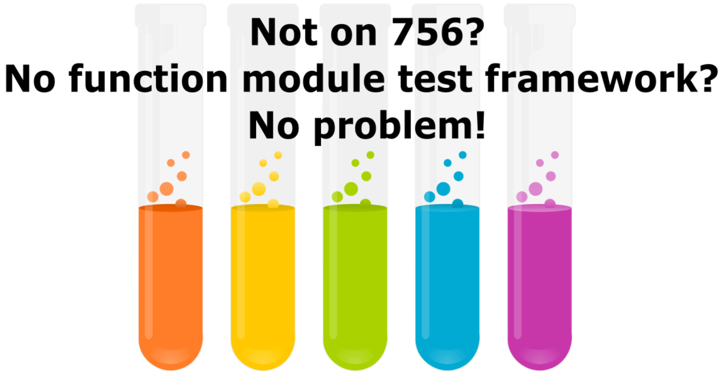
Edit article

View post



Not on 756? No function module test framework? No problem!

Lars Hvam Petersen
Senior SAP/ABAP Consultant at Heliconia Labs

4 articles

June 2, 2024

Open Immersive Reader

TLDR: Use it standalone on GitHub actions, move the unit tests into the system when its upgraded to 756+

The Function Module Test Framework allows replacing/doubling functionality in function modules during unit test execution, this is sometimes very useful when testing logic utilizing function modules.

Its also possible to manually wrap the function module in a class + interface + factory for dependency injection, but it easily becomes a lot of scaffolding to call a single function module.

How does it work?

The below example outlines how it works overall inside a unit test. First setup the function modules to be doubled, when called during unit test, the function module is redirected to "METHOD if_ftd_invocation_answer~answer"

```
  METHOD test.

    DATA lt_deps    TYPE if_function_test_environment=>tt_function_dependencies.
    DATA li_env     TYPE REF TO if_function_test_environment.
    DATA lv_message TYPE string.

    INSERT 'ABC' INTO TABLE lt_deps.
    li_env = cl_function_test_environment=>create( lt_deps ).

    li_env->get_double( 'ABC' )->configure_call( )->ignore_all_parameters( )->then_answer( me ).

    CALL FUNCTION 'ABC'
      EXPORTING
        integer = 2
      IMPORTING
        message = lv_message.

    cl_abap_unit_assert=>assert_equals(
      act = lv_message
      exp = gc_hello_world ).

    li_env->clear_doubles( ).

  ENDMETHOD.

  METHOD if_ftd_invocation_answer~answer.

    DATA ref TYPE REF TO data.

    FIELD-SYMBOLS <fs> TYPE any.

    ref = arguments->get_importing_parameter( 'INTEGER' ).
    ASSIGN ref->* TO <fs>.

    cl_abap_unit_assert=>assert_equals(
      act = <fs>
      exp = 2 ).

    result->get_output_configuration( )->set_exporting_parameter(
      name  = 'MESSAGE'
      value = gc_hello_world ).

  ENDMETHOD.
```

What's the problem?

The framework is delivered as part of 756, not everyone is upgraded to that version yet. But every developer needs to test functionality in the system in an easy way.

What's the solution?

I suggest writing and running tests outside of the real system, and when the system is upgraded, then move the test cases into the system.

open-abap can run ABAP code standalone, and editing ABAP files is made easy through the Standalone ABAP Development Extension Pack.

The above example is a testcase in open-abap, which runs on GitHub actions for free, without access to any internal corporate network.

🤠

Published by

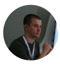
Lars Hvam Petersen
Senior SAP/ABAP Consultant at Heliconia Labs
Published • 5s

4 articles

Not on 756? No function module test framework? No problem! #abap

Like Comment Share

0 Comments



Add a comment...

Lars Hvam Petersen
Senior SAP/ABAP Consultant at Heliconia Labs

More from Lars Hvam Petersen

Stuff I'd like removed from the #ABAP roadmap

Lars Hvam Petersen on Linke...

Running abap2UI5 backend in browser

Lars Hvam Petersen on Linke...

Running Scrypt in Rust on ABAP on Raspberry PI Zero

Lars Hvam Petersen on Linke…