

[Products](#) [Industries](#) [Support](#) [Training](#) [Community](#) [Developer](#) [Partner](#)

[About](#)



[Home](#) / [Community](#) / [Blogs](#)

[+ Actions](#)

# ABAP and Swagger/OpenAPI

February 3, 2018 | 110 Views | [Edit](#)



[Lars Hvam](#)

more by this author

ABAP Development

[ABAP Connectivity](#) | [openapi](#) | [swagger](#)



[Unfollow](#)

## Swagger

Swagger is a set of tools for creating, generating, documenting, and testing RESTful services.

It is centered around a specification file(<https://swagger.io/specification/>),

which contains the description of the REST services, much like a SOAP WSDL or OData metadata file.

There are many powerful tools in the Swagger family, which all are open source:

- [Editor](#)
- [UI](#)
- [Inspector](#)

These tools are free and works in the browser. SAP also use Swagger as part of [API management](#).

## ABAP

A [while ago](#) I started building a tool for integrating [ABAP and Swagger](#) using an inside-out approach.

Lets jump right into it and see how it works, suppose I want to expose the methods of the following class as REST services:

```
CLASS zcl_todo DEFINITION PUBLIC CREATE PUBLIC.

PUBLIC SECTION.
METHODS create
IMPORTING
    is_data      TYPE ztodo_data
RETURNING
    VALUE(rs_key) TYPE ztodo_key .
METHODS delete
IMPORTING
    is_key TYPE ztodo_key .
METHODS list
RETURNING
    VALUE(rt_list) TYPE ztodo_tt .
METHODS update
IMPORTING
    iv_guid TYPE ztodo_key-guid
    is_data TYPE ztodo_data .

ENDCLASS.

CLASS ZCL_TODO IMPLEMENTATION.

METHOD create.

DATA: ls_todo TYPE ztodo.

TRY.
    rs_key-guid = cl_system_uuid=>if_system_uuid_static~create_uu
CATCH cx_uuid_error.
    ASSERT 0 = 1.
ENDTRY.
```

```

MOVE-CORRESPONDING rs_key TO ls_todo.
MOVE-CORRESPONDING is_data TO ls_todo.

INSERT ztodo FROM ls_todo.
ASSERT sy-subrc = 0.

ENDMETHOD.

METHOD delete.

DELETE FROM ztodo WHERE guid = is_key-guid.
ASSERT sy-subrc = 0.

ENDMETHOD.

METHOD list.

SELECT * FROM ztodo INTO TABLE rt_list.          "#EC CI_NOWHE

ENDMETHOD.

METHOD update.

DATA: ls_todo TYPE ztodo.

ls_todo-guid = iv_guid.
MOVE-CORRESPONDING is_data TO ls_todo.

UPDATE ztodo FROM ls_todo.
ASSERT sy-subrc = 0.

ENDMETHOD.
ENDCLASS.

