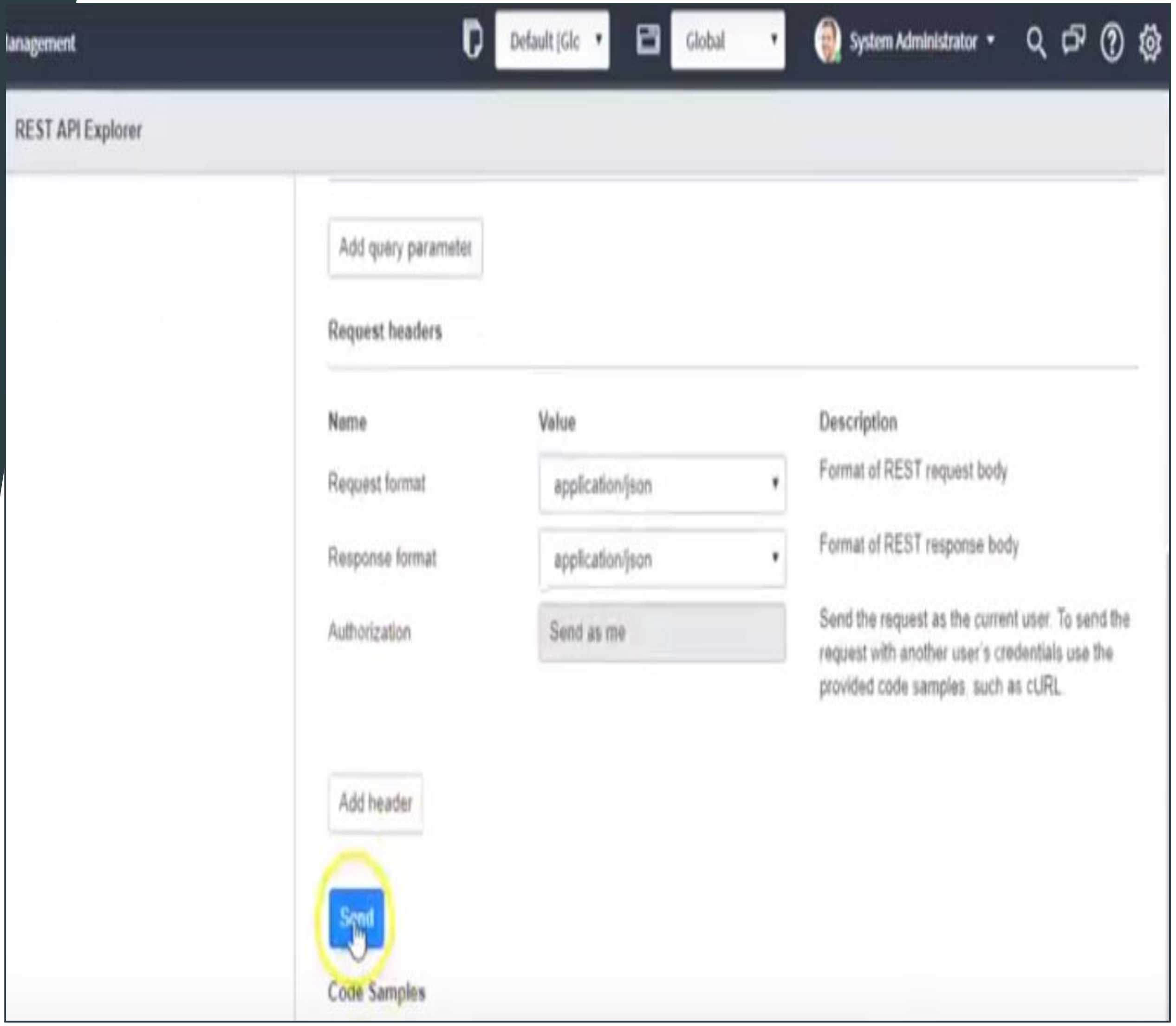
Incidents New Go to Number Search

All > Active = true

Number	Opened	Short description	Caller	Priority	State	Cate
INC0000059	2016-08-10 09:14:29	Unable to access team file share	Rick Berzle	3 - Moderate	New	Inquiry
INC0000058	2016-08-10 09:37:45	Performance problems with email	Bow Ruggeri	5 - Planning	New	Inquiry
INC0000057	2016-08-10 09:14:59	Performance problems with wifi	Bertie Luby	5 - Planning	New	Inquiry

Webservice Demo

- Paste in the Explorer and click Send button.



anagement

Default [Glc

Global

System Administrator

REST API Explorer

Add query parameter

Request headers

Name	Value	Description
Request format	application/json	Format of REST request body
Response format	application/json	Format of REST response body
Authorization	Send as me	Send the request as the current user. To send the request with another user's credentials use the provided code samples, such as cURL.

