

HTTP_Resource_Services

Application service module that provides simple [CRUD](#) operations for resources that map directly to database tables.

Syntax

```
Response = HTTP_Resource_Services(@Service, @Params)
```

Returns

The meaning of the response value depends on the service.

Parameters

Parameter	Description
@Service	The name of the service being requested. Required.
@Params	Generic parameters. Refer to a specific service to determine the actual parameters used.

Remarks

[HTTP_Resource_Services](#) is a helpful application service module that web service routines can use to perform common database operations. This works well for web services that are tightly associated with a single database table. For example, a URL with [/customers](#) is likely to be directly linked to a CUSTOMERS database table.

Both the included [HTTP_Users_Services](#) and [HTTP_Contacts_Services](#) sample web service routines make use of [HTTP_Resource_Services](#).

Services

Service	Description
GetObject	Usage: HTTP_Resource_Services('GetObject', TableName, KeyID, ColumnNames, PropertyNames, DataTypes, MVGroupNames, NestedGroupNames, BaseSelfRelURL, KeyParts, KeyDelimiter, AllowKeyProperty) Comments: Creates one or more new resource objects. The database table, Key ID, and list of column names are provided, these will be used to create properties within the resource object. Returns: A handle to the JSON object.
GetObject ts	Usage: HTTP_Resource_Services('GetObjects', TableName, Filter, ColumnNames, PropertyNames, DataTypes, MVGroupNames, NestedGroupNames, BaseSelfRelURL, KeyParts, KeyDelimiter, AllowKeyProperty) Comments: Creates one or more new resource objects. The database table, Key ID, and list of column names are provided, these will be used to create properties within the resource object. Returns: An @FM delimited list of handles to the JSON objects.
ParseRe source	Usage: HTTP_Resource_Services('ParseResource', SerializedResource) Comments: Parses the indicated serialized resource. Returns: A handle to the JSON object.

AddProperty	Usage: HTTP_Resource_Services('AddProperty', ObjectHandle, PropertyName, PropertyValue, DataType) Comments: Adds a property to the indicated resource object along with any value provided. Returns: A handle to the JSON object.
AddProperties	Usage: HTTP_Resource_Services('AddProperties', ObjectHandle, PropertyNames, PropertyValues, DataTypes) Comments: Adds one or more properties in the indicated resource object. This is a wrapper for the <i>AddProperty</i> service to make it easier to add multiple properties to a resource object in one call. Returns: A handle to the JSON object.
AddSubProperty	Usage: HTTP_Resource_Services('AddSubProperty', ObjectHandle, PropertyName, SubPropertyName, SubPropertyValue, DataType) Comments: Adds a sub-property to the indicated property of the indicated resource object along with any value provided. This is a one-service alternative to creating a temporary resource object and property (via <i>GetObject</i> with <i>AddProperty</i>) and then adding this object to the primary resource as a property (via <i>AddProperty</i> using 'ObjectHandle' data type). Returns: A handle to the JSON object.
AddSubProperties	Usage: HTTP_Resource_Services('AddSubProperties', ObjectHandle, PropertyName, SubPropertyNames, SubPropertyValues, DataTypes) Comments: Adds one or more sub-properties to the indicated property of the indicated resource object. This is a wrapper for the <i>AddSubProperty</i> service to make it easier to add multiple sub-properties to a property in one call. Returns: A handle to the JSON object.
AddSubResourceObject	Usage: HTTP_Resource_Services('AddSubResourceObject', ObjectHandle, PropertyName, SubResourceObjectHandle) Comments: Adds a sub-resource object handle to the indicated property of the indicated resource object. Any pre-existing sub-resources will be preserved and new ones will be appended. Returns: A handle to the JSON object.
AddSubResourceObjects	Usage: HTTP_Resource_Services('AddSubResourceObjects', ObjectHandle, PropertyName, SubResourceObjectHandles) Comments: Adds one or more sub-resource object handles to the indicated property of the indicated resource object. This is a wrapper for the <i>AddSubResourceObject</i> service to make it easier to add multiple sub-resource object handles to a property in one call. Returns: A handle to the JSON object.
AddSubResource	Usage: HTTP_Resource_Services('AddSubResource', ObjectHandle, PropertyName, SubResourcePropertyNames, SubResourcePropertyValues, DataTypes) Comments: Adds a sub-resource to the indicated property of the indicated resource object. A sub-resource object handle is created using the sub-resource property names and values and then added. Any pre-existing sub-resources will be preserved and new ones will be appended. Returns: A handle to the JSON object.

