Behind XCO

… and much further

Software-Heroes

Agenda

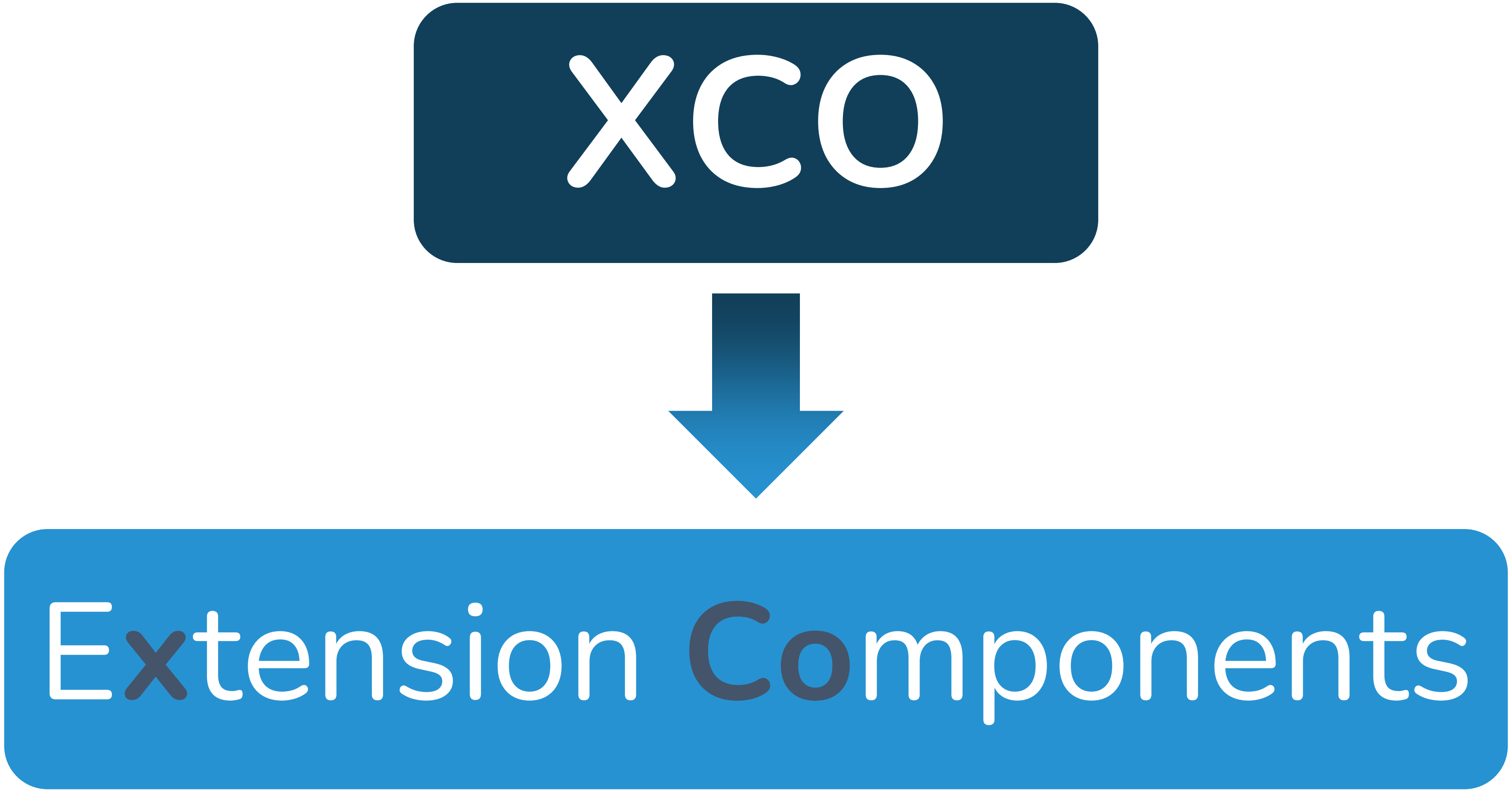
1. Introduction
2. Overview
3. Examples
4. Conclusion

