BROADCOM®
MAINFRAME SOFTWARE

Getting Started with Consuming REST APIs in Gen's CICS and Java applications

July 27, 2022

Clay Rabun
Gen Product Engineer

Disclaimer

Certain information in this presentation may outline Broadcom's general product direction. This presentation shall not serve to (i) affect the rights and/or obligations of Broadcom or its licensees under any existing or future license agreement or services agreement relating to any Broadcom software product; or (ii) amend any product documentation or specifications for any Broadcom software product. This presentation is based on current information and resource allocations as of ***July* 2022** and **is subject to change or withdrawal by Broadcom at any time without notice**. **The development, release and timing of any features or functionality described in this presentation remain at Broadcom's sole discretion**.

Notwithstanding anything in this presentation to the contrary, upon the general availability of any future Broadcom product release referenced in this presentation, Broadcom may make such release available to new licensees in the form of a regularly scheduled major product release. Such release may be made available to licensees of the product who are active subscribers to Broadcom maintenance and support, on a when and if-available basis. The information in this presentation is not deemed to be incorporated into any contract.

THIS PRESENTATION IS FOR YOUR INFORMATIONAL PURPOSES ONLY. Broadcom assumes no responsibility for the accuracy or completeness of the information. TO THE EXTENT PERMITTED BY APPLICABLE LAW, BROADCOM PROVIDES THIS DOCUMENT "AS IS" WITHOUT WARRANTY OF ANY KIND, INCLUDING, WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT. In no event will Broadcom be liable for any loss or damage, direct or indirect, in connection with this presentation, including, without limitation, lost profits, lost investment, business interruption, goodwill, or lost data, even if Broadcom is expressly advised in advance of the possibility of such damages.

What does Consuming REST APIs mean?

Consumption vs Publishing

Consumption means that the Gen-generated application can invoke a service endpoint

- Gen now provides this functionality with the new “Call REST” statement
  - HTTP call
  - JSON content
  - May be described by an OpenAPI Specification
- Similar to the functionality provided by the “Call External” statement
  - SOAP call
  - XML content
  - Described by a WSDL

Publishing means that the Gen-generated application can make itself available as a service endpoint

- Response Systems provides this functionality with their Web Services GENius product

How do I get started?

1. Apply PTFs
2. Update Encyclopedia Schema Tables
3. Convert Existing Models
4. Add Call REST statement
5. Configure Execution Environment

Apply PTFs

Apply PTFs

Workstation

- TSN86221 / LU03174 – Toolset support for Call REST statement
- TSN86225 / LU05124 – Toolset support for Call REST statement security schemes
- GEN86212 / LU03172 – COBOL and Java generator support for Call REST statement
- BTN86209 / LU03169 – Build Tool support for Call REST statement for Java applications
- RTJ86208 / LU03173 – Java Runtime support for Call REST statement
- RTJ86210 / LU05125 – Java Runtime support for Call REST statement security schemes

OR

- WKS86300 / LU06327 – Gen 8.6.3 Consolidation PTF

Apply PTFs

Client Server Encyclopedia (CSE)

- CSI86210 / LU04187 – HP/UX CSE support for Call REST statement
- CSN86212 / LU03171 – Windows CSE support for Call REST statement
- CSR86210 / LU04188 – AIX CSE support for Call REST statement

OR

- CSI86300 / LU06350 – Gen 8.6.3 Consolidation PTF (HP/UX CSE)
- WKS86300 / LU06327 – Gen 8.6.3 Consolidation PTF (Windows CSE)
- CSR86300 / LU06338 – Gen 8.6.3 Consolidation PTF (AIX CSE)

Apply PTFs

Host Encyclopedia (HE) and z/OS Runtimes

- LU03643 / FMID CEHB860 – Prerequisite PTF to update your SMP/E CSI
  - After APPLYing this PTF, you must customize and execute the TIDDDEF1 CEHBSAMP member before APPLYing the remaining PTFs
- LU03408 / FMID CEG6860 – Host Encyclopedia support for Call REST statement
  - After APPLYing this PTF, you must rebind the necessary DB2 Packages. Use the PTFBIND job with TIB03408 as the input clist.
  - The Sample and Help models have been re-converted so the Call REST statement can be added to these models. Use CEJOB12 and CEJOB13 to reload these models to your Host Encyclopedia, if desired.
- LU03409 / FMID CEHB860 – Common Modules support for Call REST statement
- LU03410 / FMID CEHC860 – CICS Runtime Modules support for Call REST statement
- LU05281 / FMID CEHC860 – CICS Runtime Modules support for Call REST statement security schemes

Update Encyclopedia Schema Tables

BROADCOM
MAINFRAME SOFTWARE

Update Encyclopedia Schema Tables

New system objects are required to support the Call REST statement. Update your encyclopedia's Schema tables to pick up these new objects.

- On the CSE, run cse_config and select the "Update Encyclopedia tables to this release" configuration option to reload the Schema tables
- On the HE, use CEJOB05 (or CAJOB05) to reload the Schema tables

Convert Existing Models

BROADCOM
MAINFRAME SOFTWARE

Convert Existing Models

To add the Call REST statement to an existing model, the model must first be converted (or re-converted) to the 9.2.A6 schema level after the encyclopedia's Schema tables have been updated. If you are adding the Call REST statement to a new model, conversion is not necessary.

- On the CSE, use the Encyclopedia Client or the convmodl server command to perform model conversion
- On the HE, use the 1.3.11 option from the HE Main Menu to access the model conversion utilities