AddSubResources	Usage: HTTP_Resource_Services('AddSubResources', ObjectHandle, PropertyName, SubResourcePropertyNames, SubResourcePropertyValuesArray, DataTypes) Comments: Adds one or more sub-resources to the indicated property of the indicated resource object. This is a wrapper for the <i>AddSubResource</i> service to make it easier to add multiple sub-resources to a property in one call. Returns: A handle to the JSON object.
AddLinkRelation	Usage: HTTP_Resource_Services('AddLinkRelation', ObjectHandle, Relation, URL, Title, IsTemplate, IgnoreResourceID, IgnoreProperty, IgnoreQueryParam) Comments: Adds a <i>_links</i> relation to the indicated resource object. This is part of the HAL specification. Returns: N/A
AddLinkRelations	Usage: HTTP_Resource_Services('AddLinkRelations', ObjectHandle, Relations, URLs, Titles, IsTemplates, IgnoreResourceIDs, IgnoreProperties, IgnoreQueryParams) Comments: Adds one or more <i>_links</i> relations to the indicated resource object. This is part of the HAL specification. Returns: N/A
AddEmbeddedResources	Usage: HTTP_Resource_Services('AddEmbeddedResources', ObjectHandle, EmbeddedResourceName, EmbeddedResourceObjectHandles, Singular) Comments: Adds one or more embedded resources in the indicated resource object. Embedded resources are stand-alone resources on their own (and usually have their own endpoint), but are included, at least partially, in another resource due to the close relationship between them. This is often done to eliminate multiple requests to pull in all significant resource data. This is part of the HAL specification. Returns: N/A
AddFormAction	Usage: HTTP_Resource_Services('AddFormAction', ObjectHandle, Name, Method, URL, Title, Fields, FieldProperties) Comments: Adds one <i>_forms</i> action in the indicated resource object. Due to the complexity and depth of a single action, this service will only add one action at a time. Additional actions can be added by calling this service multiple times. Returns: N/A
GetSerializedResource	Usage: HTTP_Resource_Services('GetSerializedResource', ObjectHandle, Style) Comments: Serializes the indicated resource object and then releases the object. Returns: The serialized JSON resource.
LoremIpsum	Usage: HTTP_Resource_Services('LoremIpsum') Comments: Creates a dummy HTTP response using <i>lorem ipsum</i> content. This is a placeholder service used when new endpoint handlers are automatically created. It is expected to be replaced by the actual API logic. Returns: N/A. Nothing is returned per se, but the HTTP response body will already be updated with the <i>lorum ipsum</i> content and the HTTP response status will be set to 200.

GetURLTemplate	Usage: HTTP_Resource_Services('GetURLTemplate', URL, IgnoreResourceID, IgnoreProperty, IgnoreQueryParam) Comments: Returns the indicated URL as a URL Template. Returns: A templated URL path.
GetDatabaseItem	Usage: HTTP_Resource_Services('GetDatabaseItem', TableName, SelfURL, KeyID, ColumnNames, ItemArrayLabel) Comments: Gets the resource item from a database table. The format will be HAL+JSON. Returns: A database row in HAL+JSON format.
GetDatabaseItems	Usage: HTTP_Resource_Services('GetDatabaseItems', Filter, TableName, SelfURL, ColumnNames, ItemArrayLabel) Comments: Gets the resource items from a database table. The format will be HAL+JSON. Returns: Database rows in HAL+JSON format.
DeleteDatabaseItem	Usage: HTTP_Resource_Services('DeleteDatabaseItem', TableName, KeyID) Comments: Deletes the resource item from a database table. Returns: N/A
PutDatabaseItem	Usage: HTTP_Resource_Services('PutDatabaseItem', TableName, SelfURL, KeyID, AllowedColumnNames) Comments: Creates a new or updates a specific resource item in the database table. To conform to the requirements of the PUT method, the entire resource will be replaced with the content being passed in, even if only a few properties (e.g., database columns) are being updated. Thus, it is assumed that both changed and unchanged properties will be passed into the request body. Returns: N/A
PostDatabaseItem	Usage: HTTP_Resource_Services('PostDatabaseItem', TableName, SelfURL, AllowedColumnNames) Comments: Creates a new resource item in the database table. To conform to the requirements of the POST method, the Key ID will be created automatically by the server and the entire resource will be created with the content being passed in. Returns: N/A
PatchDatabaseItem	Usage: HTTP_Resource_Services('PatchDatabaseItem', TableName, SelfURL, KeyID, AllowedColumnNames) Comments: Updates a specific resource item in the database table. To conform to the requirements of the PATCH method, only specific properties (e.g., database columns) of the resource will be updated. All properties not identified will be left alone. Returns: N/A

GetColumnNames	Usage: HTTP_Resource_Services('GetColumnNames', TableName, ColumnNames) Comments: Gets the column names from the dictionary of the table provided that will be used to create the resource. If the <i>ColumnNames</i> argument is used then this service will validate the list as valid columns. In either case, the following rules will be applied: • Regardless of the case used to in the <i>ColumnNames</i> argument it will be upper cased to match against the dictionary names correctly. • All column names will be returned in lower case. • All column names bound to a Key ID of part 0 (i.e., the entire key) will be removed. • All non-master column names will be removed to avoid duplicates. • SRP FrameWorks audit column names (MODIFIED_BY, MODIFIED_DATE, and MODIFIED_TIME) will be removed. Returns: An @FM delimited list of database column names.
GetColumnValues	Usage: HTTP_Resource_Services('GetColumnValues', TableName, ColumnNames, ItemID) Comments: Gets the column values from the database row from the provided table and ID that will be used to create the resource. This is returned as an @FM delimited list of values. Returns: An @FM delimited list of database column values for a specific Key ID.
GetMVGroupNames	Usage: HTTP_Resource_Services('GetMVGroupNames', TableName, ColumnNames) Comments: Gets the MV group names for the dictionary columns passed in. Note: the <i>ColumnNames</i> argument should be the same list that was returned from the <i>GetColumnNames</i> service. This will return an @FM delimited list of MV group names. Thus, for non-multivalue fields, the values will be empty. The MV group names will be returned in lower case. Gets the column values from the database row from the provided table and ID that will be used to create the resource. This is returned as an @FM delimited list of values. Returns: An @FM delimited list of MV group names.

Params

The proper use of the generic arguments are defined in the definition of each service above.