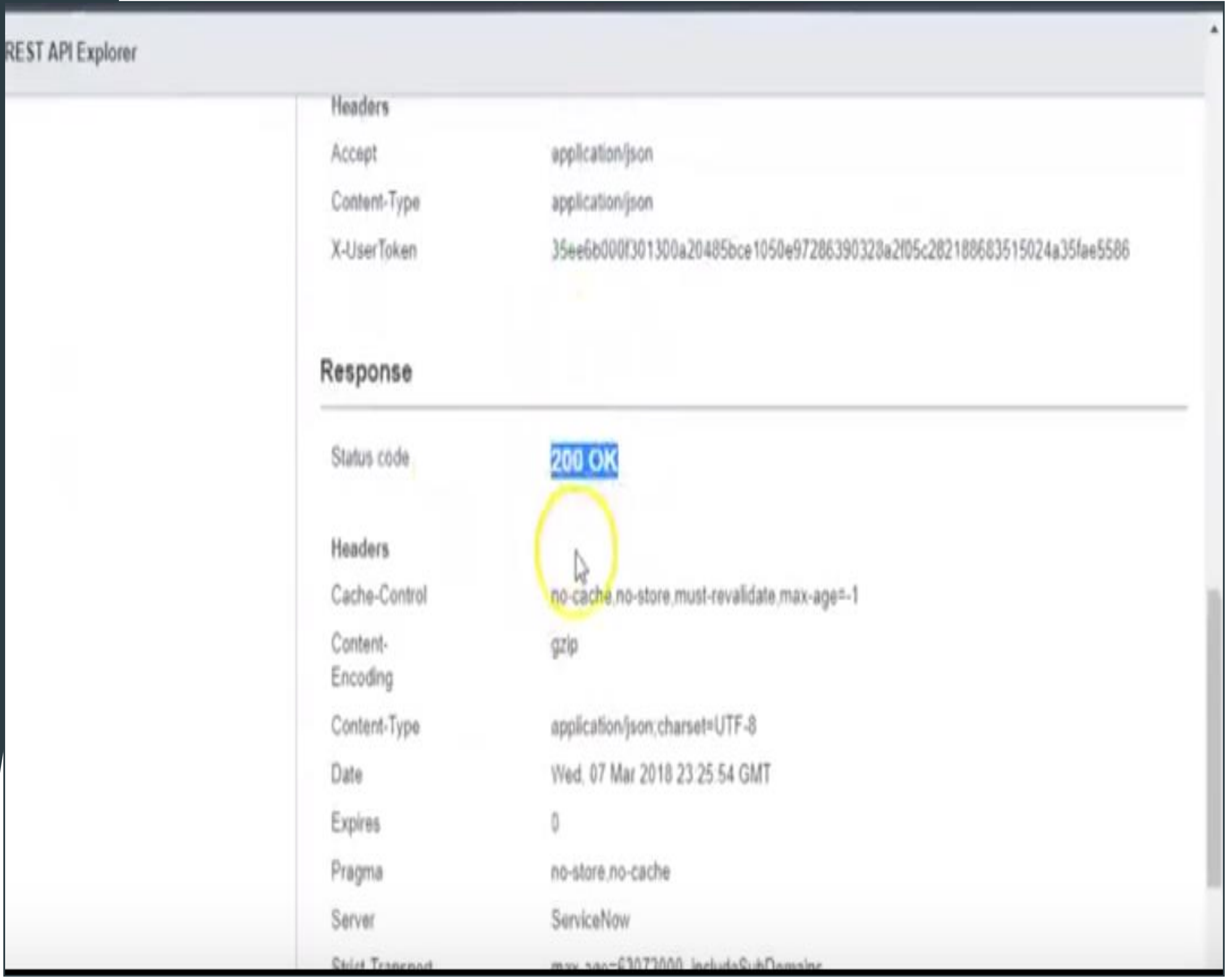
Add header

Send

Code Samples

Webservice Demo

- We can observe success message with status code 200 OK


REST API Explorer

Headers

Accept	application/json
Content-Type	application/json
X-UserToken	35ee6b000f301300a20485bce1050e97286390328a2f05c282188683515024a35fae5586

Response

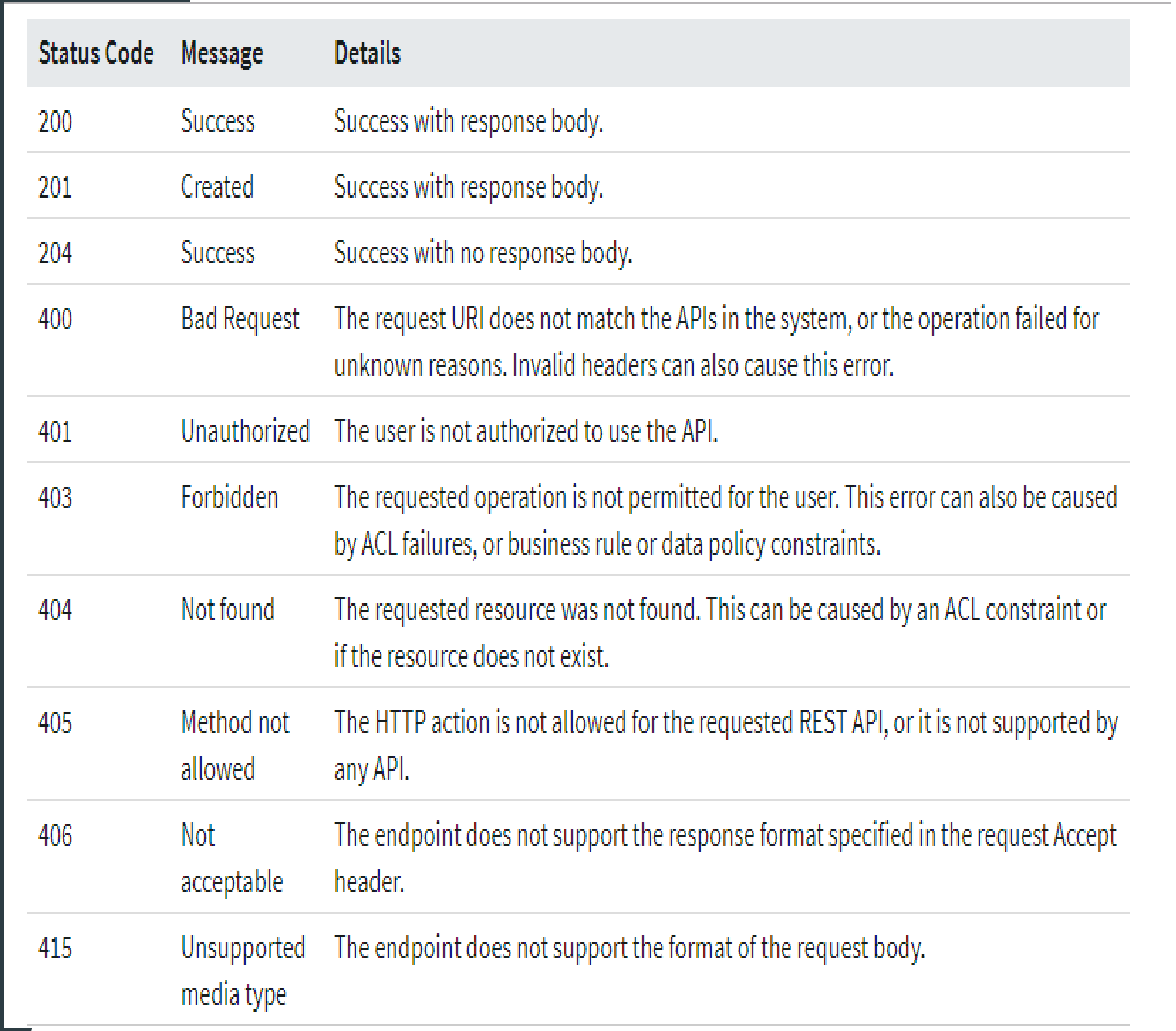
Status code	200 OK

Headers

Cache-Control	no-cache,no-store,must-revalidate,max-age=-1
Content-Encoding	gzip
Content-Type	application/json;charset=UTF-8
Date	Wed, 07 Mar 2018 23:25:54 GMT
Expires	0
Pragma	no-store,no-cache
Server	ServiceNow
Strict-Transport	max-age=63072000; includeSubDomains