Add Call REST statement

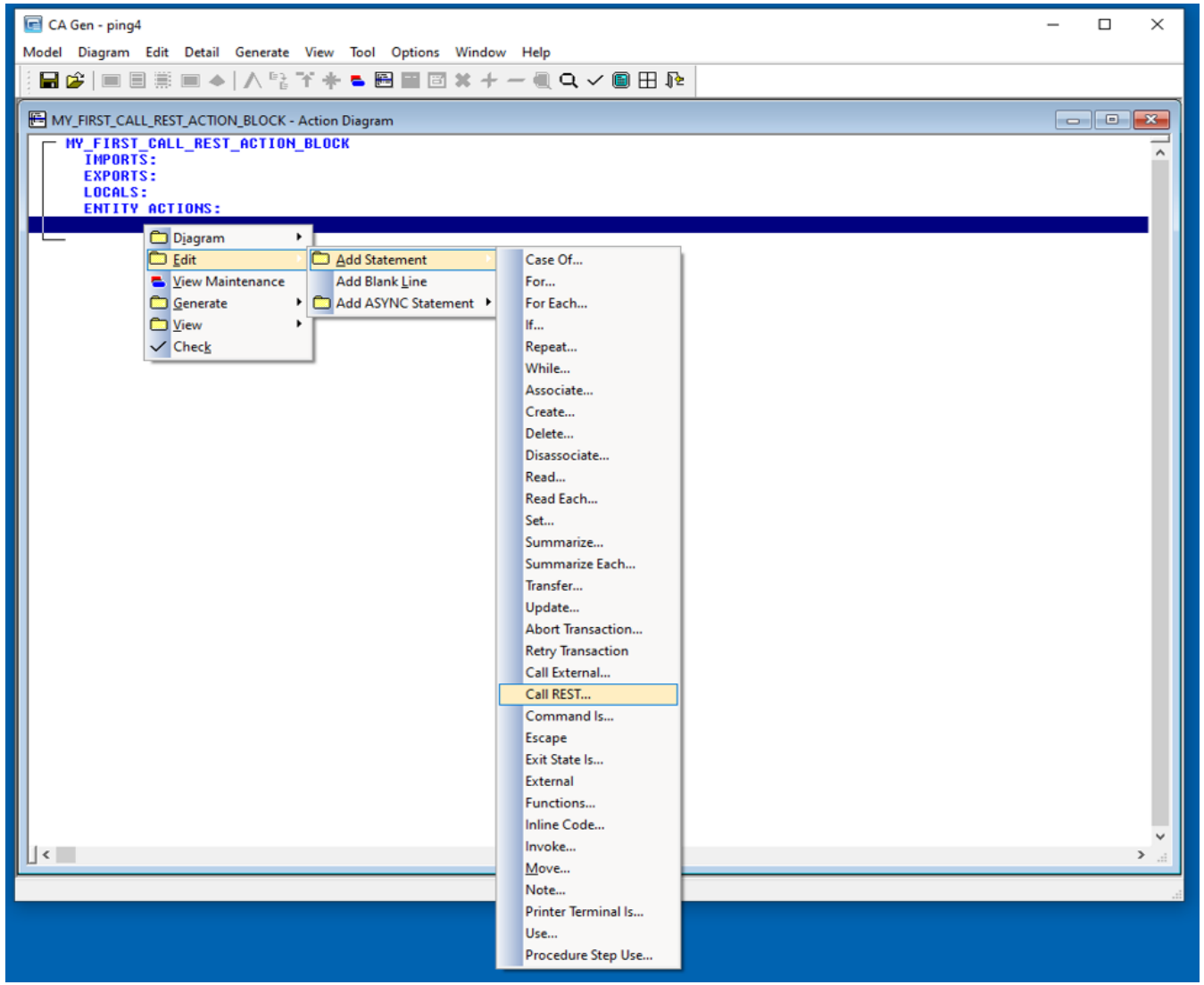
BROADCOM
MAINFRAME SOFTWARE

Add Call REST statement

The Call REST statement is how your Gen application can use the functionality provided by a RESTful Web Service. The following restrictions apply to the Call REST statement:

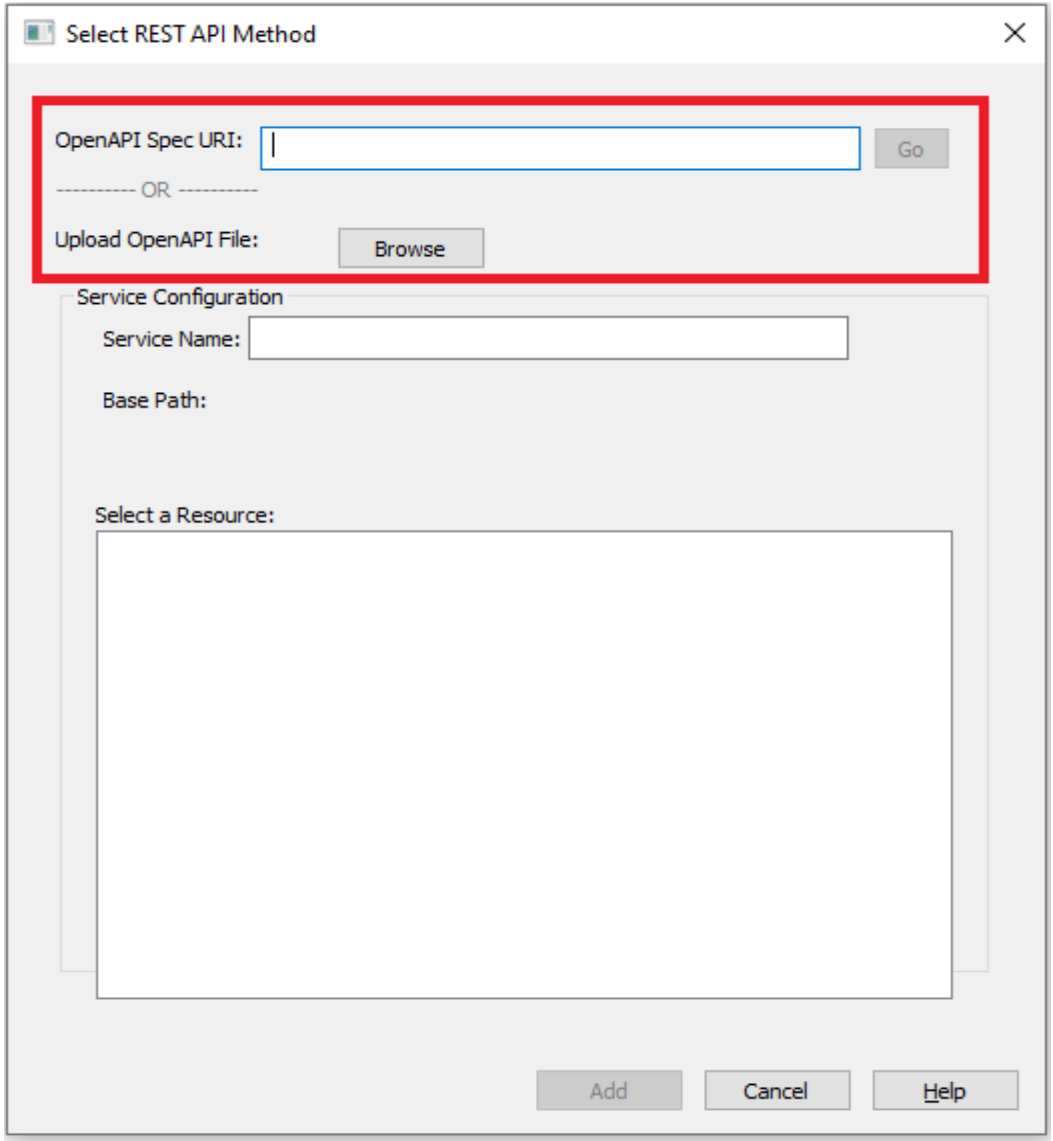
- Call REST statements are not allowed in Procedure Steps
- Each Call REST statement must be added to a separate Action Diagram
- Only the Call REST statement and Note statements can exist in the Action Diagram
- Only Import and Export Views are used in the Action Diagram. Local and Entity Action Views are not used
- Action Diagrams containing a Call REST statement cannot have their “Dynamic Link (z/OS)” property set to Compatibility