Photo from Jakub Dziubak on Unsplash

Introduction

Software-Heroes

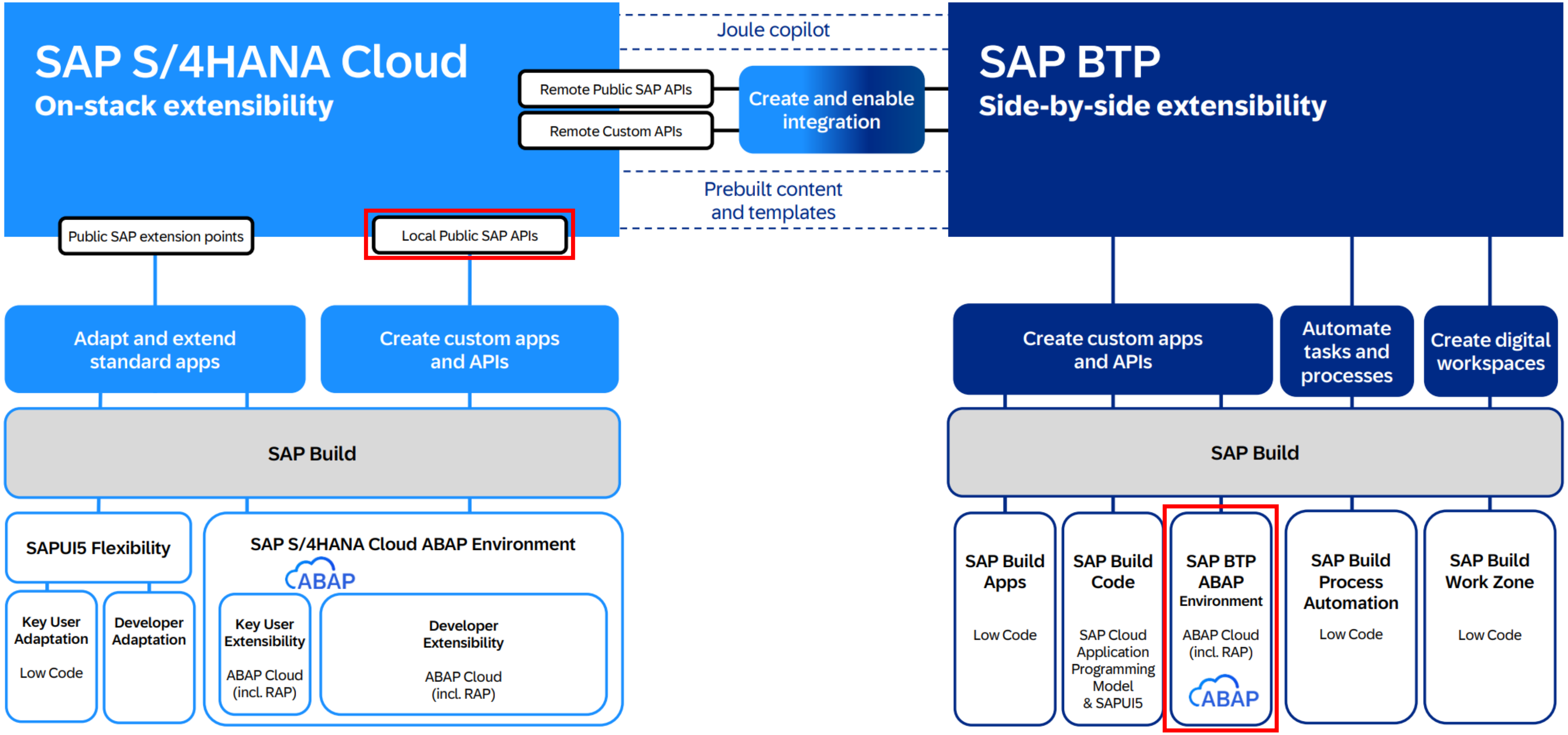
Introduction



XCO

Extension Components

Introduction



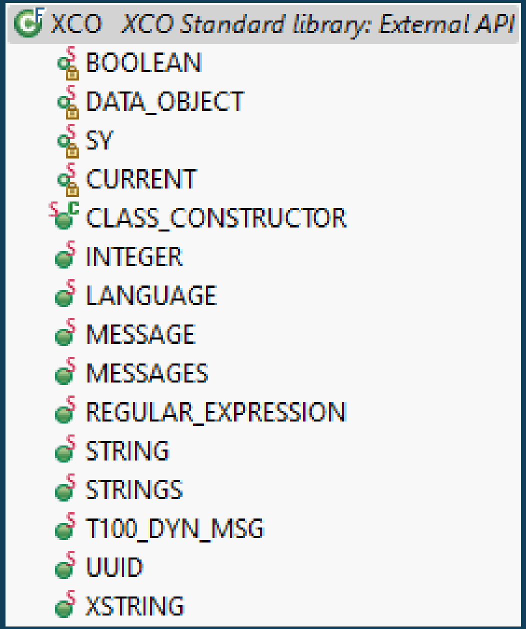
Software-Heroes

Photo from Markus Spiske on Unsplash

Overview

Overview

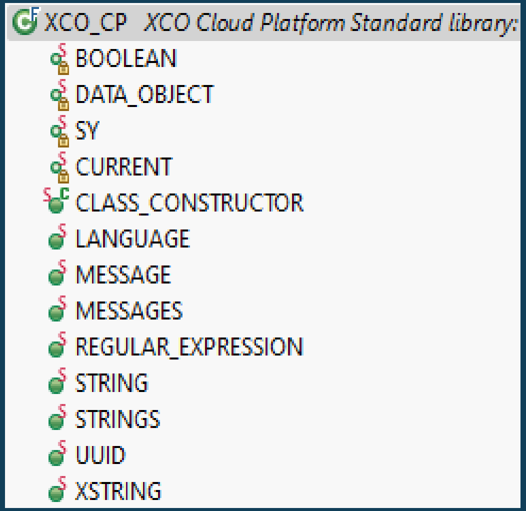
Versions



XCO   *XCO Standard library: External API*

- BOOLEAN
- DATA_OBJECT