Webservice Demo

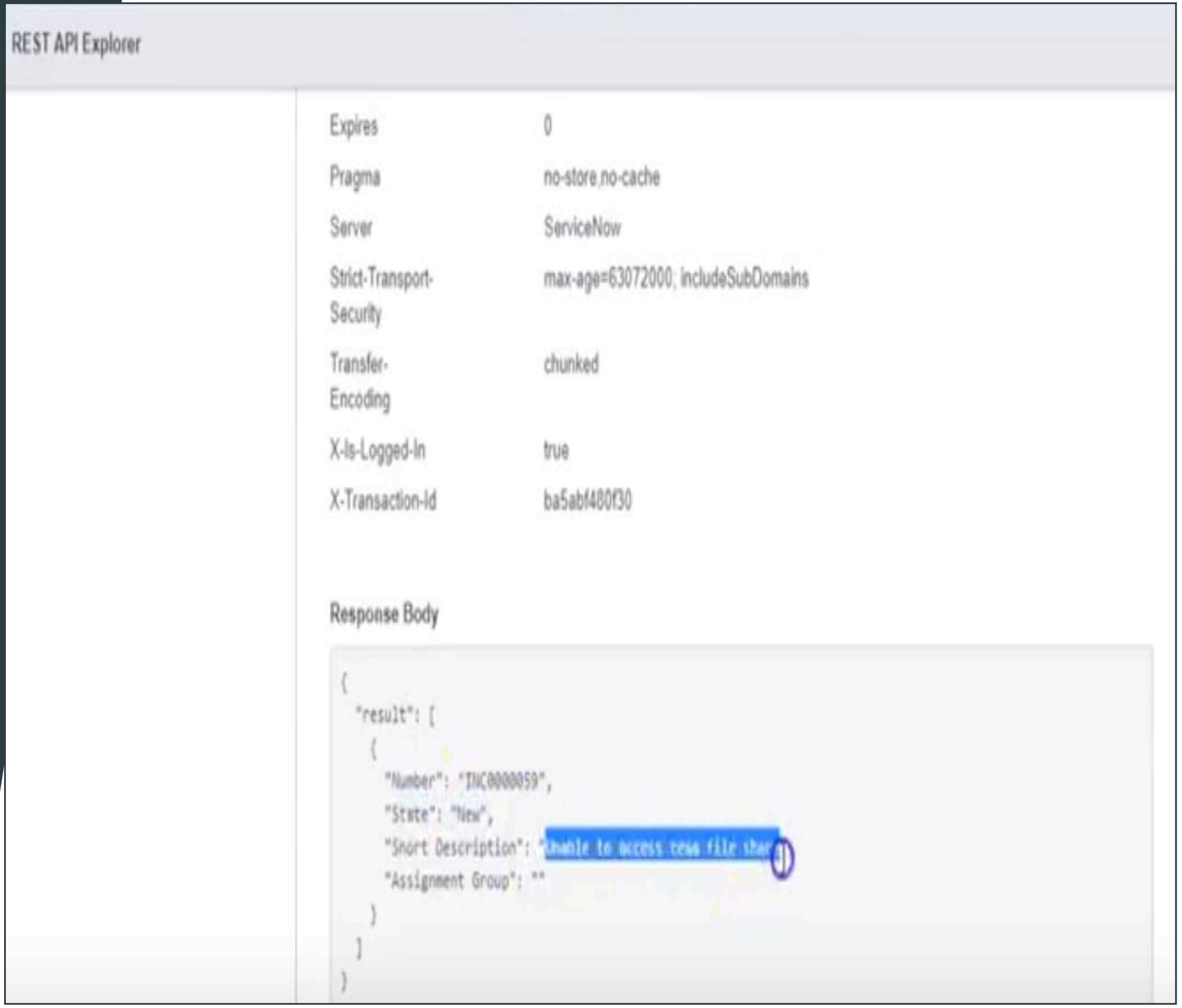
- REST Messages sent to a ServiceNow instance return a specific HTTP response code.



Status Code	Message	Details
200	Success	Success with response body.
201	Created	Success with response body.
204	Success	Success with no response body.
400	Bad Request	The request URI does not match the APIs in the system, or the operation failed for unknown reasons. Invalid headers can also cause this error.
401	Unauthorized	The user is not authorized to use the API.
403	Forbidden	The requested operation is not permitted for the user. This error can also be caused by ACL failures, or business rule or data policy constraints.
404	Not found	The requested resource was not found. This can be caused by an ACL constraint or if the resource does not exist.
405	Method not allowed	The HTTP action is not allowed for the requested REST API, or it is not supported by any API.
406	Not acceptable	The endpoint does not support the response format specified in the request Accept header.
415	Unsupported media type	The endpoint does not support the format of the request body.

Webservice Demo

- We can observe details of incident ticket in Response received



REST API Explorer

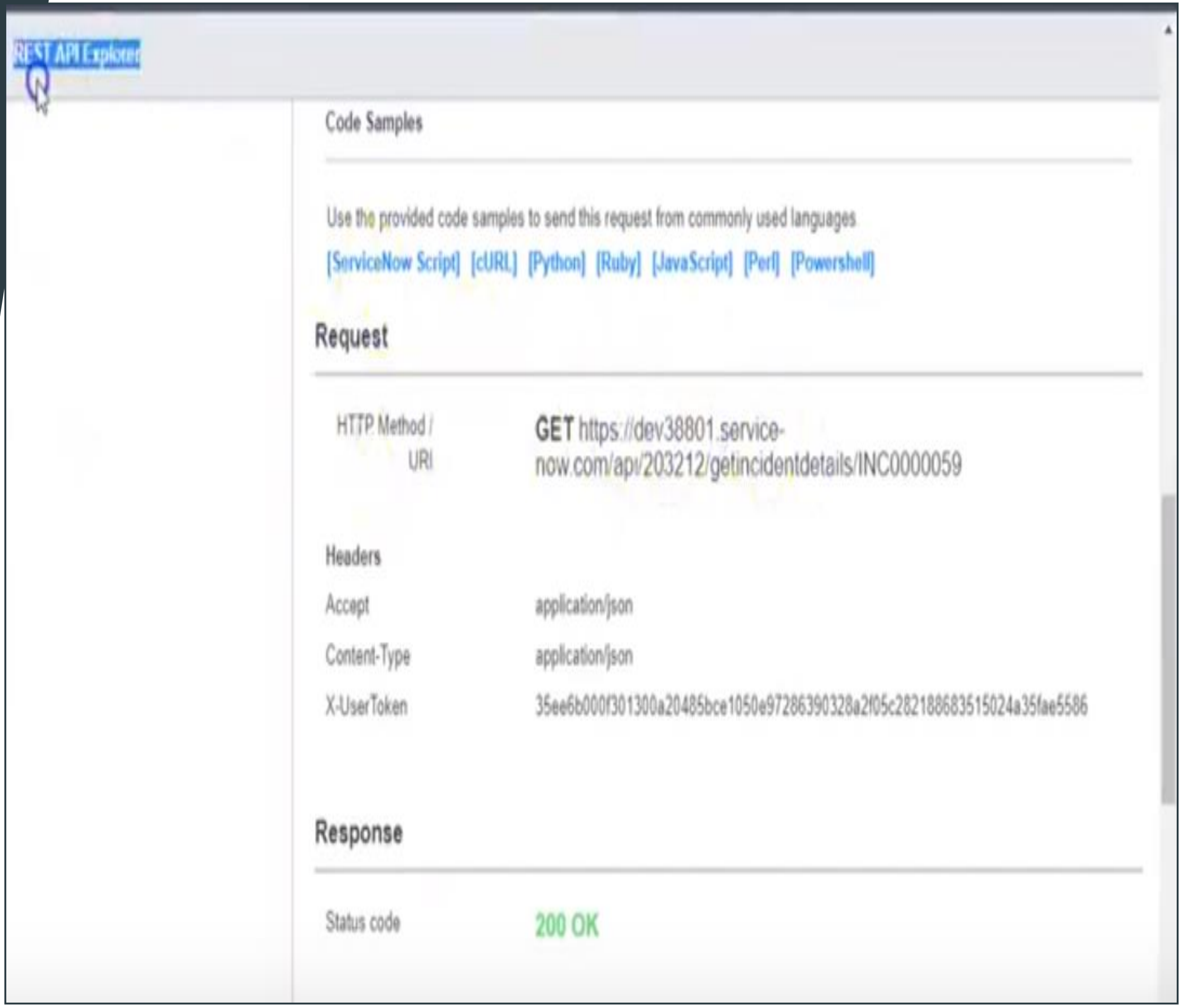
Expires	0
Pragma	no-store,no-cache
Server	ServiceNow
Strict-Transport-Security	max-age=63072000; includeSubDomains
Transfer-Encoding	chunked
X-Is-Logged-In	true
X-Transaction-Id	ba5abf480f30

