REST Webservice Methods

Below are all the SDK functions for registering REST methods and all the properties for describing them according to the OpenAPI standard.

Registering custom Webservice methods

```
SDK::setRestOperation($name, $handlerFilePath, $handlerMethodName, $params, $permission, $mcpSupport, $info)
```

- `$name`: method name called by REST webservice;
- `$handlerFilePath`: file path where the function is defined;
- `$handlerMethodName`: name of the function to use from the specified handler file;
- `$params`: if provided, the associative array of parameter names with their definition (see Parameter Definition (`$params`))
- `$permission` (since vtenext 23.08): one of “read”, “write” or “readwrite”, describing the kind of operation of this webservice. Used with additional accesskey for the users;
- `$mcpSupport` (since vtenext 26.XX): default to 0 ("disabled"), it can be set to 1 to allow the registration of this method to MCP servers;
- `$info` (since vtenext 26.XX): additional information used to display this method, for documentation purposes (see `$info` Parameter)

This function returns the id of the new Webservice if created successfully, and `false` otherwise.

```
SDK::setRestOperationInfo(string $name, array $operation = [], array $parameters = [], array $requestExamples = [], array $responses = [])
```

This function takes `name` as the name of the operation, and the arrays described in the `$info` Parameter section.

This function returns `true` if the webservice information was set up successfully, and `false` otherwise.

Parameter Definition (`$params`)

The `$params` function parameter is an associative array that has the name of the parameter as the key, and the type of the parameter.

The types currently registered are `string`, `encoded`, `datetime`, `double`, and `boolean`.

`$info` Parameter

The `$info` parameter of the function `SDK::setRestOperation` is an optional associative array that has the keys **`operation`**, **`parameters`**, **`examples`**, and **`responses`**.

This parameter is passed accordingly to `SDK::setRestOperationInfo`, which is the method responsible for filling the information for the Webservice method that is needed for the OpenAPI specification and the MCP servers.

`operation` Key

The `operation` key is an associative array with the following optional keys:

- `description`: `string`, human-readable description of the Webservice method;
- `tags`: `string`, comma-separated list of tags to group the Webservice method with;
- `tool_description`: `string`, specific description for MCP tool, if not provided the MCP server will default the generic description.

`parameters` Key

The `parameters` key is an associative array with the name of the parameter as the key, and an associative array that contains additional information, used for documentation purposes.

This information is used to further document the parameters of the webservice method, while still adhereing to the conventions already established with the use of this SDK.

The associative array for the parameter has the following, optional, attributes:

- `required`: `bool`, whether the parameter is mandatory;
- `default_value`: `mixed`, the default value of the parameter;
- `description`: `string`, informative description of the parameter;
- `example`: `mixed`, an example value of the parameter;
- `extra`: associative `array`, complementary information that can't be documented otherwise.

The `extra` attribute has the following, optional, attributes, that are of type `string` except when noted:

- `type`: used, for example, when the parameter is registered as `encoded`, allowing it to "cast" it as `object` or `array`;

- `schema`: the name of the schema definition that describes the structure of the parameter, used when `type` is `object`;
- `items`: required when `type` is `array`, specifies the typing of the values contained in the array; it can be a simple type such as `string`, `int`, `float`, `double`, `bool`, or the name of a schema definition;
- `enum`: `array` of values that are supported by the parameter.

`examples` Key

The `examples` key is a list of associative arrays that showcase a full usage of the webservice method. This key is optional.

The associative array of a single example has the following attributes:

- `name`: `string`, the name chosen for the example;
- `summary`: `string`, brief description of the example;
- `data`: associative `array`, the payload passed by the request.

`responses` Key

The `responses` key is an associative array with the status code of the response as the key, and an associative array that contains the information of the response.

The associative array of a response has the following optional attributes:

- `description`: `string`, brief description of the response;
- `fields`: associative `array`, the payload of the response, see `fields` Attribute
- `examples`: associative `array`, example representations of the response, see `examples` Key;
- `ref`: `string`, predefined information of the response

`fields` Attribute

The `fields` attribute is an associative array with the name of the field as the key, and an associative array that contains the information of the field.

The associative array of a field has the following optional attributes:

- `type`: the type of the field, can be `string`, `int`, `float`, `double`, `bool`, `object`, or `array`; if omitted, the field can be of "any" type;
- `schema`: if `type` is `object`, the name of the schema definition that describes the structure of the field;
- `items`: required when `type` is `array`, specifies the typing of the values contained in the array; it can be a simple type such as `string`, `int`, `float`, `double`, `bool`, or the name of a schema definition;
- `required`: `bool` whether the field is mandatory;

- `default`: `mixed`, the default value of the field;
- `description`: `string`, informative description of the field;
- `example`: `mixed`, an example value of the field;
- `enum`: `array` of values that are supported by the field.

Deregistering custom Webservice methods

```
SDK::unsetRestOperation(string $name)
```

- `$name`: name of the webservice method to delete.

This function returns `true` on success and `false` on failure.

```
SDK::unsetRestOperationInfo(string $name, string $type = 'all')
```

- `$name`: name of the webservice method whose information need to be deleted;
- `$type`: the type of information to delete, can be `all`, `operation`, `parameters`, `requestExamples`, `responses`, and `responseExamples`.

This function returns `true` on success and `false` on failure.

Registering custom Webservice schemas

```
SDK::setRestOperationSchema(string $name, array $fields)
```

- `$name`: name of the schema;
- `$fields`: see `fields` Attribute.

This function returns the id of the new webservice schema if created successfully, and `false` otherwise.

Deregistering custom Webservice schemas

```
SDK::unsetRestOperationSchema(string $name)
```

`$name` is the method name of the webservice schema to delete.

This function returns `true` on success and `false` on failure.

Revision #2
Created 2026-06-25 14:14:06 UTC by m.maporti
Updated 2026-10-20 09:53:21 UTC by m.maporti