- SY
- CURRENT
- CLASS_CONSTRUCTOR
- INTEGER
- LANGUAGE
- MESSAGE
- MESSAGES
- REGULAR_EXPRESSION
- STRING
- STRINGS
- T100_DYN_MSG
- UUID
- XSTRING

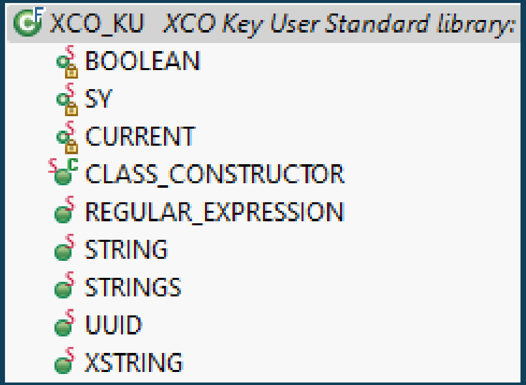
Standard ABAP



XCO_CP   *XCO Cloud Platform Standard library:*

- BOOLEAN
- DATA_OBJECT
- SY
- CURRENT
- CLASS_CONSTRUCTOR
- LANGUAGE
- MESSAGE
- MESSAGES
- REGULAR_EXPRESSION
- STRING
- STRINGS
- UUID
- XSTRING