Response Body

```
{
  "result": [
    {
      "Number": "INC0000059",
      "State": "New",
      "Short Description": "Unable to access team file shar
      "Assignment Group": ""
    }
  ]
}
```

Webservice Demo

- We tested the functionality of REST API on the instance we are working on.
- We can also test the functionality of this REST API defined in the instance through tools for particular browser
- For Firefox browser there is RESTClient tool for testing functionality and for Chrome there is POSTMAN to test the functionality



Code Samples

Use the provided code samples to send this request from commonly used languages

[ServiceNow Script] [cURL] [Python] [Ruby] [JavaScript] [Perl] [Powershell]

Request

HTTP Method / URI	**GET** https://dev38801.service-now.com/api/203212/getincidentdetails/INC0000059

Headers

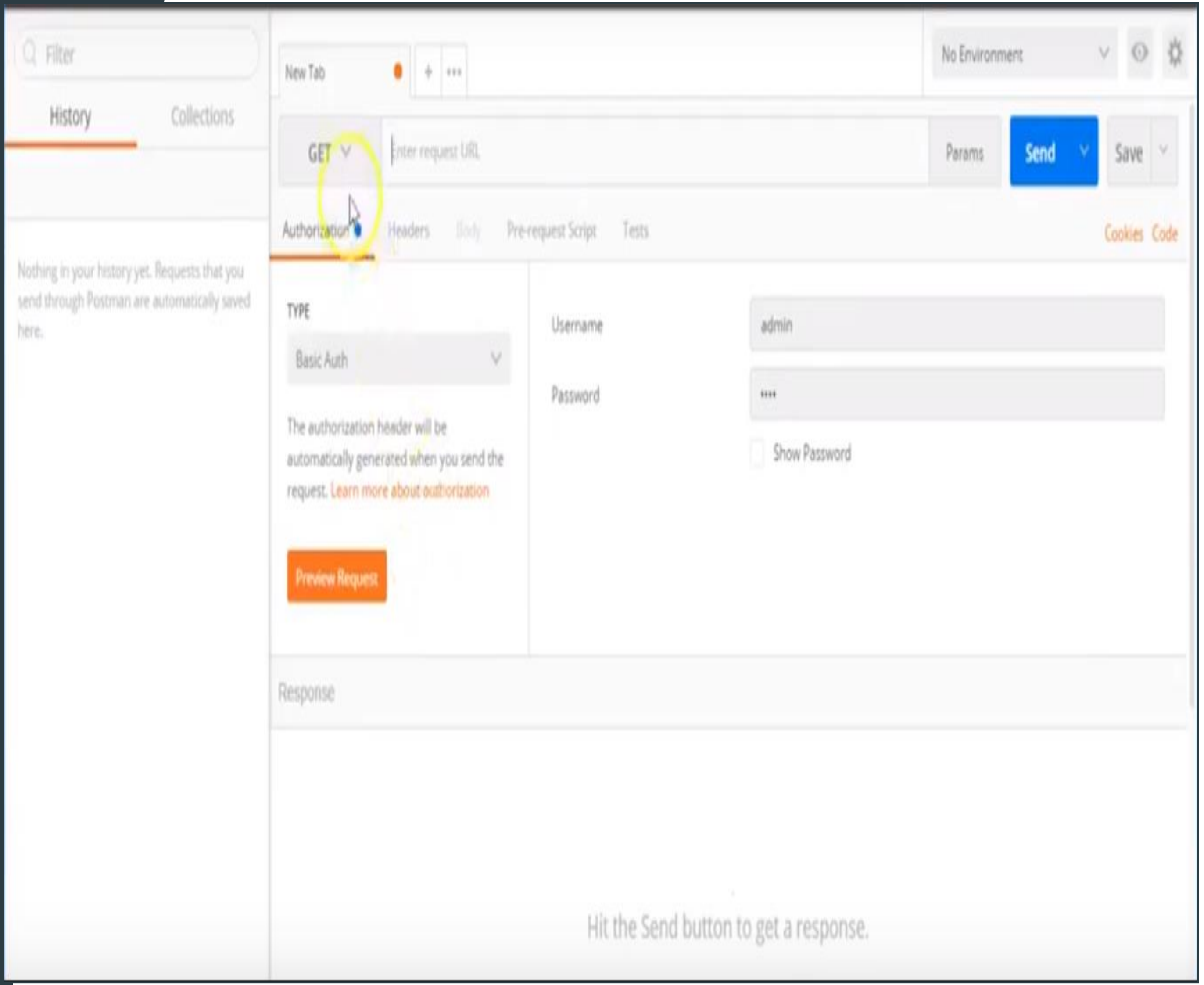
Accept	application/json
Content-Type	application/json
X-UserToken	35ee6b000f301300a20485bce1050e97286390328a2f05c28218868351 5024a35fae5586

Response

Status code	200 OK

Webservice Demo

- **Download https://www.getpostman.com/downloads/ for Chrome.**
- **Use the Username and Password we used for the SNOW instance with admin and password**
- **Select the method we used in scripted web method API**



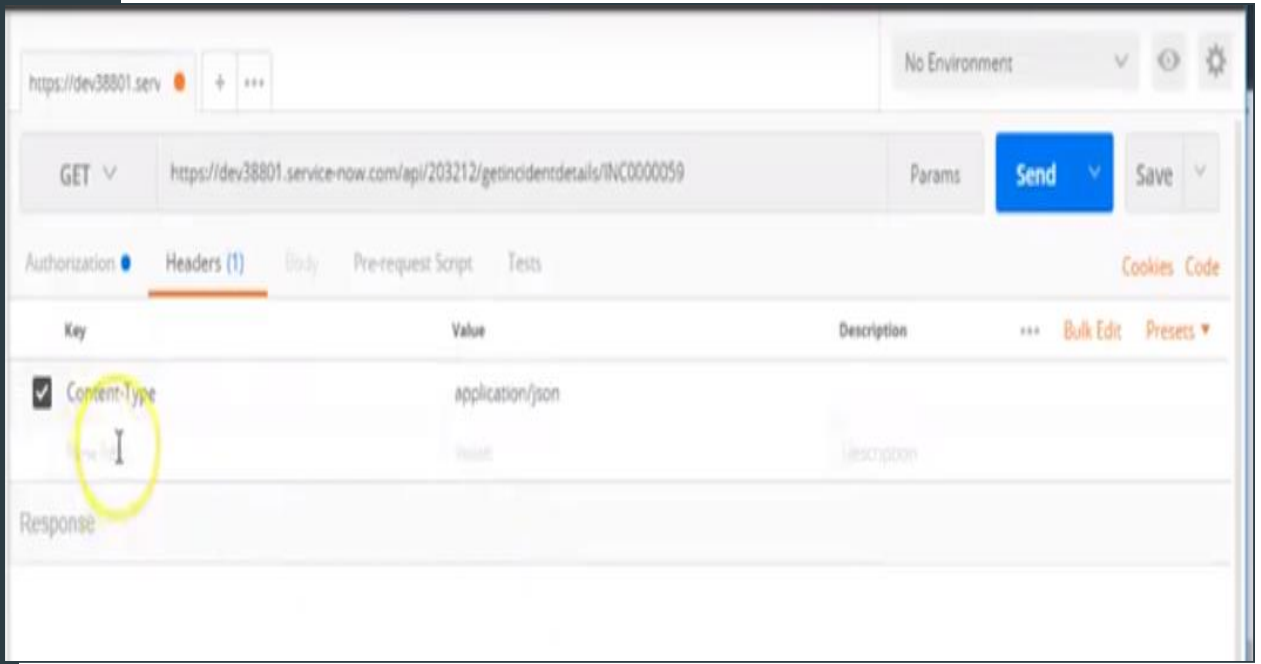
Nothing in your history yet. Requests that you send through Postman are automatically saved here.

