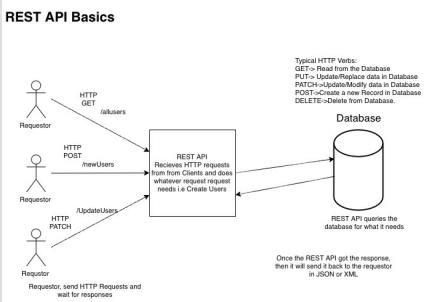
ServiceNow REST - Cheat Sheet Cheat Sheet

by Bibin Gokuldas (bibingokuldas) via cheatography.com/69673/cs/17659/

REST API Basics



Scripted REST APIs

API URIs	This part the for the scripted rest has to define name_space, api_id, resource_path, version
API Query Parameter	When defining a scripted resource, which parameter is mandatory for the request can also be defined
API Error Objects*	Scripted REST APIs provide multiple ways to send an error in a response to a requesting client.

*Multiple error objects are available in scripted REST API scripts to report error information to requesting clients.All scripted REST API error objects use the sn_ws_err namespace.
Error objects available are 400,404,406,409,415

REST API Best Practices

Follow REST API Conventions	REST API conventions define specific behaviour for each type of method. For ex : *GET* : to Query the data, *POST* : to create Data,*PUT and PATCH* : to Update data, *DELETE* : to delete records.
Use Versioning to control changes to API	Use versioning to implement new functionalities, so that the existing functionalities will not impact
Return an Informative HTTP Status code	Return a status code, which inform the requestor about the success and failures(defined in the response codes section)
Return useful error information	Provide the requestor with enough information of why the failure occurred. Error message is a mix of error message and error code

REST API Best Practices (cont)

Enforce and test Access Controls	Enforce existing access controls and require additional access to modify the data. Use the GlideRecordSecure* API in scripted RESI services
Build Tests to verify functionality	Tests should validate the response code, headers, and body content as appropriate for each resource you implement. You can also use tests to validate authentication requirements, and to confirm that errors return useful responses.

*GlideRecordSecure API Ensure that the ACLs are defined on the underlying data are applied to the requested user.

REST Security

How is REST API Secure?

> The REST API uses basic authentication or OAuth to enforce access controls to web resources. ACLs are defined on tables to restrict the data viewership.

Will all tables be available for the REST API Access?

> By default, Yes. All tables including system tables, and scoped tables are available.

How can I restrict a table Access through web services?

> In the table properties, uncheck the option for Allow access to this table via web services.

Does REST API support CORS?

> Cross Origin Resource Security is supported.

How can I Define CORS Rules?

> CORS Rules can be defined in sys_cors_rule. Which allows to specify a domain and Selection of methods to expose.

How to disable CORS Support for Instance?

> CORS support on instance is defined by glide.rest.cors.enabled set it to false for No CORS

Can I use OAuth with REST?

> Yes, use OAuth token for REST Requests

Can I use MFA with REST?

> Yes again, with a REST Request, if MFA is enabled then append token to end of users password ex:userid:passwrdtoken. Encode using base64 encoding

By **Bibin Gokuldas**
(bibingokuldas)
cheatography.com/bibingokuldas/

Published 29th October, 2018.
Last updated 20th November, 2018.

Building Blocks REST API

API	API allows to select a specific Application Programming interface, which is available in SNOW	Ex TableAPI,AggregateAPI
Namespace	REST APIs provided by ServiceNow has **now** namespace	Scripted REST may use a different one
Method	REST enables the use of few methods like **GET**,**POST**,**DELETE**,**PATCH**	Not all the APIs available from ServiceNow would have all the methods available
Request Header	Allows to specify a header for the Integration	Can add Custom header as per requirement
Query parameter	Allows to specify an encoded query for the REST Call	Can add more query parameters or even a sys_id for some methods

You can prepare the sample request using the REST API Explorer in ServiceNow. ServiceNow REST URI looks like this <LINKFORSNOW>/api/now/apiname/. For ex : if we are using a table API for POST then the link look something like below : POST <LINKOFServiceNow>/api/now/table/tablename

REST API Rate Limit

To prevent excessive inbound REST API requests, set rules that limit the number of inbound REST API requests processed per hour.

There is an option to create Rate Limit for users with specific roles, or for all users.