ABAP Cloud



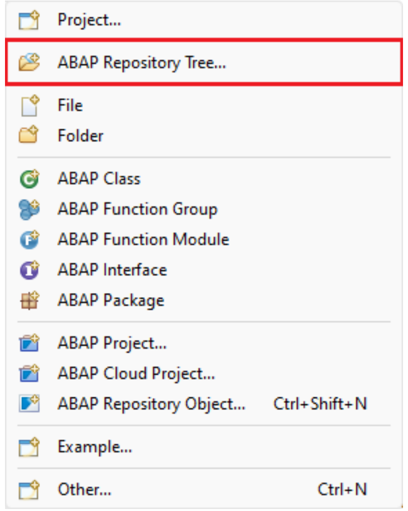
XCO_KU   *XCO Key User Standard library:*

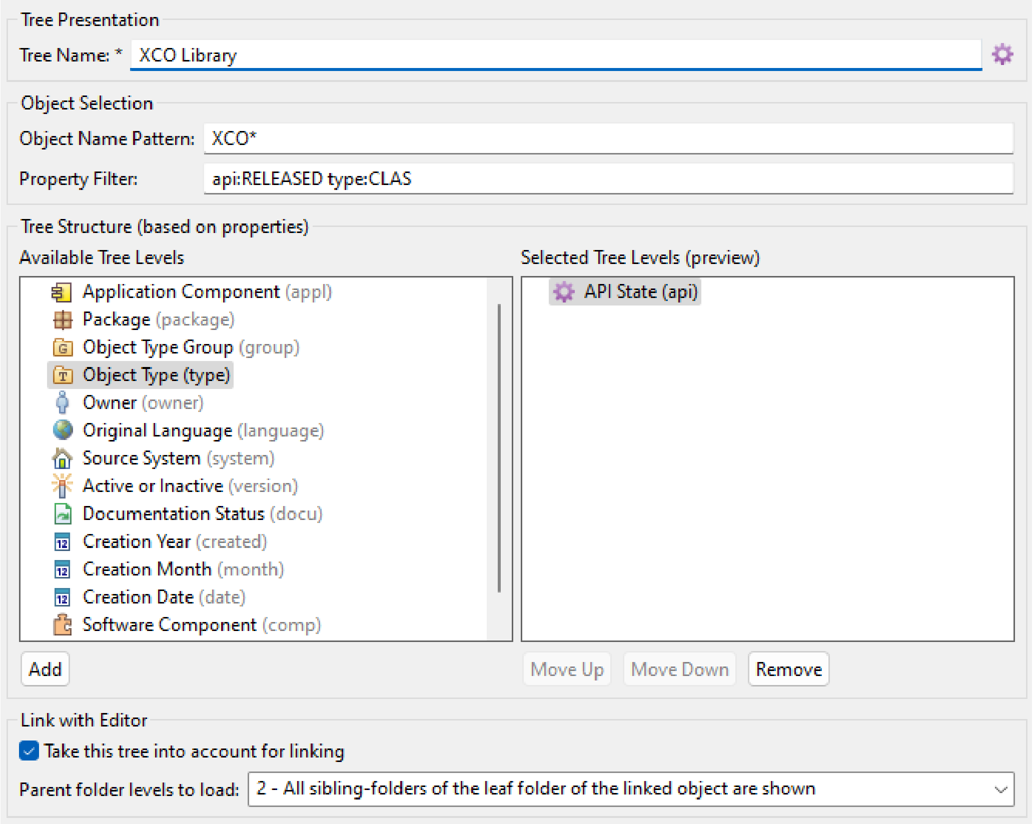
- BOOLEAN
- SY
- CURRENT
- CLASS_CONSTRUCTOR
- REGULAR_EXPRESSION
- STRING
- STRINGS
- UUID
- XSTRING

Key User Apps

Overview

List in ADT




Tree Presentation

Tree Name: * XCO Library

Object Selection

Object Name Pattern: XCO*

Property Filter: api:RELEASED type:CLAS

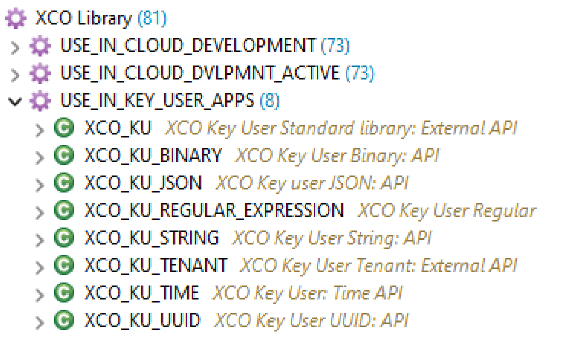
Tree Structure (based on properties)