The authorization header will be automatically generated when you send the request. Learn more about authorization

Hit the Send button to get a response.

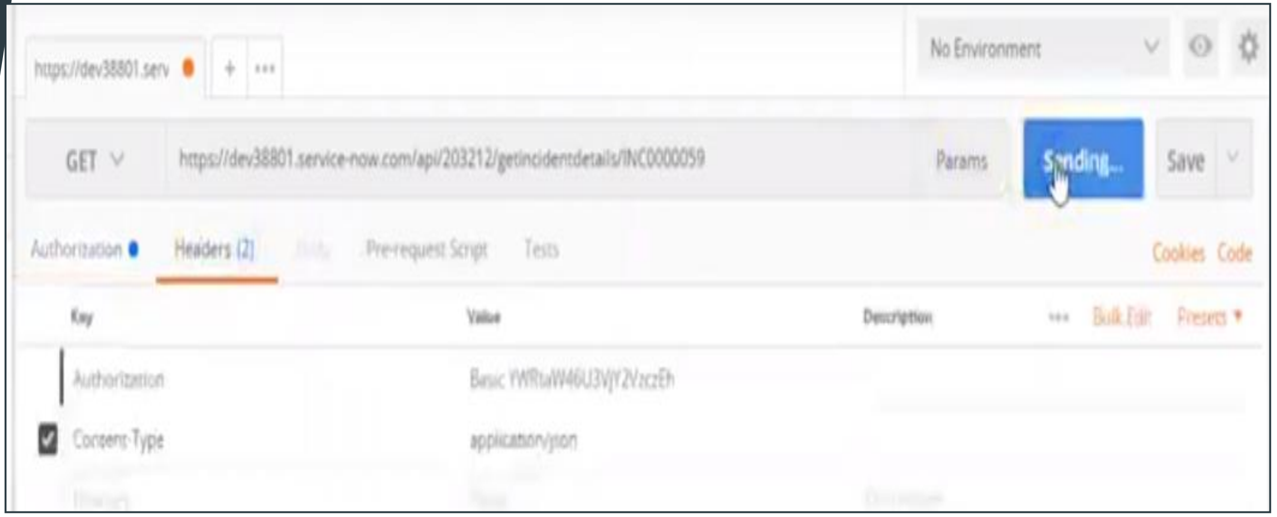
Webservice Demo

- Pass the URL which we observed in REST API Explorer
- Type as Basic Auth and give admin access to the instance
- In the header pass Content-Type as application/json
- Click on Send button.



GET https://dev38801.service-now.com/api/203212/getincidentdetails/INC0000059

Key	Value	Description
Content-Type	application/json	

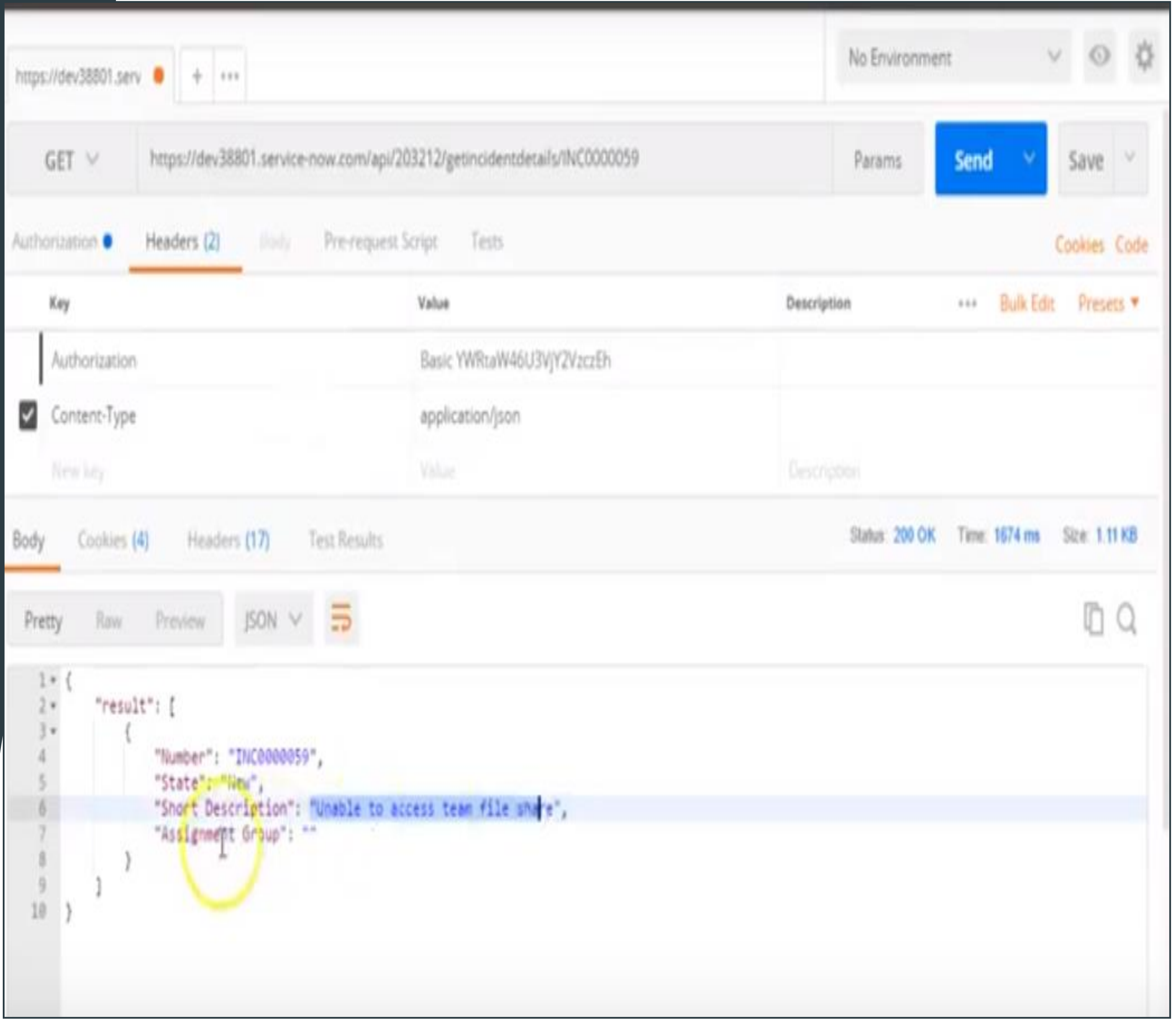


GET https://dev38801.service-now.com/api/203212/getincidentdetails/INC0000059

Key	Value	Description
Authorization	Basic YWRtaW46U3VjY2VzczEh	
Content-Type	application/json	