Add Call REST statement



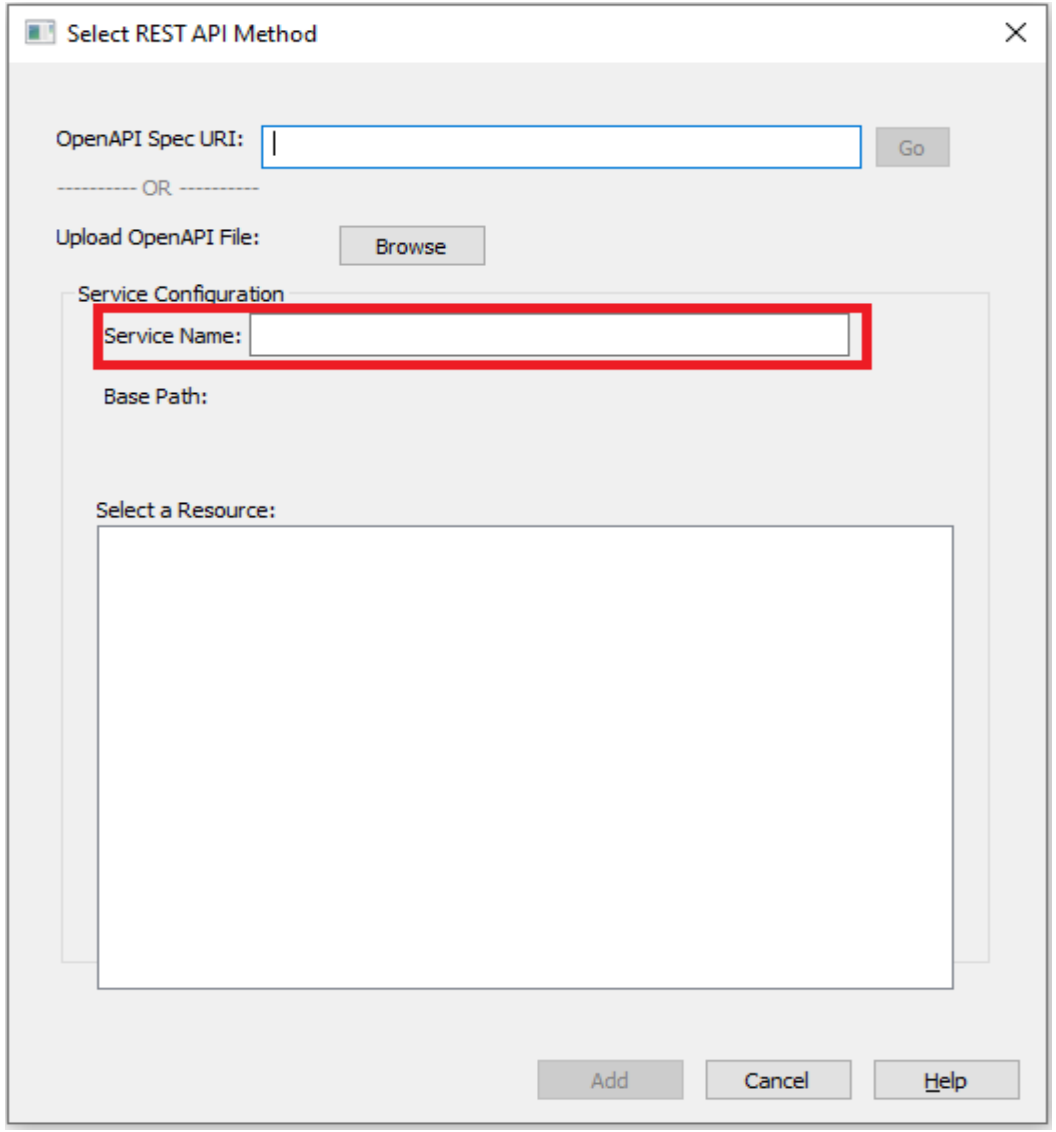
- After creating a new Action Diagram, add a Call REST statement by selecting the Edit / Add Statement / Call REST… menu item

Add Call REST statement



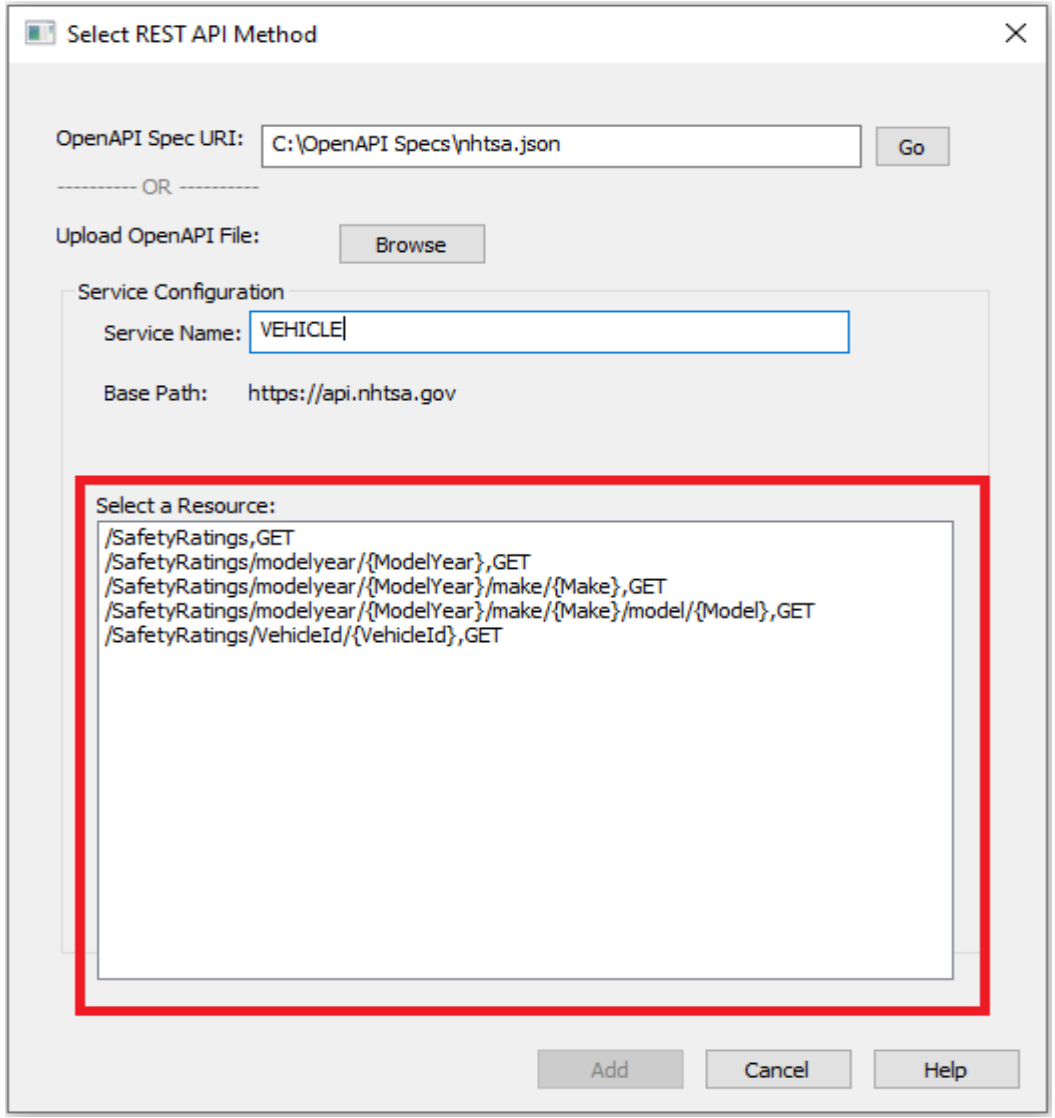
- An OpenAPI specification is a format for describing REST APIs. The OpenAPI specification allows an API developer to define:
  - Available endpoints and the operations for each endpoint
  - The input and output parameters for the endpoint and operation
  - Authentication for the service
  - Additional information about the API
- The Gen Toolset can process either JSON or YAML specifications