Selected Tree Levels (preview): API State (api)

Link with Editor

☑ Take this tree into account for linking

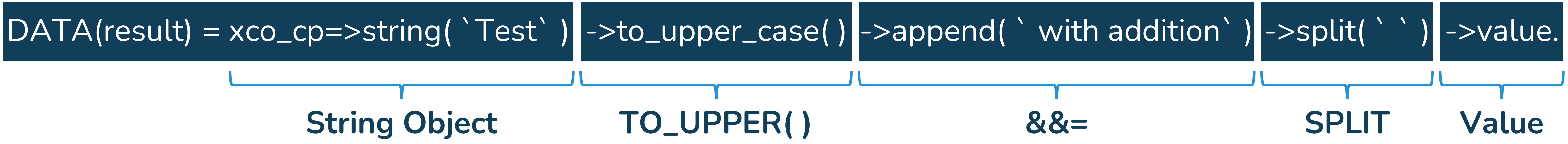
Parent folder levels to load: 2 - All sibling-folders of the leaf folder of the linked object are shown


XCO Library (81)
- USE_IN_CLOUD_DEVELOPMENT (73)
- USE_IN_CLOUD_DVLPMNT_ACTIVE (73)
- USE_IN_KEY_USER_APPS (8)
  - XCO_KU *XCO Key User Standard library: External API*
  - XCO_KU_BINARY *XCO Key User Binary: API*
  - XCO_KU_JSON *XCO Key user JSON: API*
  - XCO_KU_REGULAR_EXPRESSION *XCO Key User Regular*
  - XCO_KU_STRING *XCO Key User String: API*
  - XCO_KU_TENANT *XCO Key User Tenant: External API*
  - XCO_KU_TIME *XCO Key User: Time API*
  - XCO_KU_UUID *XCO Key User UUID: API*

Software-Heroes

Overview

Use in development

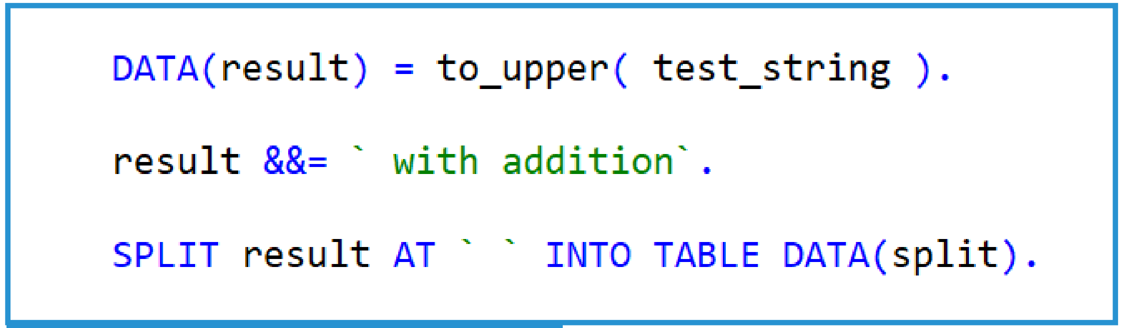


```
DATA(result) = xco_cp=>string( `Test` ) ->to_upper_case( ) ->append( ` with addition` ) ->split( ` ` ) ->value.
```

Factory for Instancing

Fluent Interface Pattern

VALUE for access



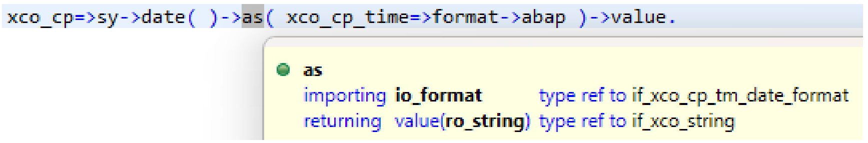
```
DATA(result) = to_upper( test_string ).

result &&= ` with addition`.

SPLIT result AT ` ` INTO TABLE DATA(split).
```