Webservice Demo

- We can retrieve all the details regarding incident tickets
- Thus this is the method to use Postman to test the scripted REST API functionality



https://dev38801.service-now.com/api/203212/getincidentdetails/INC0000059

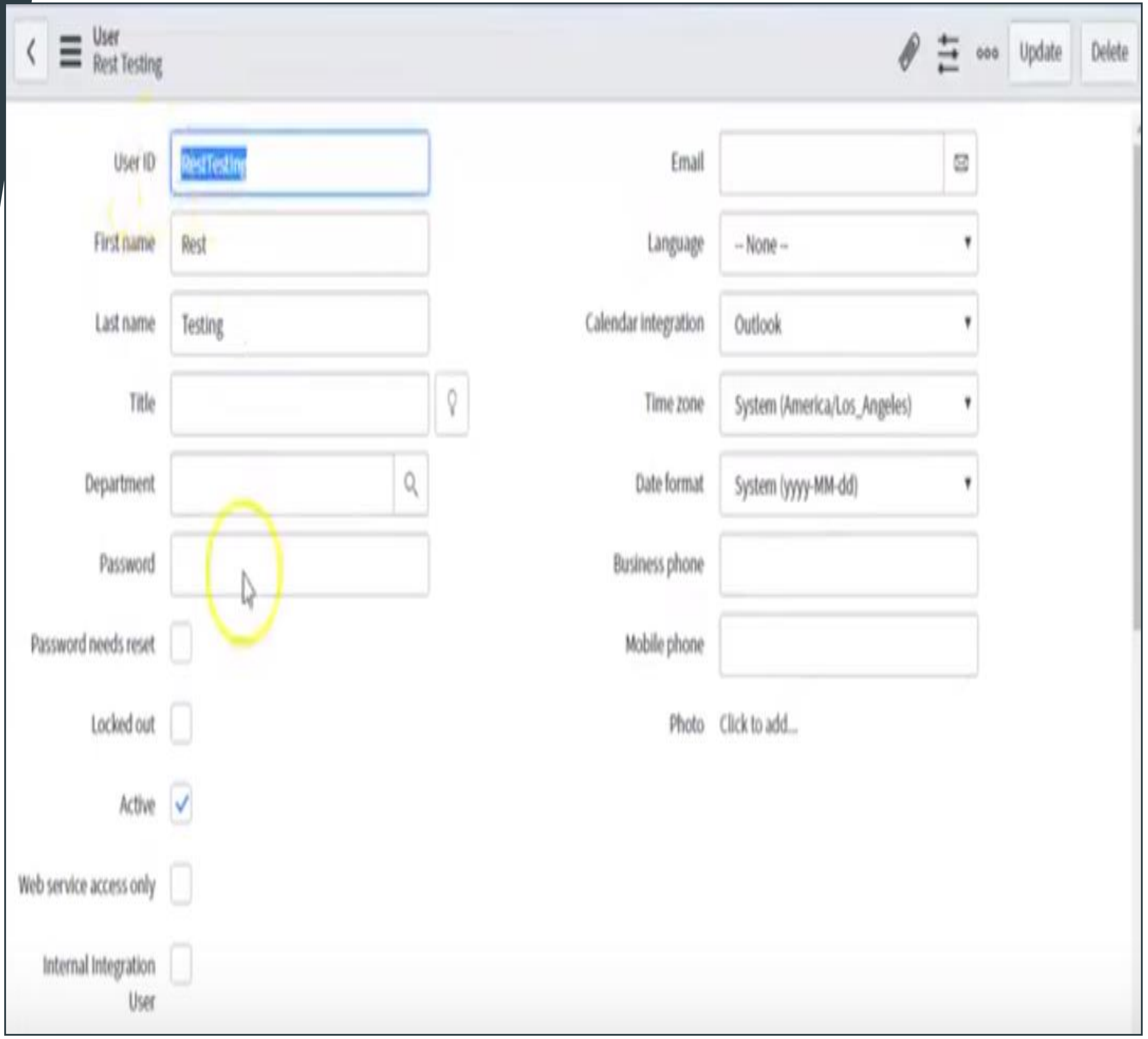
Key	Value	Description
Authorization	Basic YWRtaW46U3VjY2VzczEh	
✓ Content-Type	application/json	
New key	Value	Description

Status: 200 OK Time: 1674 ms Size: 1.11 KB

```
{
    "result": [
        {
            "Number": "INC0000059",
            "State": "New",
            "Short Description": "Unable to access team file share",
            "Assignment Group": ""
        }
    ]
}
```

Webservice Demo

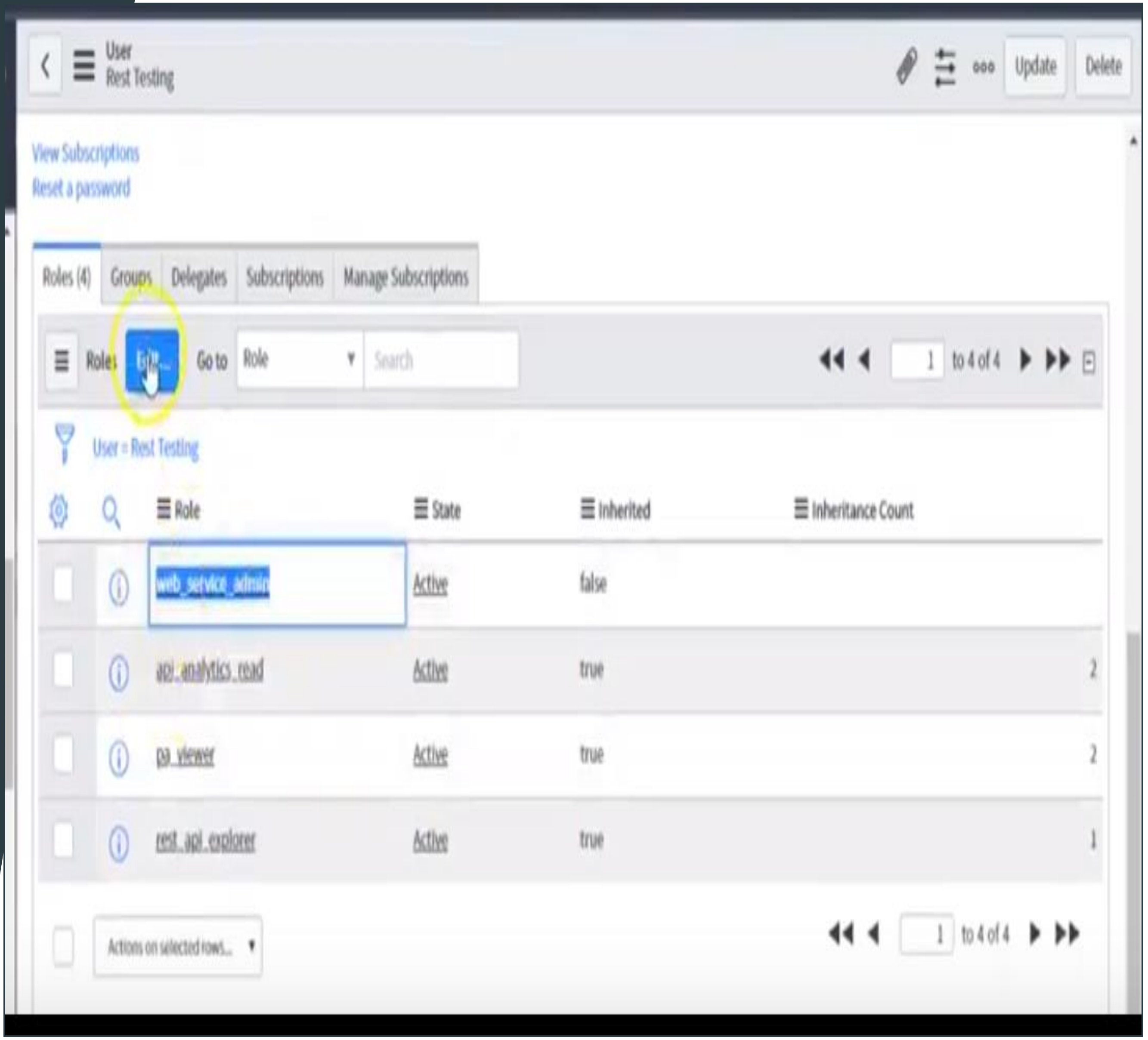
- Its important we must not give admin credentials to test REST API functionality for POSTMAN tool as it is exposing entire instance.
- Create a user for REST Testing having Web service access