Add Call REST statement



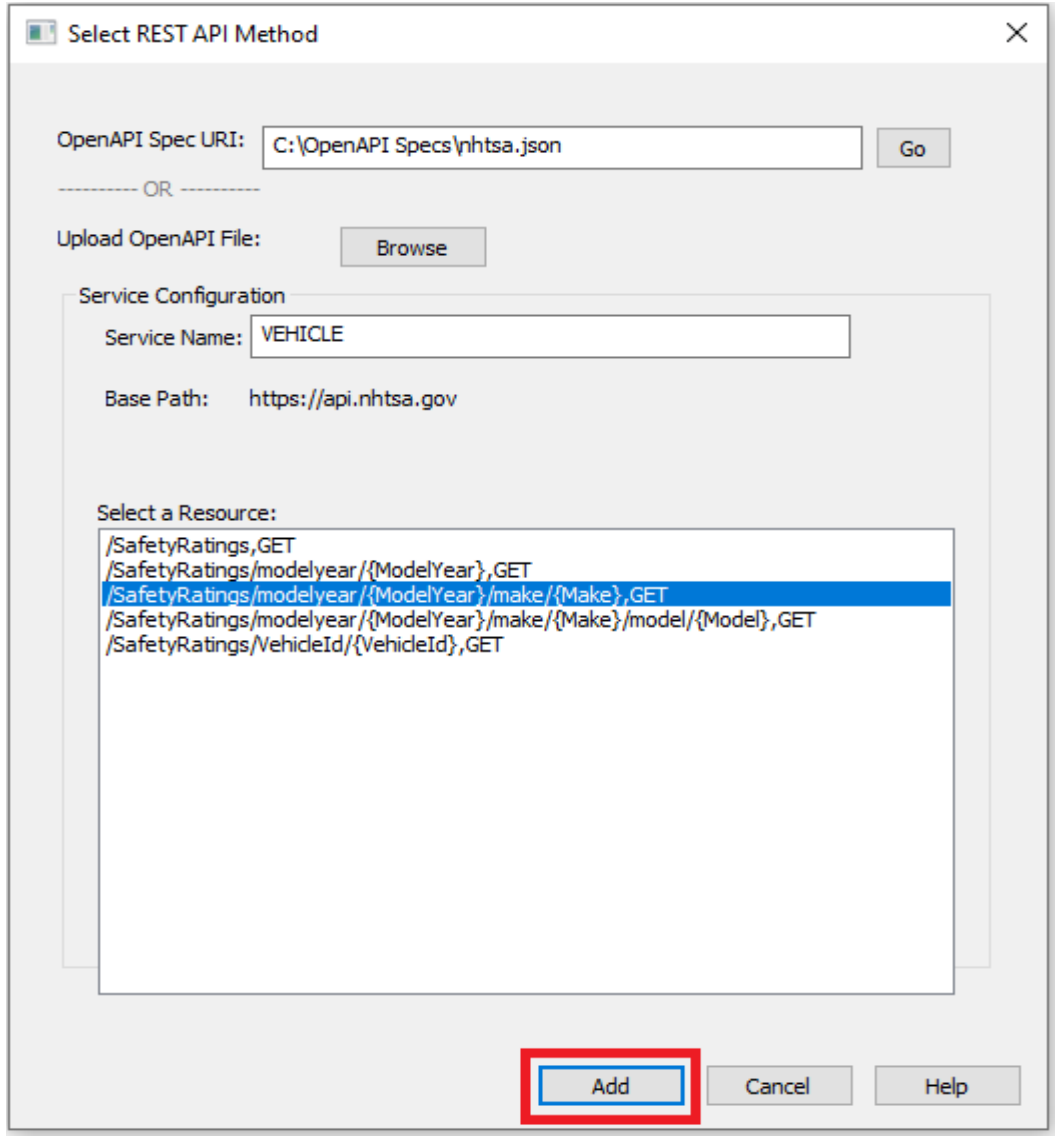
- The Service Name is a logical name or alias representing the Host and Port of the machine hosting the RESTful web service
  - Can be up to 32 characters
  - For Java, used as a key into the callrest.properties file
  - For CICS, first 8 characters must match name of URIMAP defined in CICS region

Add Call REST statement



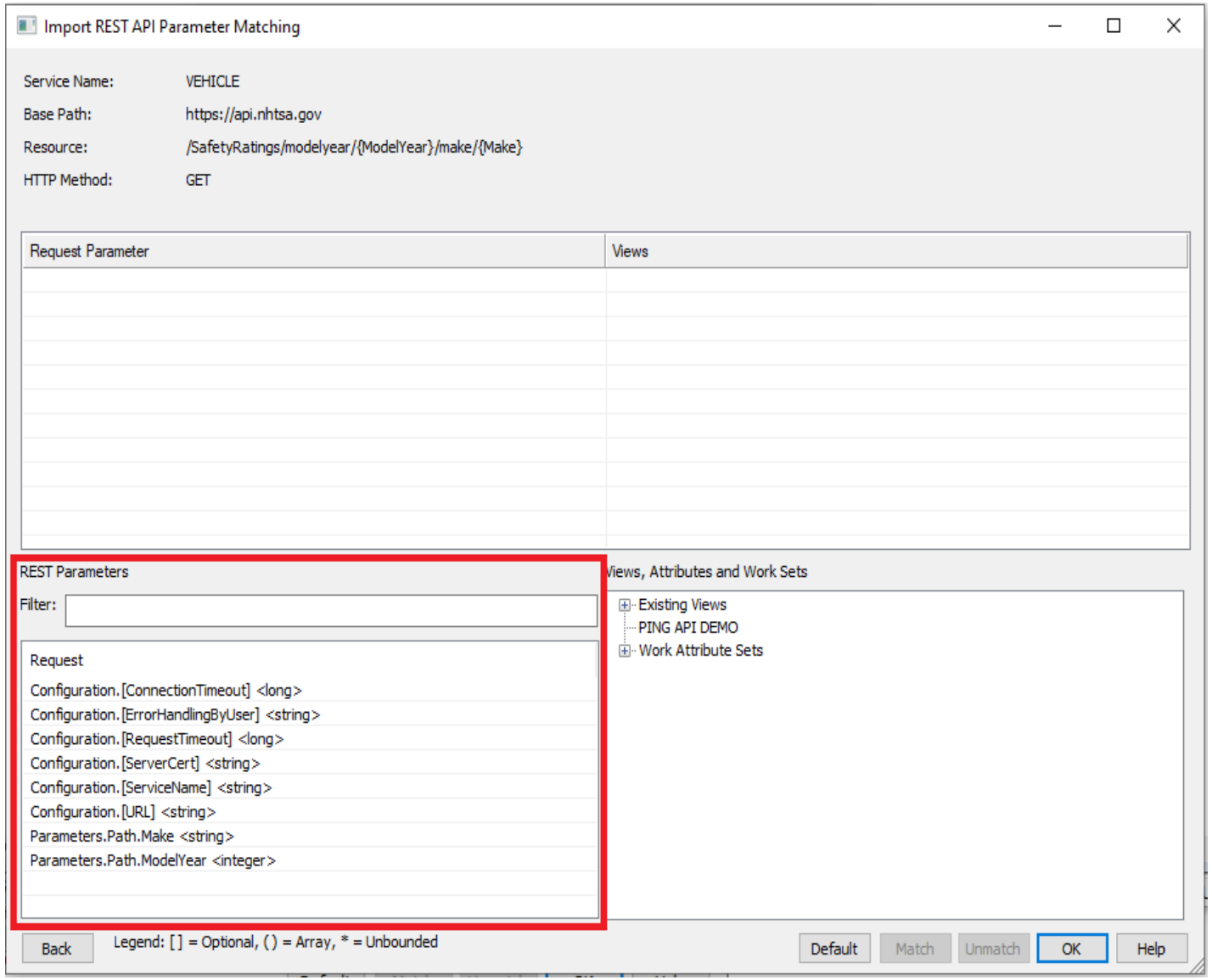
- The Select a Resource listbox contains a list of available service endpoints and operations
  - GET
  - POST
  - PUT
  - DELETE