ABAP native statement

Overview

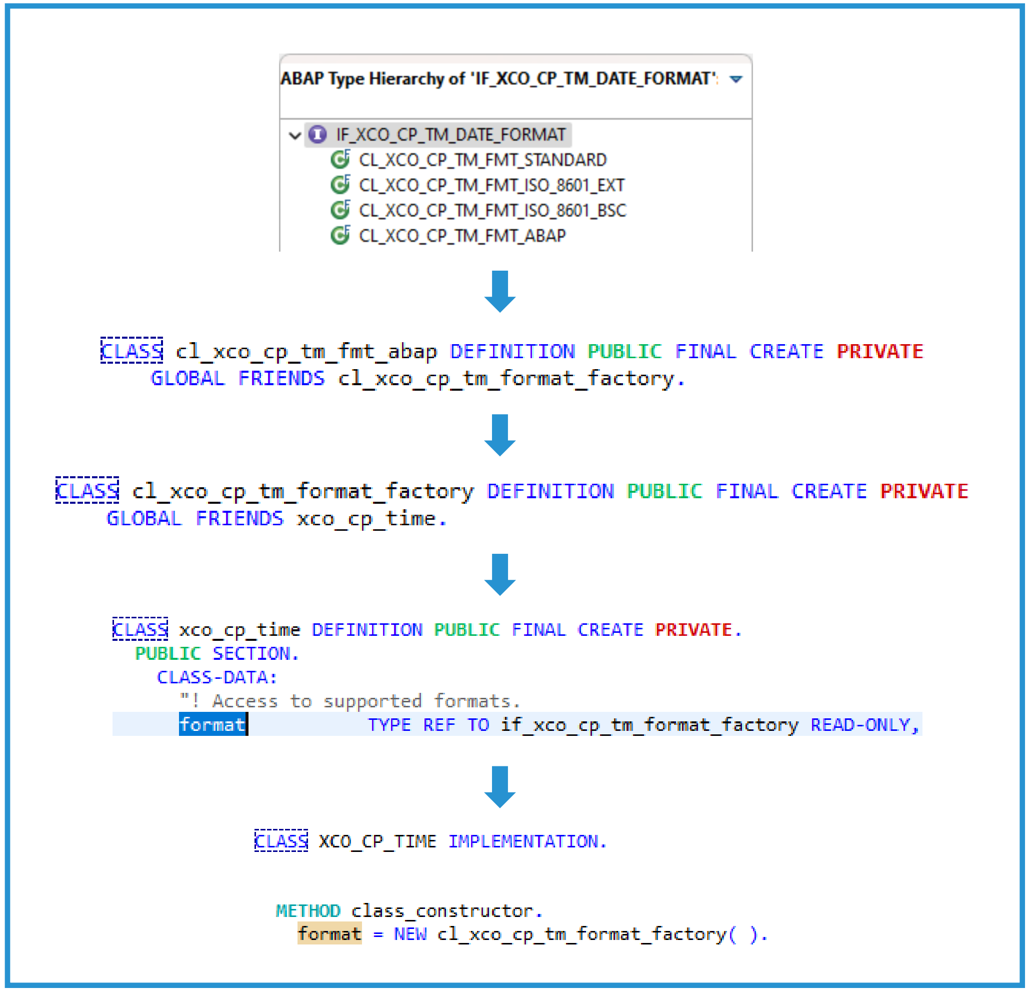
Find the correct options

- Get a formatter for the AS method of class
- Along the object hierarchy
- Public attribute as Factory



```
xco_cp=>sy->date( )->as( xco_cp_time=>format->abap )->value.
```

```
as
importing io_format          type ref to if_xco_cp_tm_date_format
returning  value(ro_string)  type ref to if_xco_string
```



