Personal Insights

Lars Hvam

July 5, 2021 1 minute read

Thoughts on stable ABAP APIs

Follow RSS feed Like

0 Likes 7 Views 0 Comments Edit

In this blog post, let's look at **APIs** inside an ABAP system. A API in ABAP can be seen as a set of ABAP objects, for simplicity take DOMA + DTEL + TABL + TTYP + CLAS + INTF as an example.

I consider changes to an API to be **stable** if the changes in the API do not cause errors in the ABAP programs that consume the API.

Consistency

The set of objects in the API must be consistent, all objects referred must also be considered part of the API.

Example: The following interface refers to DTEL MANDT, which refers to DOMA MANDT, both being part of the definition for the "moo" method, and thus should be considered part of the API.

```
INTERFACE zif_bar.
  METHODS moo IMPORTING iv_mandt TYPE mandt.
ENDINTERFACE.
```

The consumer of an API must be able to lookup all the details by looking just at the API, nothing more, nothing less.

Implementation code

Method implementation code is not part of the API, it is up to the provider of the API to have any implementation, which can change independently of the consumer.

Example: This implementation code refers to TABL T000, but its implementation code so not part of the API provided by CLAS ZCL_BAR

```
CLASS zcl_bar DEFINITION.
  PUBLIC SECTION.
    INTERFACES zif_bar.
ENDCLASS.

CLASS zcl_bar IMPLEMENTATION.
  METHOD zif_bar~moo.
    DATA impl TYPE t000.
  ENDMETHOD.
ENDCLASS.
```

Use of the API is possible

If an interface is part of the API, the consumer must be able to implement the interface. Likewise, if a method is part of the API, a consumer must be able to call the API.

This easily follows if the API is consistent, but as ABAP is a large language, it takes effort to catch everything that should be part of the API.

Building new APIs

When building new APIs consider the following:

INTF should not reference CLAS objects, do proper OO design and define interfaces properly.

Don't add Function Groups/Function Modules as part of an API, function modules have been considered obsolete for many years.

Stable changes

Following changes(and more?) are possible to make while keeping the API stable,

DOMA

- Compatible type changes, for example: extending a field from 20 characters to 30 characters

DTEL

- Changing to a type compatible domain
- Changing descriptions, it will not break the consumer programs

TABL

- Adding fields

TTYP

- Type compatible changes

INTF

- New optional parameters in methods

CLAS

- New methods
- New optional parameters in methods

Unstable changes

DOMA + DOMA

- Incompatible type changes

TABL

- Removing fields
- Renaming fields
- Incompatible type changes

TTYP

- Setting a new table key

INTF

- New methods(might cause syntax error in consumer implementations)
- Removing methods
- Removing mandatory method parameters
- Adding INTERFACES

CLAS

- Removing methods
- Removing mandatory method parameters
- Adding INTERFACES

And I probably missed a lot of details, comments welcome below 🙂

Alert Moderator

Assigned tags

ABAP Development | #ABAP |

Related Blog Posts

Implementation Patterns for CDS Views in SAP S/4HANA at SAP TechEd 2017

By **Carine Tchoutouo Djomo** , Jan 30, 2018

First version: SDK for ABAP development tools

By **Michael Schneider** , May 10, 2013

Monday morning thoughts: more on ABAP in the cloud

By **DJ Adams** , Oct 08, 2018

Related Questions

ABAP APIs

By **Former Member** , Apr 15, 2008

Is SORT in ABAP stable?

By **Former Member** , Jun 04, 2013

your thoughts on the new ABAP editor

By **Former Member** , Jun 27, 2007

Be the first to leave a comment

Add Comment

Find us on