Add Call REST statement



- The Add pushbutton will be enabled once an OpenAPI specification has been processed, a Service Name has been entered, and a Resource has been selected

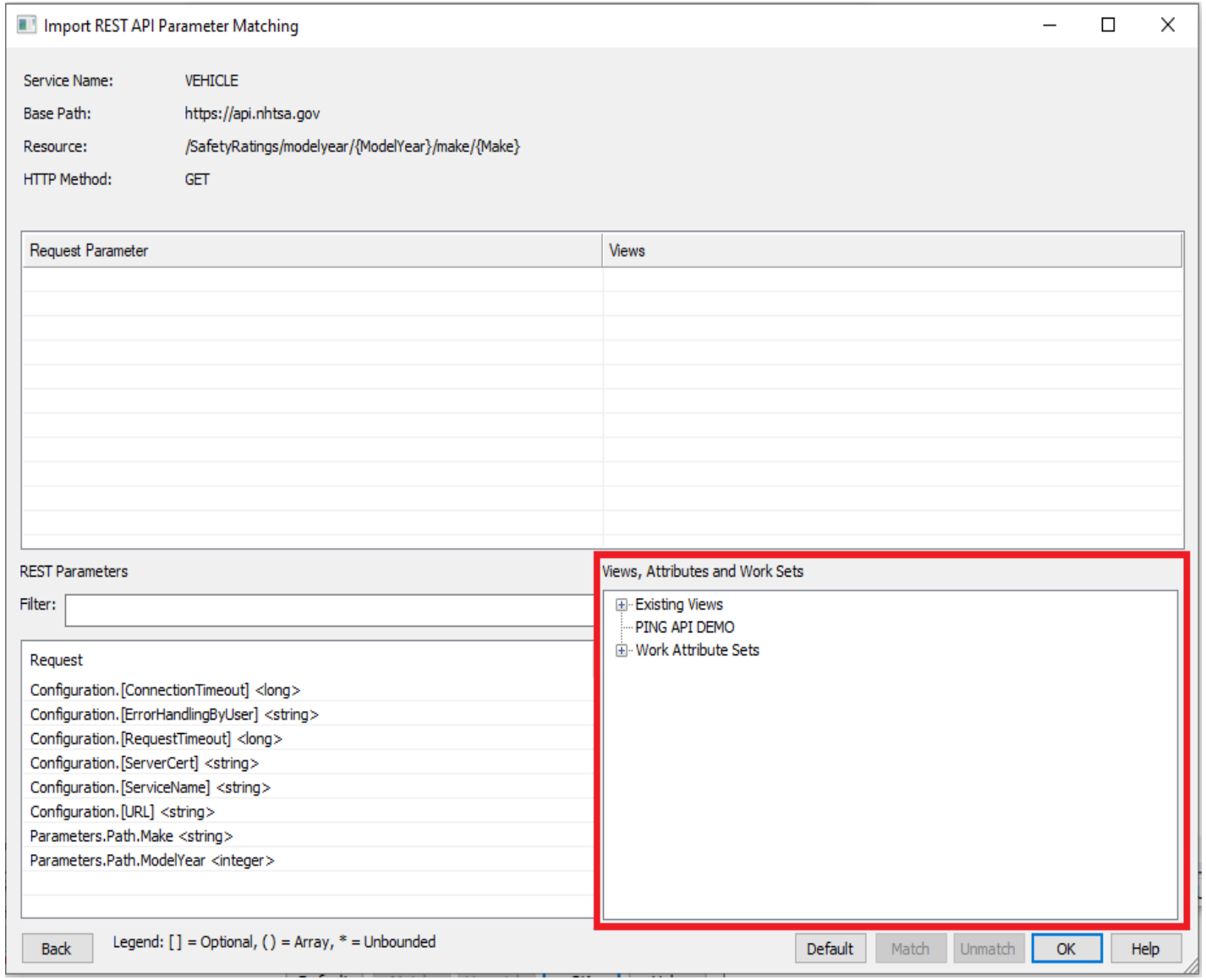
Add Call REST statement



Import REST Parameters

- Retrieved from OpenAPI Spec
- Shows allowable inputs to RESTful service
  - Request.Body
  - Request.Parameters.Cookie
  - Request.Parameters.Header
  - Request.Parameters.Path
  - Request.Parameters.Query
- Shows JSON datatypes
- Request.Configuration and Request.Authentication.OAuth2.Configuration
  - Added by Gen Toolset – Not from OpenAPI Spec
  - Allows PAD logic to provide override values