The table for creating rate limit is **sys_rate_limit_rules**.

In the basic Response Header, the Rate limit would be specified for ex : x-RateLimit-Limit -->10

REST API Headers

Accept	application/json, application/xml
Content-Type	pplication/json, application/xml

By design, POST, PUT, PATCH, and DELETE operations required to provide both headers.

GET operations require only the Accept header.

There is an option override the HTTP method, such as GET or POST, by setting the X-http-method-override header.

REST API Response Codes

200	Success	Success with Response Body
201	Created	Success with Response Body
204	Success	Success with Response Body

REST API Response Codes (cont)

400	Bad Request	The Request URI can't match the API.	Invalid headers, or API incorrect
401	Unauthorized	The User is not authorized to use API	
403	Forbidden	The Operation requested is not permitted for the user	ACL constraint on table, Business Rule or data policies
404	Not Found	The requested resource is not found	ACL constraint or resource not available
405	Method not allowed	The HTTP action is not allowed or not supported by the API	
406	Not acceptable	The endpoint doesn't support the response format	Response format in the Request Accept Header
415	Unsupported media type	The endpoint does not support the format of the request body.	

RESTMessageV2

execute()	Sends the REST message to Endpoint
executeAsync()	Sends the REST message asynchronously, that means the instance doesn't wait for a response from provider
getEndpoint()	Get the URL of the endpoint for the REST message.
getRequestBody()	Get the content of the REST message body.
getRequestHeader(<headername>)*	Get the value for an HTTP header specified in the REST message.
getRequestHeaders()	Get HTTP headers that were set by the REST client and the associated values.

RESTMessageV2 (cont)	
saveResponseBodyAsAttachment(tblname,recordid,filename)**	Configures the REST message to save the returned response body as an attachment record.
saveResponseBodyAsAttachment(tblname,recordid,filename,encryptcontext)*	Configure the REST message to save the returned response body as an encrypted attachment record.
setAuthenticationProfile(type,profileid) **	Set the credentials for the REST message using an existing basic auth or OAuth 2.0 profile.
setBasicAuth(username,password)	Sets basic authentication headers for the REST message.
setEccCorrelator(correlator)	Associate outbound requests and the resulting response record in the ECC queue. This method only applies to REST messages sent through a MID Server.
setEccParameter(name,value)	Override a value from the database by writing to the REST message payload. This method only applies to REST messages sent through a MID Server.
setEndpoint(endpoint)	Set the endpoint for the REST message.
setHttpMethod(method)	The HTTP method this REST message performs, such as GET or PUT.
setHttpTimeout(milliseconds)	Set the amount of time the REST message waits for a response from the web service provider before the request times out.
setLogLevel(level)	Set the log level for this message and the corresponding response.Valid values for level are basic, elevated, and all.

RESTMessageV2 (cont)	
setMIDServer(midserver)	The name of the MID Server to use. Your instance must have an active MID Server with the specified name.
setMutualAuth(profilename)	Set the mutual authentication protocol profile for the REST message.
setQueryParameter(name, value)	Append a parameter to the end of the request URL with the form name=value.
setRequestBody(body)	Set the body content to send to the web service provider when using PUT or POST HTTP methods.
setRequestBodyFromAttachment(attachmentsysid)	Sets the request body using an existing attachment record.
setRequestHeader(name,value);	Set an HTTP header in the REST message to the specified value.
setRequestorProfile(requestorcontext,requestorid)	Override the default requestor profile for the REST message in order to retrieve an OAuth access token associated with a different requestor.
setStringParameter(name,value)	Set a REST message function variable with the specified name from the REST message record to the specified value.
setStringParameterNoEscape(name,value)	XML reserved characters in the value are converted to the equivalent escaped characters.
waitForResponse(seconds)	In seconds. Wait at most 60 seconds to get response from ECC Queue/Mid Server.

*By design, this method cannot return the value for a header set automatically by the system. To grant this method access to all headers, set the property glide.http.log_debug to true.

**the input parameters for this functions are string, and recordId is the sysid of the record

*encryptcontext should specify the sysid of the encryption context

By **Bibin Gokuldas**
(bibingokuldas)

cheatography.com/bibingokuldas/

Published 29th October, 2018.
Last updated 20th November, 2018.