User ID	RestTesting	Email	
First name	Rest	Language	-- None --
Last name	Testing	Calendar integration	Outlook
Title		Time zone	System (America/Los_Angeles)
Department		Date format	System (yyyy-MM-dd)
Password		Business phone	
Password needs reset	☐	Mobile phone	
Locked out	☐	Photo	Click to add...
Active	✓		
Web service access only	☐		
Internal Integration User	☐		

Webservice

- Give the role of web_service_admin to the user.



User
Rest Testing

View Subscriptions
Reset a password

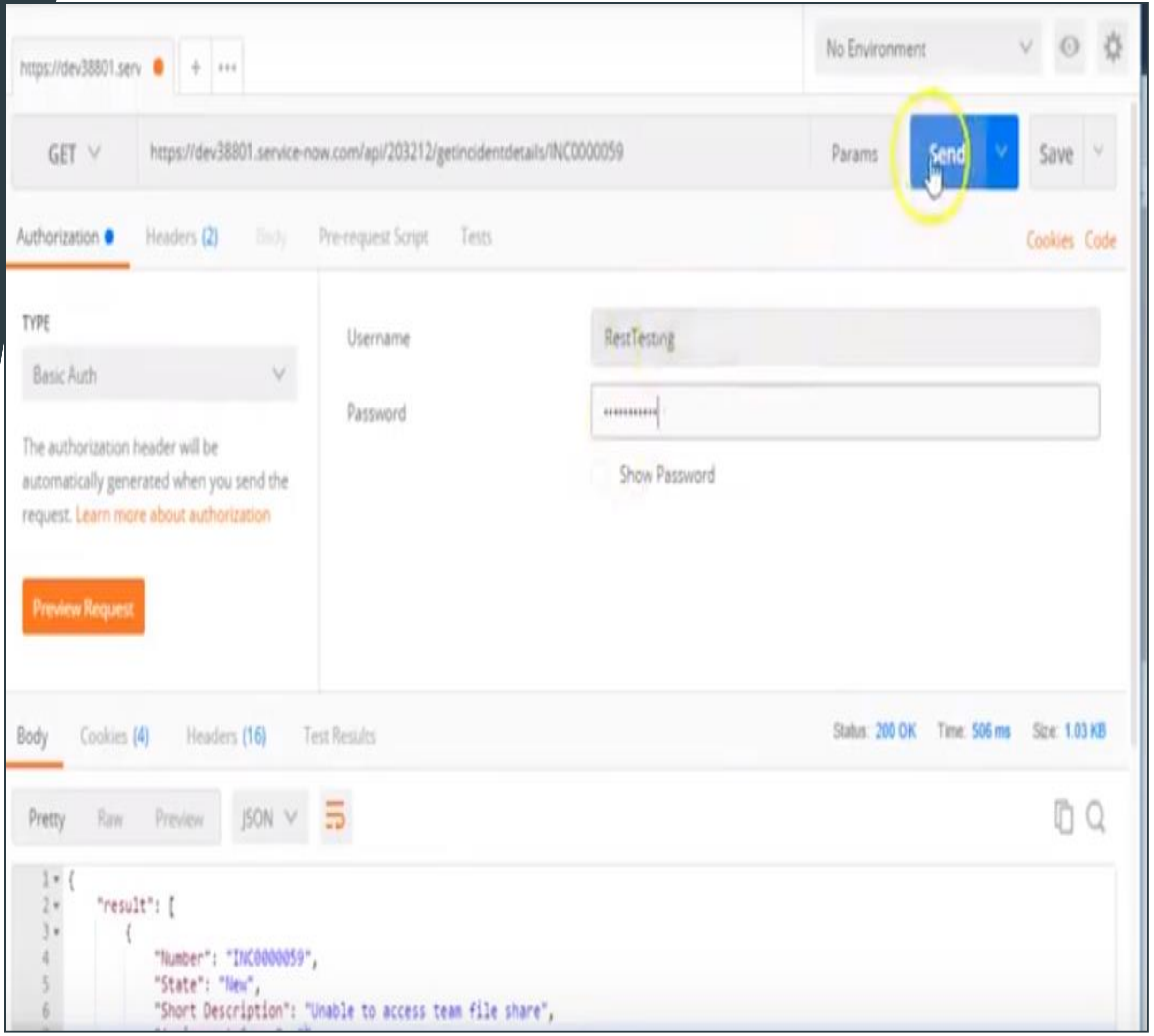
Roles (4) | Groups | Delegates | Subscriptions | Manage Subscriptions

User = Rest Testing

Role	State	Inherited	Inheritance Count
web_service_admin	Active	false	
api_analytics_read	Active	true	2
pa_viewer	Active	true	2
rest_api_explorer	Active	true	1

Webservice

- Use that Username and Password updated as part of that user



GET https://dev38801.service-now.com/api/203212/getincidentdetails/INC0000059

TYPE: Basic Auth

The authorization header will be automatically generated when you send the request. Learn more about authorization