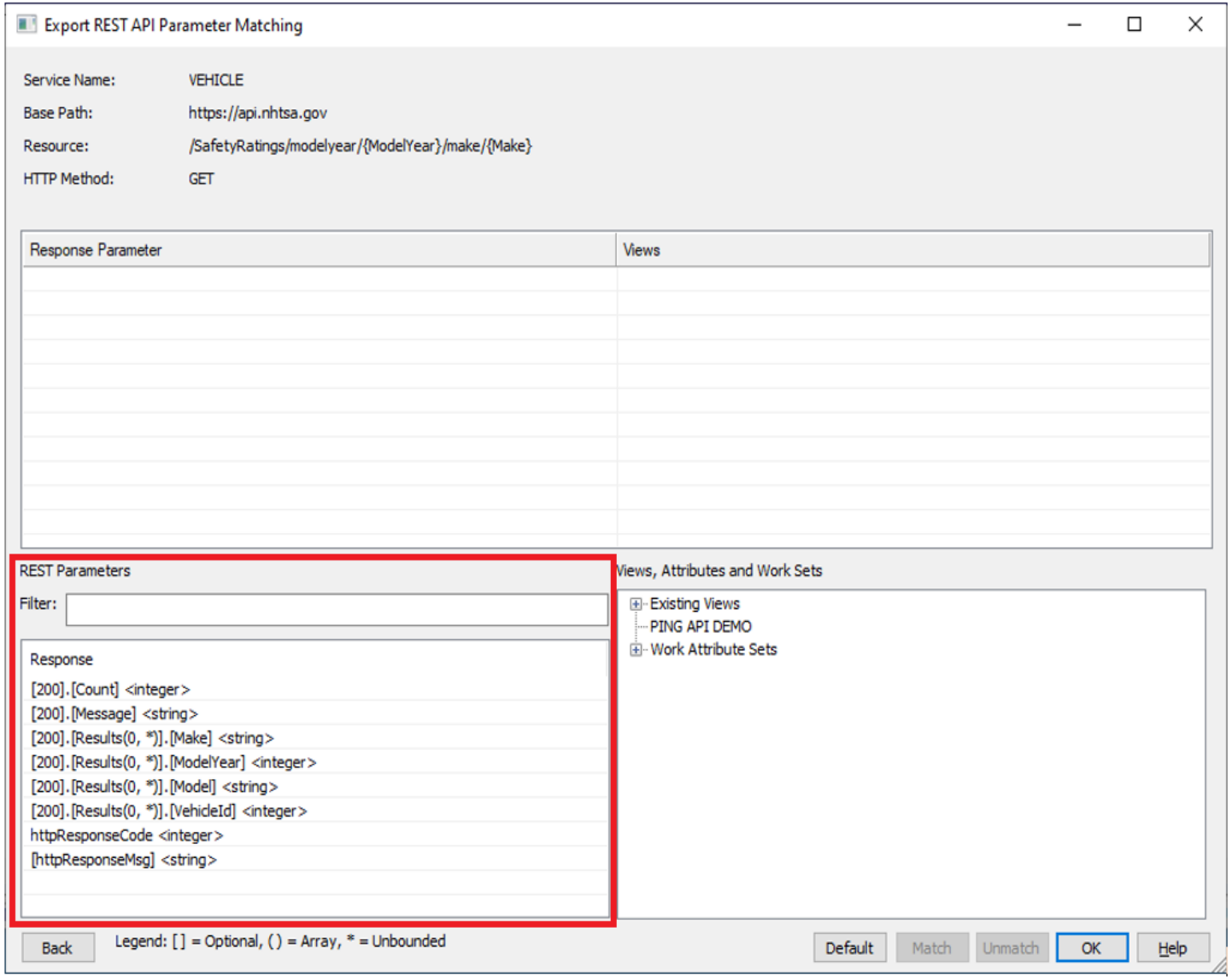
Add Call REST statement



Import Views, Attributes and Work Sets

- Import Views to be matched to Import REST Parameters
- Default pushbutton will create views using these Gen domain types and sizes
- Can manually create views using these allowable Gen domain types
- Gen does not currently support the BLOB domain type for views on the z/OS platform

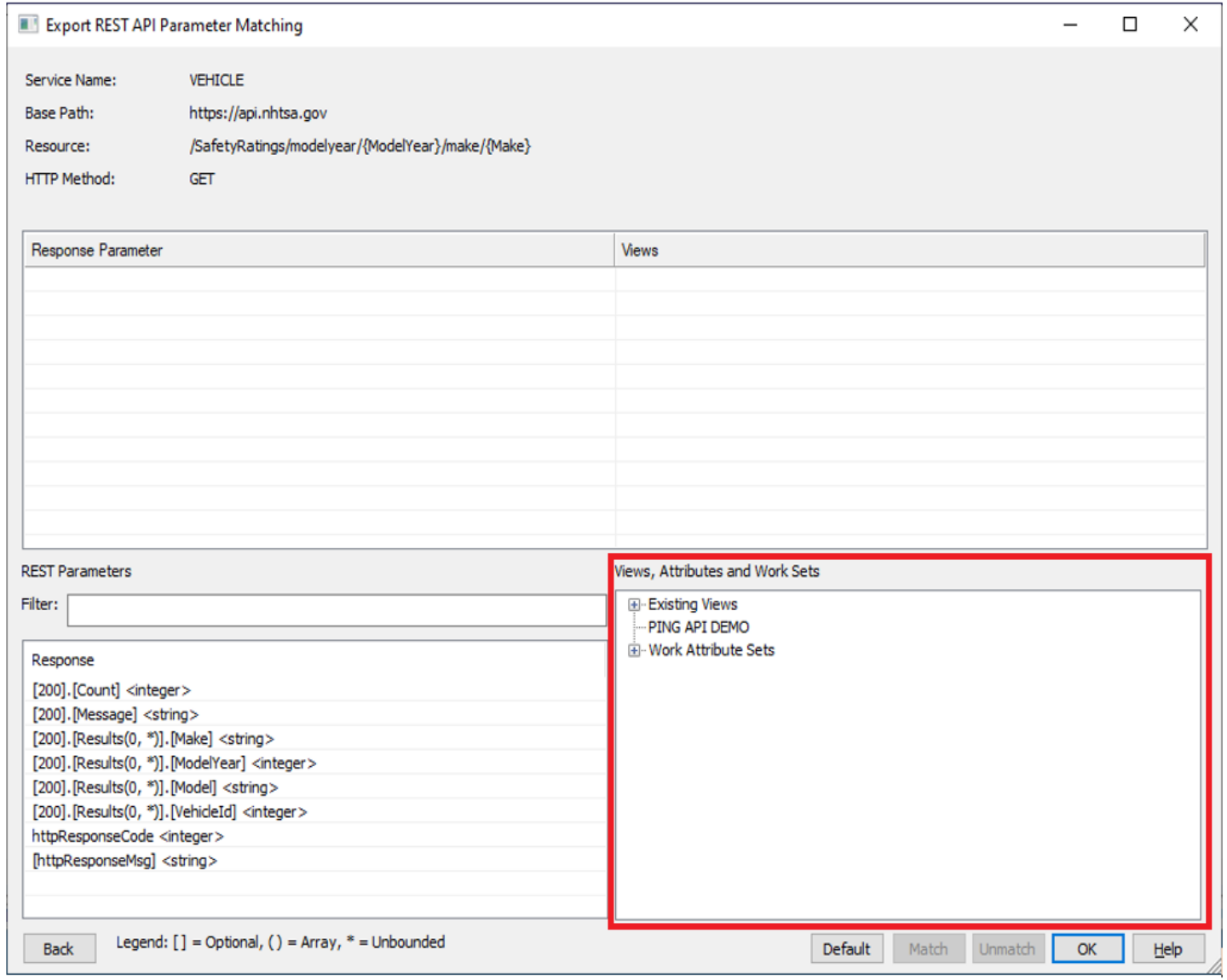
Add Call REST statement



Export REST Parameters

- Retrieved from OpenAPI Spec
- Shows possible outputs from RESTful service
  - Response.200
  - Response.400
  - Response.nnn
- Shows JSON datatypes
- Response.httpResponseCode, Response.httpResponseMsg, Response.authResponseCode and Response.authResponseMsg
  - Added by Gen Toolset – Not from OpenAPI Spec
  - Used for error handling