```

Interface ZIF\_SWAG\_HANDLER must be implemented, and metadata added for the methods to be exposed,

```

METHOD zif_swag_handler~meta.

FIELD-SYMBOLS: <ls_meta> LIKE LINE OF rt_meta.

APPEND INITIAL LINE TO rt_meta ASSIGNING <ls_meta>.
<ls_meta>-summary    = 'List'(001).
<ls_meta>-url-regex  = '/list$'.
<ls_meta>-method     = zcl_swag=>c_method-get.
<ls_meta>-handler    = 'LIST'.

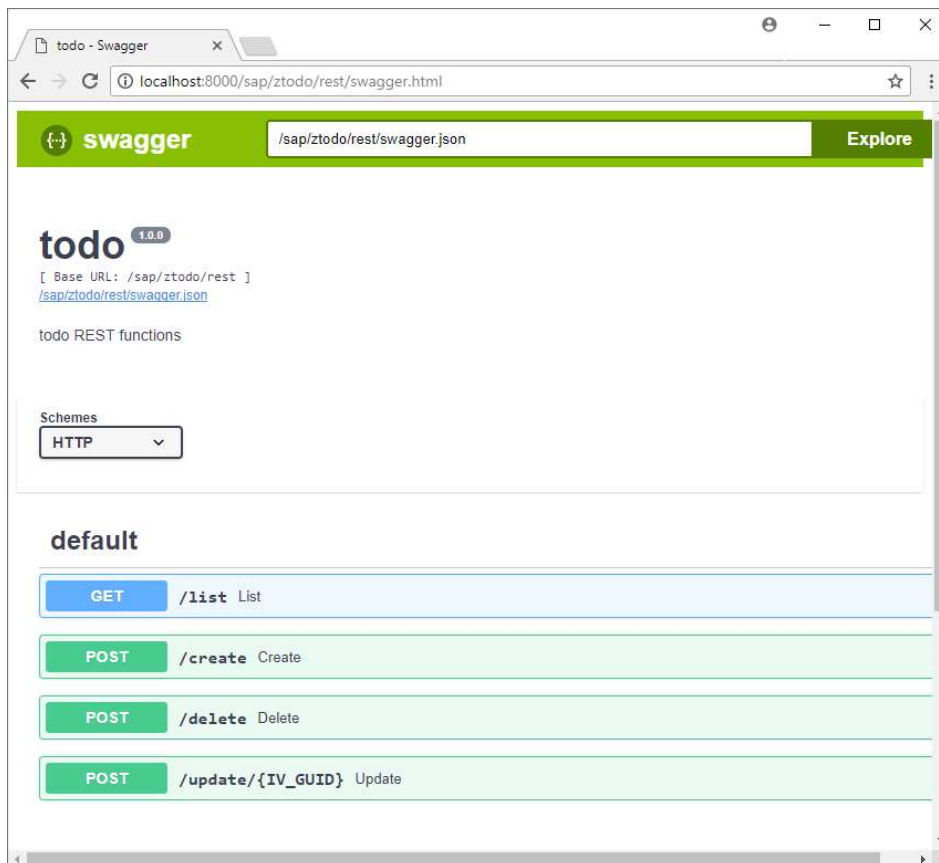
APPEND INITIAL LINE TO rt_meta ASSIGNING <ls_meta>.
<ls_meta>-summary    = 'Create'(002).
<ls_meta>-url-regex  = '/create$'.
<ls_meta>-method     = zcl_swag=>c_method-post.
<ls_meta>-handler    = 'CREATE'.

APPEND INITIAL LINE TO rt_meta ASSIGNING <ls_meta>.
<ls_meta>-summary    = 'Delete'(003).
<ls_meta>-url-regex  = '/delete$'.
<ls_meta>-method     = zcl_swag=>c_method-post.
<ls_meta>-handler    = 'DELETE'.

```

```
APPEND INITIAL LINE TO rt_meta ASSIGNING <ls_meta>.  
<ls_meta>-summary = 'Update'(004).  
<ls_meta>-url-regex = '/update/(w+)$'.  
APPEND 'IV_GUID' TO <ls_meta>-url-group-names.  
<ls_meta>-method = zcl_swag=>c_method-post.  
<ls_meta>-handler = 'UPDATE'.  
  
ENDMETHOD.
```

Add the handler to a custom SICF node, and thats about it! [ABAP-Swagger](#) will automatically generate a spec from the method definitions, and it is possible to use the Swagger UI to test the services. ABAP-Swagger is open source and works from 702 and up.



The full example can be found at [https://github.com/larshp/todo\\_logic](https://github.com/larshp/todo_logic)

Find more Open Source ABAP projects on [dotabap.org](https://dotabap.org)

*ABAP is like the sun, it will keep shining for a million years*

Alert Moderator

Be the first to leave a comment

Add Comment

Share & Follow



[Privacy](#)

[Terms of Use](#)

[Legal Disclosure](#)

[Copyright](#)

[Trademark](#)

[Sitemap](#)

[Newsletter](#)