

Behind XCO

... and much further

Agenda

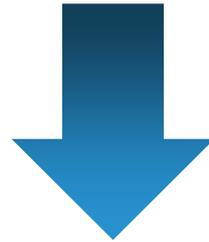
1. Einführung
2. Überblick
3. Beispiele
4. Zusammenfassung



Photo from Jakub Dziubak on Unsplash

Einführung

XCO



Extension Components

Einführung

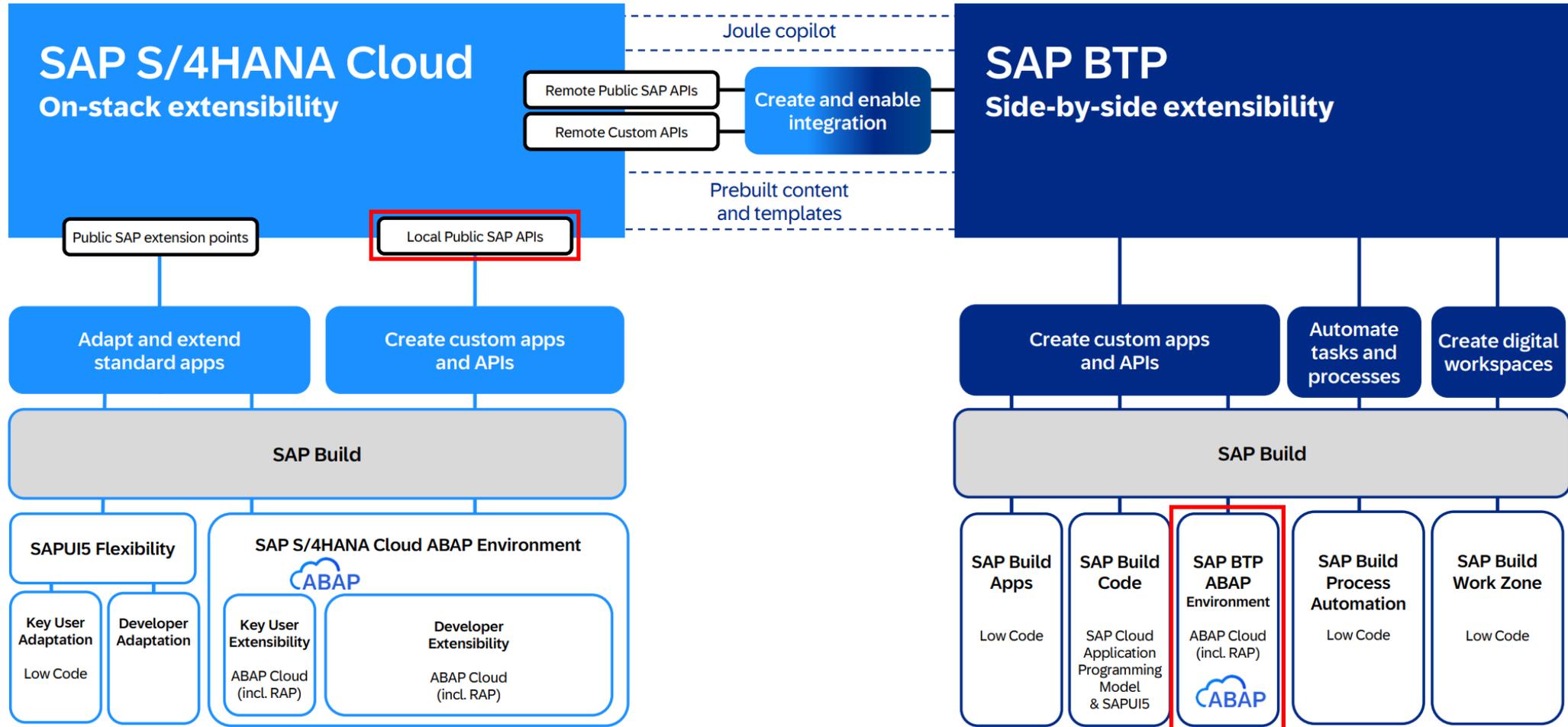




Photo from Markus Spiske on Unsplash

Überblick

Überblick

Versionen

 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

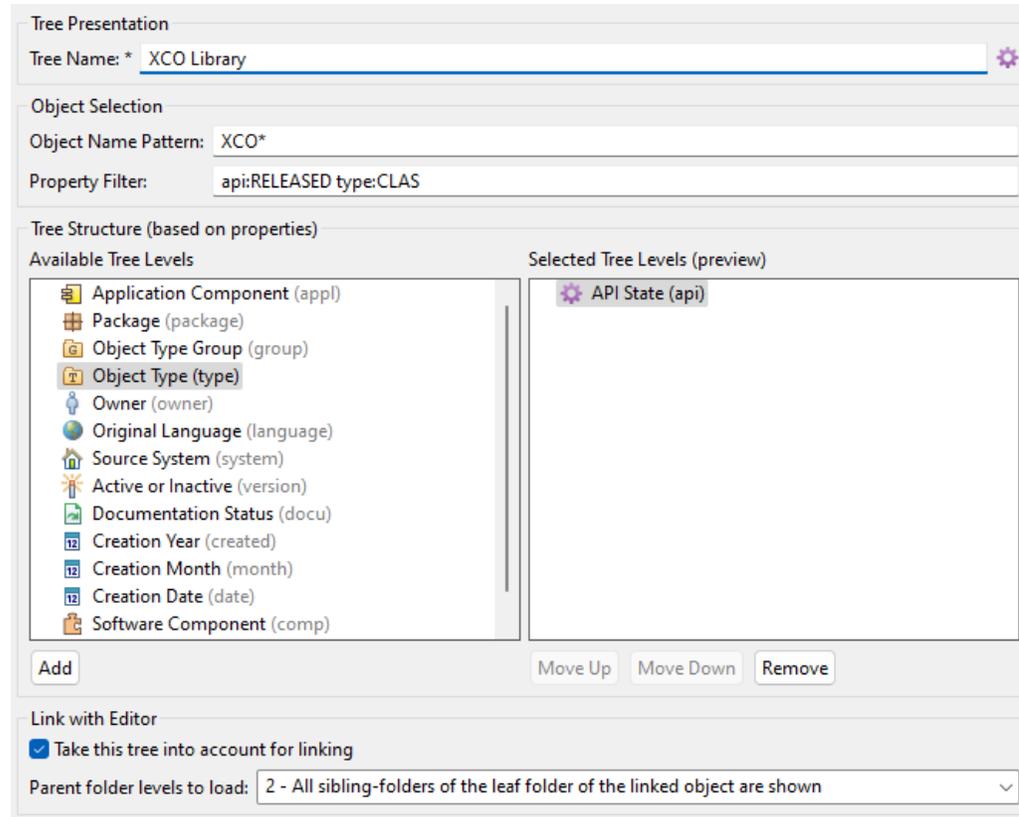
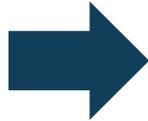
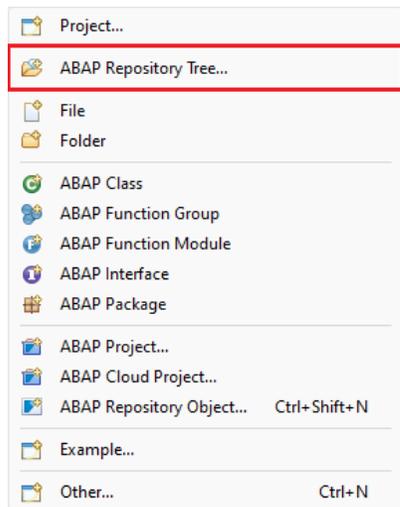
 XCO_KU XCO Key User Standard library:

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

Key User Apps

Überblick

Übersicht in ADT



- ⚙ XCO Library (81)
 - > ⚙ USE_IN_CLOUD_DEVELOPMENT (73)
 - > ⚙ USE_IN_CLOUD_DVLPMT_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

Überblick

Nutzung in der Entwicklung

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

String Object TO_UPPER() &&= SPLIT Value

Factory für Instanziierung

Fluent Interface Pattern

VALUE für den Zugriff

```
DATA(result) = to_upper( test_string ).  
result &&= ` with addition` .  
SPLIT result AT ` ` INTO TABLE DATA(split).
```