ABAP Type Hierarchy of 'IF_XCO_CP_TM_DATE_FORMAT'.

```
IF_XCO_CP_TM_DATE_FORMAT
  CL_XCO_CP_TM_FMT_STANDARD
  CL_XCO_CP_TM_FMT_ISO_8601_EXT
  CL_XCO_CP_TM_FMT_ISO_8601_BSC
  CL_XCO_CP_TM_FMT_ABAP
```

```
CLASS cl_xco_cp_tm_fmt_abap DEFINITION PUBLIC FINAL CREATE PRIVATE
  GLOBAL FRIENDS cl_xco_cp_tm_format_factory.
```

```
CLASS cl_xco_cp_tm_format_factory DEFINITION PUBLIC FINAL CREATE PRIVATE
  GLOBAL FRIENDS xco_cp_time.
```

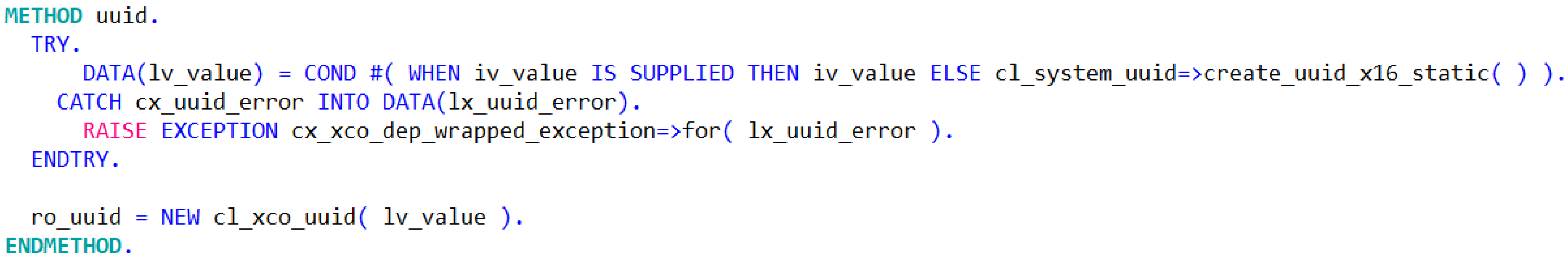
```
CLASS xco_cp_time DEFINITION PUBLIC FINAL CREATE PRIVATE.
  PUBLIC SECTION.
    CLASS-DATA:
      "! Access to supported formats.
      format          TYPE REF TO if_xco_cp_tm_format_factory READ-ONLY,
```

```
CLASS XCO_CP_TIME IMPLEMENTATION.


  METHOD class_constructor.
    format = NEW cl_xco_cp_tm_format_factory( ).
```