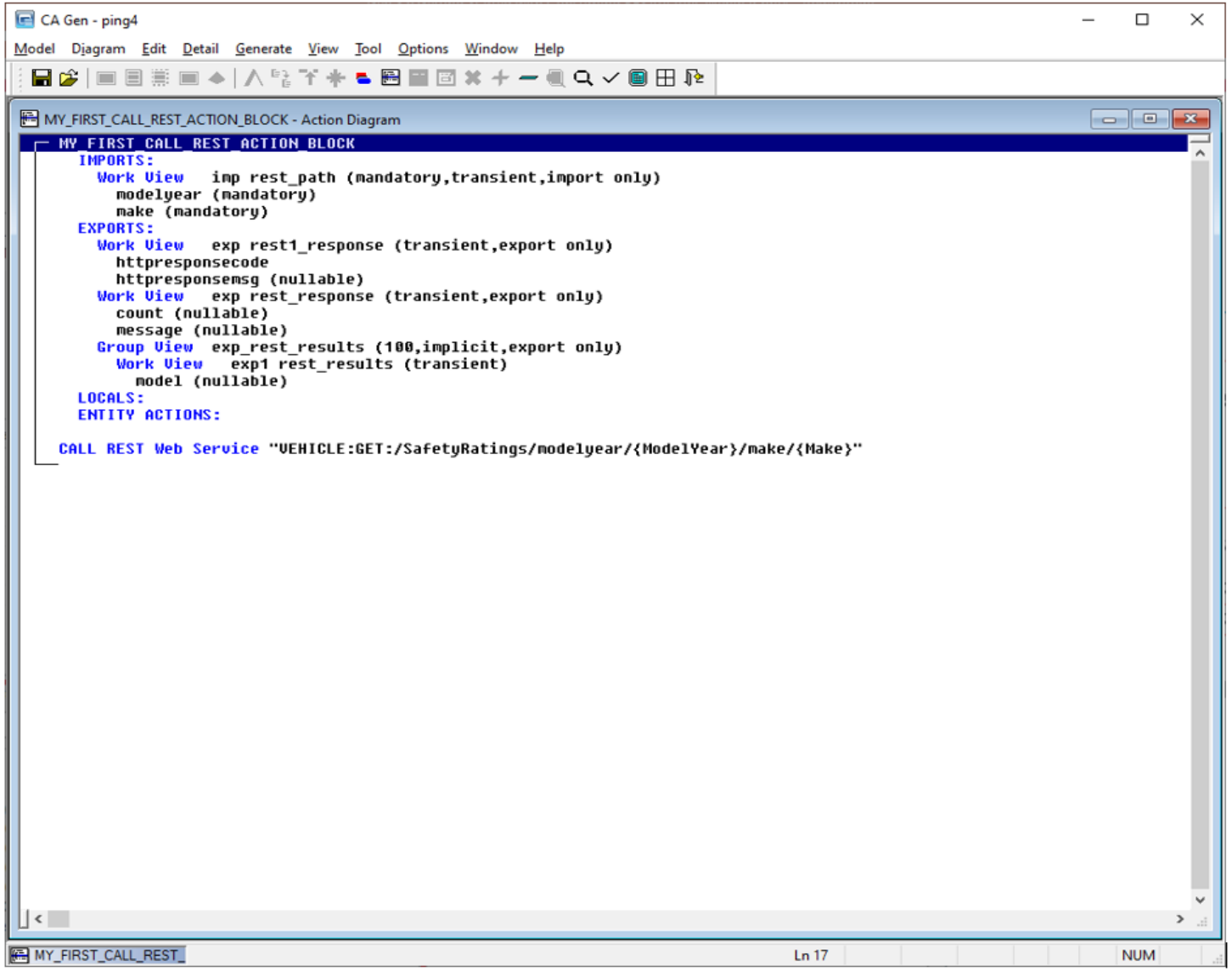
Add Call REST statement



[Export Views, Attributes and Work Sets]

- Export Views to be matched to Export REST Parameters
- Default pushbutton will create views using these [Gen domain types and sizes]
- Can manually create views using these [allowable Gen domain types]
- When the Default pushbutton is used, the views created may not be optimal for the service's data requirements
  - View attribute sizes may be larger than necessary
  - Numeric View attribute decimal places may not match service's data
  - Repeating group view cardinality may not match service's data
  - Multiple repeating group views created when the same view could be reused
  - View attributes are not marked as being Nullable

Add Call REST statement



- After CALL REST statement is added, the Toolset shows the Service Name, HTTP Method and the service endpoint path
- To mark views as being Nullable
  - Go to View Maintenance
  - Select view attribute(s)
  - Right-click and select Edit
  - Select Nullable

Call REST statement error handling

- The Gen runtimes normally populate the Response.httpResponseCode and Response.httpResponseMsg parameters after returning back from the service as a means to provide the opportunity for error handling
- If these parameters are matched to Gen views in the Call REST Action Diagram, then PAD logic could be written to take different actions depending upon the httpResponseCode value
  - The Response.httpResponseCode and Response.httpResponseMsg parameters could be matched to Export views that are returned from the Call REST Action Diagram
  - The calling Action Diagram could then query those views to determine what action to take
- The Inline Code approach used by the Call External statement is also available for cases where the Response.httpResponseCode and Response.httpResponseMsg are not able to be populated

Configure Execution Environment

Configure Execution Environment

- The Base URL from the OpenAPI Spec is likely not the correct URL
  - May access one machine in Development and another in QA and another in Production
  - Wouldn’t want to have to update the OpenAPI Spec and recode the Call REST statement for each environment
  - Need a way to change the configuration without making code changes
- For Java applications, the Call REST statement can be configured at deployment time using the callrest.properties file. It can also be configured at runtime using the Configuration parameters available as import parameters to each Call REST statement. These parameters can be matched to Gen views so that their values can be set dynamically. Additionally, the Base URL can be set using the modifyURL method in the WebServiceMethodCallExit user exit.
- For example, the Base URL could be changed in the callrest.properties file with a line like the following:
  - VEHICLE=https://api.nhtsa.gov:1999/v2
- As another example, the Connection Timeout could be set for all services with a line like the following:
  - CONNECTION_TIMEOUT=60
- And the Connection Timeout could be set for just the VEHICLE service with a line like the following:
  - VEHICLE.CONNECTION_TIMEOUT=120