ABAP natives Statement

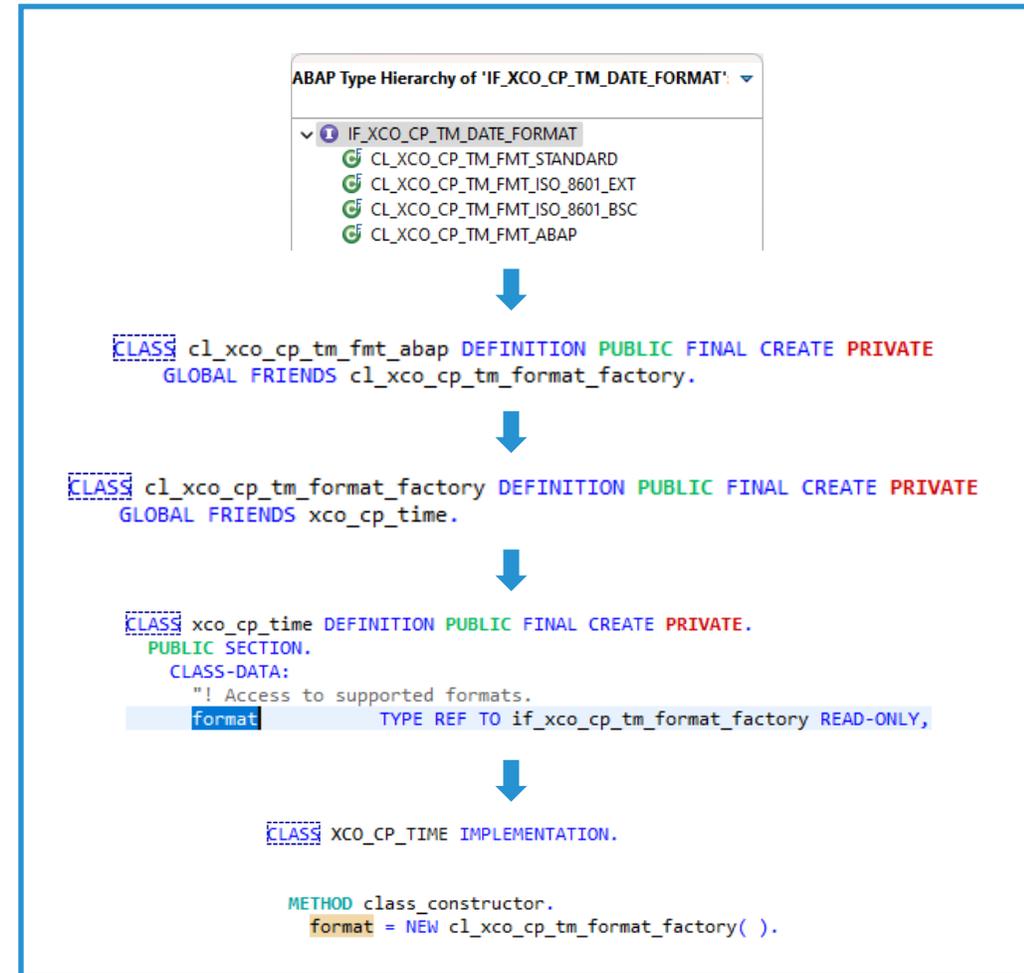
Überblick

Korrekte Einstellung finden

- Die korrekte Formatierung für die AS Methode finden
- Entlang der Objekthierarchie
- Öffentliche Attribute als 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
```



Überblick

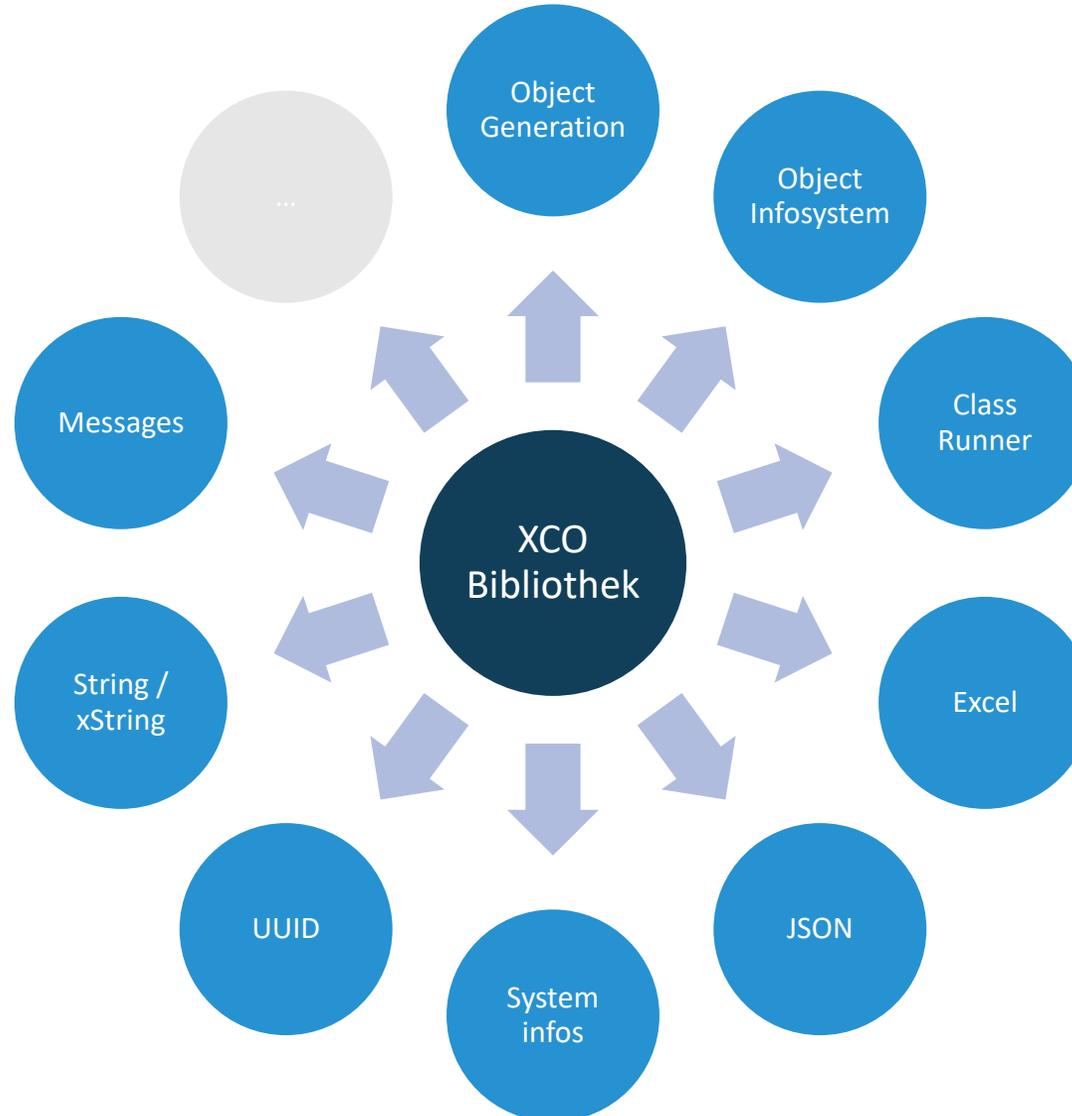
Nichts abfangen

- Viele Ausnahmen werden intern abgefangen
- Geerbt von 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 ).  
  ENTRY.  
  
  ro_uuid = NEW cl_xco_uuid( lv_value ).  