Overview

Catch nothing

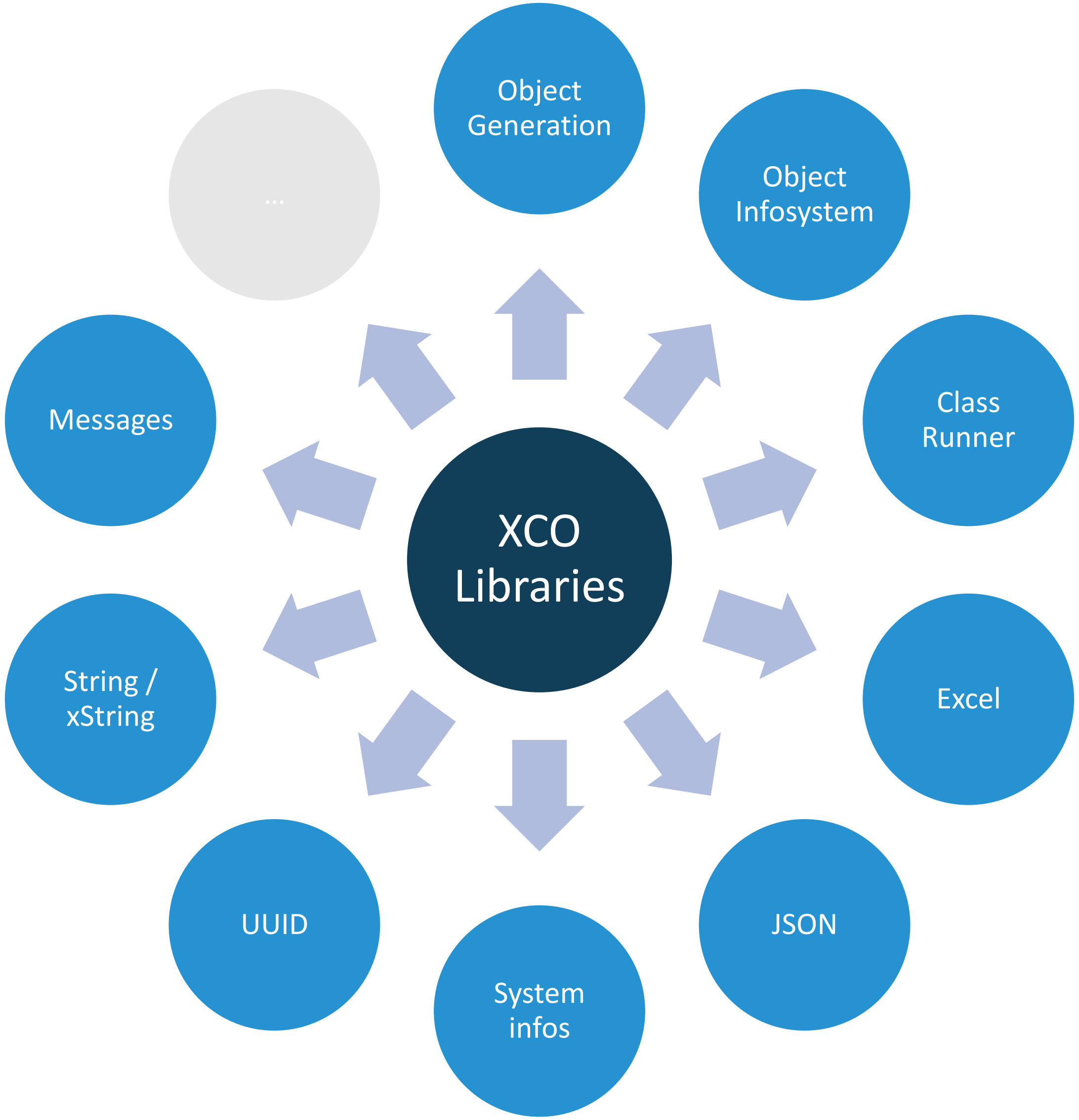
- Many exceptions are catched internally
- Inherited from CX_NO_CHECK



```
METHOD uuid.
  TRY.
      DATA(lv_value) = COND #( WHEN iv_value IS SUPPLIED THEN iv_value ELSE cl_system_uuid=>create_uuid_x16_static( ) ).
    CATCH cx_uuid_error INTO DATA(lx_uuid_error).
      RAISE EXCEPTION cx_xco_dep_wrapped_exception=>for( lx_uuid_error ).
  ENDTRY.

  ro_uuid = NEW cl_xco_uuid( lv_value ).
ENDMETHOD.
```

Overview

Libraries



Software-Heroes

Photo from Dan Cristian Pădureț on Unsplash

Examples

Examples

Software-Heroes

Foto von Joshua Hoehne auf Unsplash

Conclusion

Software-Heroes

Conclusion

Learning

- Hard to learn and understand (coming from Classic ABAP)
- Know the different types and functions

Availability

- All S/4HANA Editions (Public Cloud, Private Cloud, On-Prem)
- ABAP environment

Wishes

- Close the gap to existent classes and functions
- Add missing functions

Software-Heroes

Thank you

Download

LinkedIn

Software-Heroes