

```

class Z_CL_MM_TEST_SERV definition
    public
    final
    create public .

public section.
*** public components of class Z_CL_MM_TEST_SERV
*** do not include other source files here!!!

    interfaces IF_HTTP_EXTENSION .
protected section.
*** protected components of class Z_CL_MM_TEST_SERV
*** do not include other source files here!!!
private section.
*** private components of class Z_CL_MM_TEST_SERV
*** do not include other source files here!!!

data LOG_INSTANCE type ref to ZCL_MM_APPLICATION_LOG .

methods GET_REST_METHOD
    importing
        !IO_SERVER type ref to IF_HTTP_SERVER
    returning
        value(RO_HANDLER) type ref to ZIF_REQUEST_HANDLER .
ENDCLASS.

```

```

CLASS Z_CL_MM_TEST_SERV IMPLEMENTATION.

```

```

* <SIGNATURE>-----
* | Instance Private Method Z_CL_MM_TEST_SERV->GET_REST_METHOD
* +-----
* | [--->] IO_SERVER                                TYPE REF TO IF_HTTP_SERVER
* | [<-( )] RO_HANDLER                             TYPE REF TO ZIF_REQUEST_HANDLER
* +-----</SIGNATURE>

method GET_REST_METHOD.
    Constants: lv_class_prefix type string value 'ZCL_MM_HANDLE_'.

DATA: lt_header_fields type TIHTTPNPV,
      ls_header_fields like line of lt_header_fields,
      lv_class_name type string,
      lo_handler type ref to zif_request_handler,
      lx_create_obj type ref to CX_SY_CREATE_OBJECT_ERROR,
      msg_txt type string.

io_server->request->get_header_fields( changing fields = lt_header_fields ).

read table lt_header_fields into ls_header_fields with key name = '~request_method'.

concatenate lv_class_prefix ls_header_fields-value into lv_class_name.

try.
create object lo_handler
    type (lv_class_name)
exporting
    IO_REQUEST = io_server->request
    io_response = io_server->response.

```

```

catch CX_SY_CREATE_OBJECT_ERROR into lx_create_obj.
    io_server->response->set_status( exporting code = '400'
                                     reason = 'Invalid HTTP Method' ).

    create object me->log_instance.
    me->log_instance->add_log( exporting iv_message_no = '009'
                              iv_message_type = 'E' ).

endtry.

ro_handler = lo_handler.

endmethod.

* <SIGNATURE>-----
* | Instance Public Method Z_CL_MM_TEST_SERV->IF_HTTP_EXTENSION~HANDLE_REQUEST
* +-----
* | [--->] SERVER                                TYPE REF TO IF_HTTP_SERVER
* +-----</SIGNATURE>
METHOD if_http_extension~handle_request.

DATA: lo_handler type ref to zif_request_handler.

lo_handler = get_rest_method( io_server = server ).
if lo_handler is bound.
    lo_handler->handle_request( ).
endif.
ENDMETHOD.
ENDCLASS.

```