ENDMETHOD.
```

Überblick

Bibliotheken



Beispiele





Foto von Joshua Hoehne auf Unsplash

Zusammenfassung

Zusammenfassung



Lernen

- Hart zu lernen und zu verstehen (Klassisches ABAP)
- Unterschiede der Funktionen kennen



Verfügbarkeit

- Alle S/4HANA Editions (Public Cloud, Private Cloud, On-Prem)
- ABAP Environment

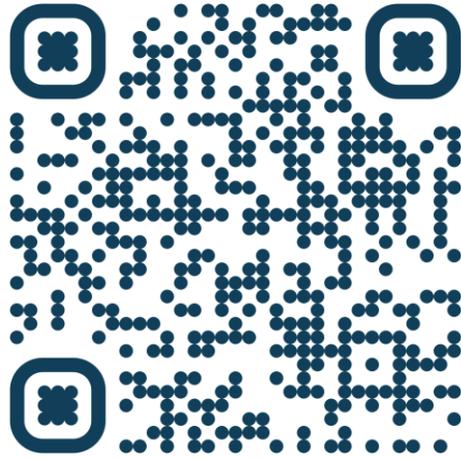


Wünsche

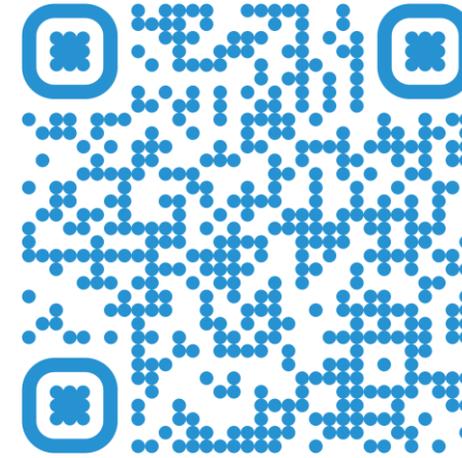
- Feature Gap zu Funktionen schließen
- Fehlende Funktionen implementieren



Danke



Download



LinkedIn