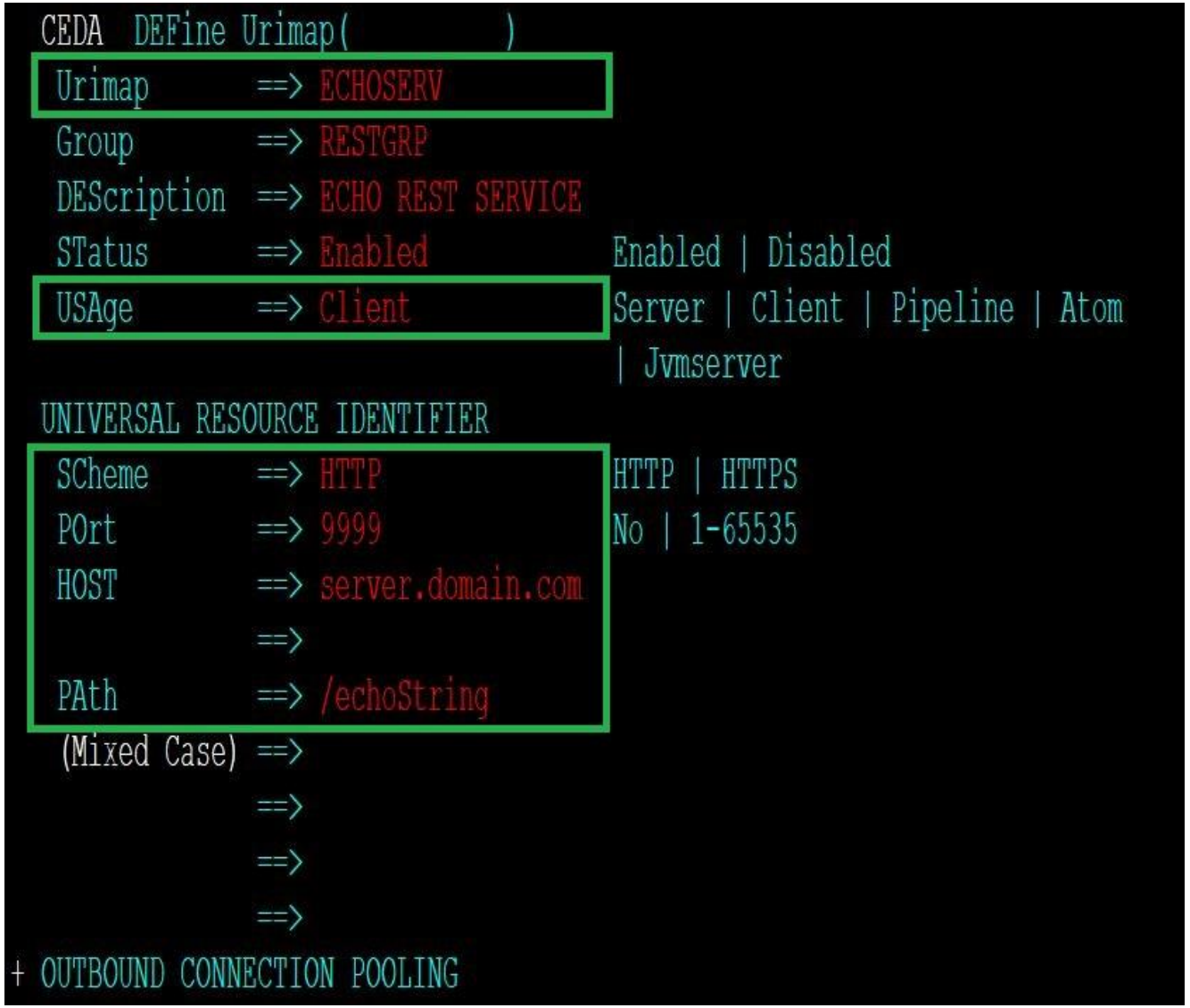
Configure Execution Environment

- Gen provides a number of methods to configure the Call REST statement for Java applications. The precedence order (lowest to highest) is as follows:
  - **Default** – If none of the configuration parameters have been defined, the code will use the URL from the OpenAPI specification and a set of default values
  - **Global** – The callrest.properties file can define parameters that apply to all REST calls. These apply to any parameter except for the URL and override the default values
  - **Service** – If a service is defined in the callrest.properties file, then parameters for that service override the global value and the defaults
  - **Runtime** – You can match configuration parameters to Gen attribute views. If these values are matched, the runtime will use those values for the REST calls, overriding any prior parameter definitions
  - **User Exit** – The URL for the RESTful Web Service can be set using the modifyURL method in the WebServiceMethodCallExit user exit. This method will override any prior URL definition

Configure Execution Environment

- For CICS applications, the Call REST statement can be configured by modifying the URIMAP resource. It can also be configured at runtime using a subset of the Configuration parameters available as import parameters to each Call REST statement. These parameters can be matched to Gen views so that their values can be set dynamically
- For example, the URIMAP resource can be updated to point to the correct Host and Port of the machine hosting the RESTful service
  - This allows the configuration to change without having to modify the application
- As another example, the Request.Configuration.ServiceName parameter could be used to modify the URIMAP name used at runtime
  - This could allow separate URIMAP resources to be defined for Development, QA and Production environments
  - This prevents you from having to modify the URIMAP resource in CICS

Configure Execution Environment



- URIMAP is the name of the URIMAP resource
  - Should be used in the Service Name field when adding a Call REST statement in the Toolset
- USAGE must be set to Client
  - The Gen application is acting as an HTTP client
- SCHEME can be HTTP or HTTPS
- HOST and PORT defines the machine that is hosting the REST service
- PATH doesn't really matter to Gen
  - Gen only uses URIMAP to make the connection to the machine that is hosting the REST service
  - Gen uses other Base URL and Path info from selections made in the Toolset

Configure Execution Environment

- SOCKETCLOSE should be set to a non-zero value in order to take advantage of connection pooling

Configure Execution Environment

- Additional configuration steps for CICS applications:
  - If you have customized either MTQDEFLT or TIRMTQB2 CEHBSAMP members, then you will need to rebuild the TIRMTQBZ user exit DLL after merging your customizations into the new members
    - Use the MK5EXITS job to rebuild the TIRMTQBZ user exit DLL
  - Modify TIRXINFO CEHBSAMP member to set the codepage for your CICS applications and rebuild the TIRXINFZ user exit DLL
    - Use the MKCRUN job to rebuild the TIRXINFZ user exit DLL
    - Cooperative packaged applications already require this step
    - Online packaged applications now require this step too if using Call REST statement
  - Add new CICS Program Definitions for TIRABRT and TIRABRTC to your CICS region(s)
    - The NDEFCICS CEHBSAMP member can be used to add these new CICS resources
  - Copy the TIRABRT, TIRABRTC, TIRMTQBZ, TIRORUNC, and TIRXINFZ DLLs to a dataset in your CICS region's DFHRPL concatenation
    - Perform a NEWCOPY on these programs
  - Add new CICS URIMAP Definitions to your CICS region(s)
    - You will need a unique URIMAP resource for each RESTful Web Service server (not each Web Service endpoint) that is being accessed

Where can I find more information?

- Tech Docs – Add a Call REST statement
- Toolset Messages
- Runtime Messages
  - Most REST runtime messages are in the TIRM3xxE range
- Press Help pushbutton on Toolset dialogs
- Knowledge Base articles
  - Gen 8.6 Consuming REST APIs: Getting Started
  - Gen 8.6 Consuming REST APIs: FAQ
  - Gen 8.6 Consuming REST APIs: Security
  - Gen 8.6 Consuming REST APIs: Authentication

BROADCOM®
MAINFRAME SOFTWARE

Thank you!