Username: RestTesting

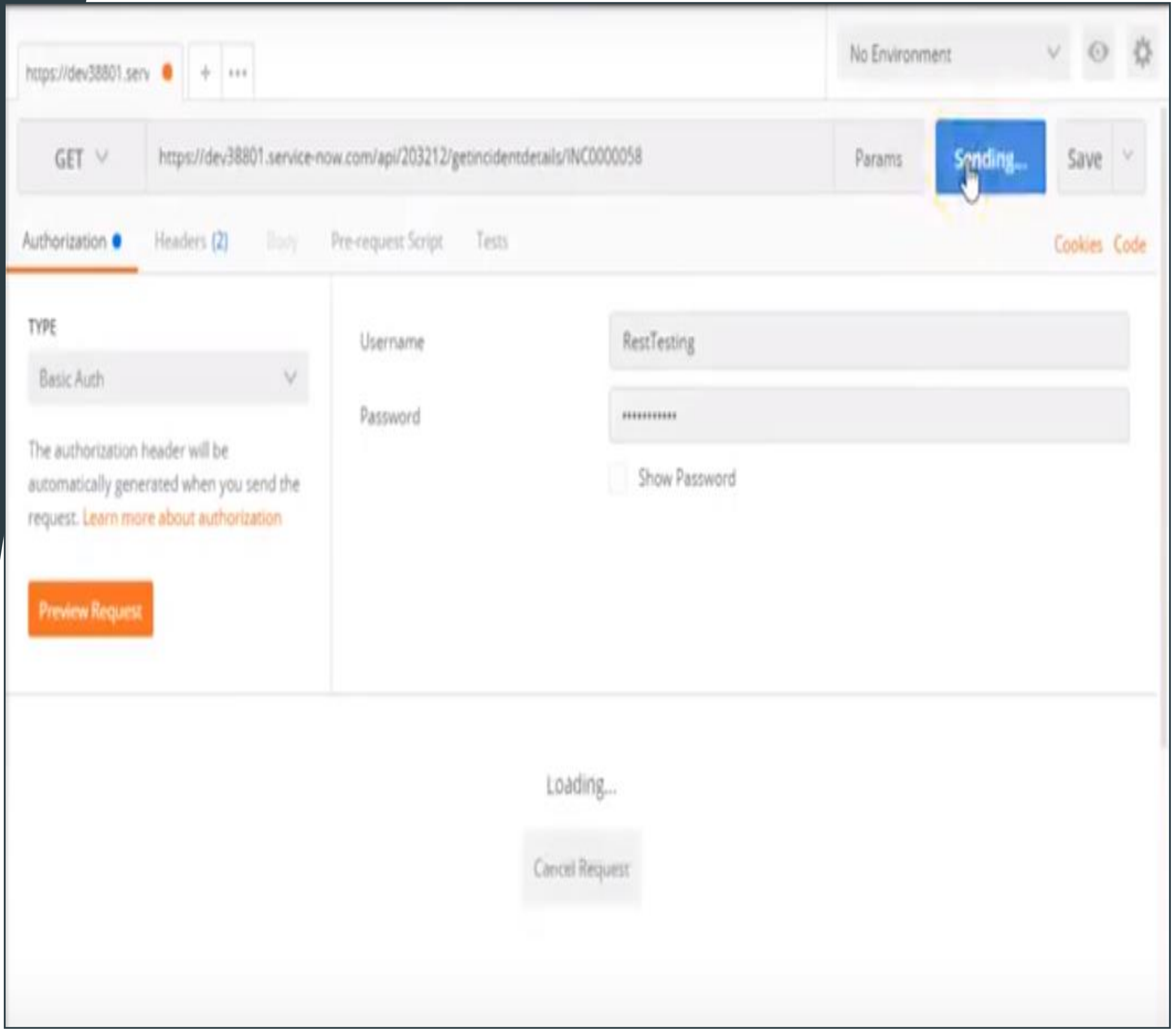
Password: ••••••••••

Status: 200 OK Time: 506 ms Size: 1.03 KB

```
{
    "result": [
        {
            "Number": "INC0000059",
            "State": "New",
            "Short Description": "Unable to access team file share",
```

Webservice

- **Now change the incident number and click send button**



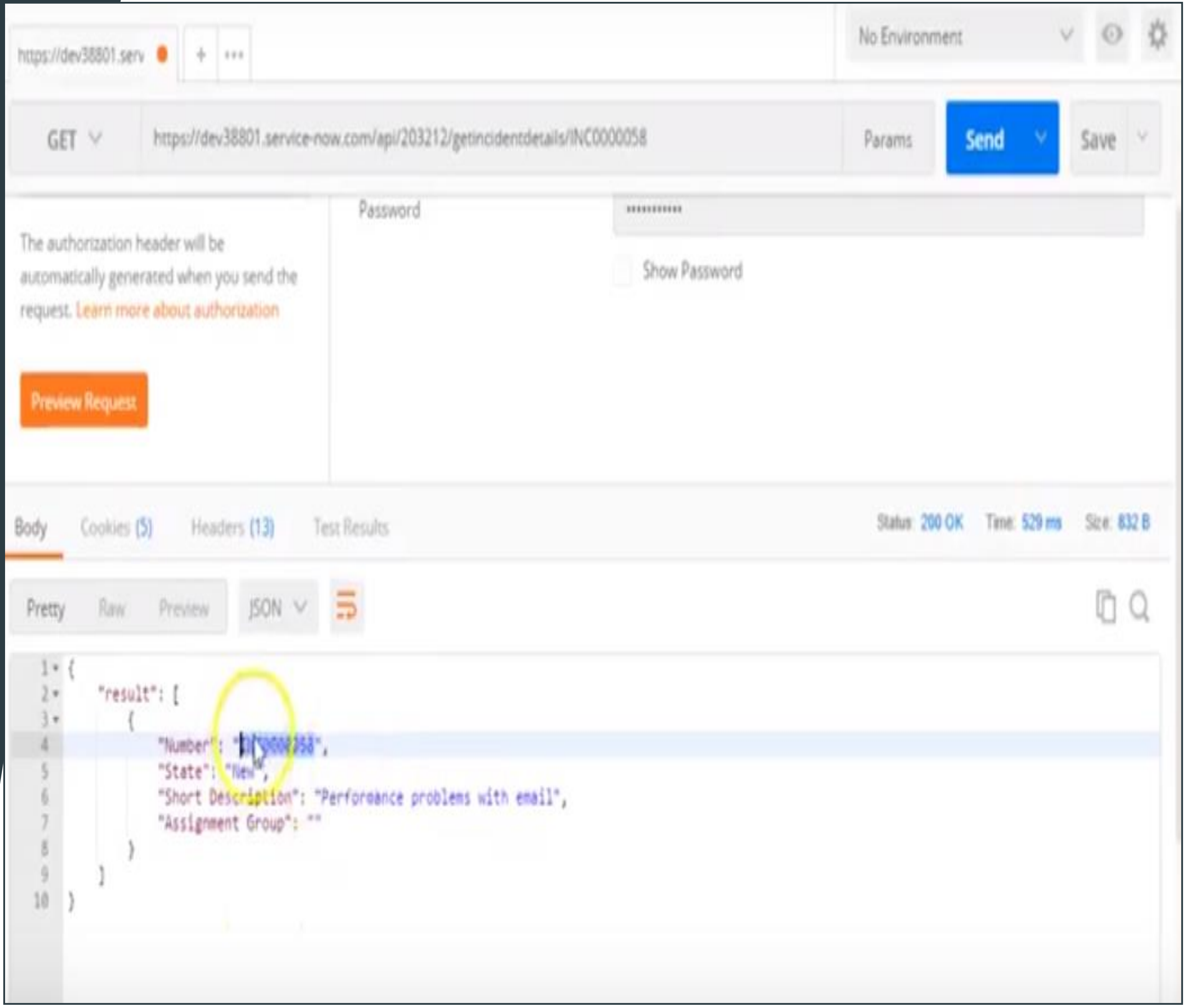
https://dev38801.service-now.com/api/203212/getincidentdetails/INC0000058

The authorization header will be automatically generated when you send the request. Learn more about authorization

Username: RestTesting
Password: ••••••••••

Webservice

- **We would be getting all details w.r.t to incident mentioned using the REST testing user credentials.**



https://dev38801.service-now.com/api/203212/getincidentdetails/INC0000058

The authorization header will be automatically generated when you send the request. Learn more about authorization

Password ••••••••••

Show Password

Status: 200 OK Time: 529 ms Size: 832 B

```
{
    "result": [
        {
            "Number": "INC0000058",
            "State": "New",
            "Short Description": "Performance problems with email",
            "Assignment Group": ""
        }
    ]
}
```

Thankyou