



Dynamische Programmierung in ABAP RAP

Webinar, 28.3.2025, 10:00 Uhr

29. Jänner 2016

2. April 2020

40

Agenda

Einleitung

Dynamic Feature Control

Dynamic EML Operations

Type Mapping for RAP

Dynamische Programmierung in ABAP RAP

Sören Schlegel

SAP Development Architect

Twitter: [@SoSchlegel87](https://twitter.com/SoSchlegel87)

LinkedIn: <https://www.linkedin.com/in/soschlegel>

Themenschwerpunkte

- Konzeption & Management von Entwicklungsprojekten
- SAP Architektur moderner Anwendungen
- ABAP RAP und S/4HANA Entwicklung
- SAP CDS, OData und Cloud
- **SAP Champion** 





Johann Föbleitner

Development Architect

eMail: johann.foessleitner@cadaxo.com

LinkedIn: <https://www.linkedin.com/in/foessleitner/>

Themen

- Clean Core, 3-Tier Model, Clean Code
- Fiori, Fiori Elements & ABAP RESTful
- S/4HANA Custom Development
- Entwickler – Trainer - Coach
- **SAP Champion** 🏆
- ABAPConf, Stammtische, Inside Tracks, Code Jams, Code Retreats



Domi Bigl

Senior Consultant at Cadaxo

eMail: dominik.bigl@cadaxo.com

LinkedIn: <https://www.linkedin.com/in/domi-bigl>

Beratungsschwerpunkte

- Konzeption & Management von Entwicklungsprojekten
- Qualitymanagement & Performanceoptimierung
- ABAP Units
- SAP Fiori, SAP UI5

- **ABAP, UI5, ... Champion**



- **Consulting, Workshops, Coaching, Produkte**
- ABAP, CDS/RAP, S/4 Extension, Custom Code Migration, SQL Cockpit, ...
- <https://www.cadaxo.com>

Brandeis Consulting

- **Consulting, Schulungen, Workshops**
- ABAP, CDS/RAP, SQLScript, ...
- <https://www.brandeis.de>



<https://www.s4fit.net/>

Community News

ABAP Webinare 2025 - Termine

Freitag, 31. Jänner 2025

Freitag, 28. März 2025

Mittwoch, 28. Mai 2025 ?? ABAP DevDays

Freitag, 26. September 2025

Freitag, 28. November 2025

Immer um 10:00 Uhr

<https://www.cadaxo.com/webinare/>



- **SAP Stammtisch Wien**

- Monatliches Treffen irgendwo in Wien
- Organisiert über die SAP Location Community Group
- Nächster Termin: 3. April 2025 - Fladerei





- **DSAG Technologietage**
 - 2. / 3. April 2025
 - 4. April DSAG-TechXchange
 - RheinMain CongressCenter, Wiesbaden
 - Keynote mit Philipp Herzig
 - 690 – 990 €





- **DSAG ABAP Development Days**
 - 27. und 28. Mai 2025
 - Palatin Kongresshotel Wiesloch
 - Clean Core Extensibility (ABAP Cloud & AI)
 - Mix aus Vorträgen, Demos, Übungen, ...
 - 545 € (für Mitglieder)



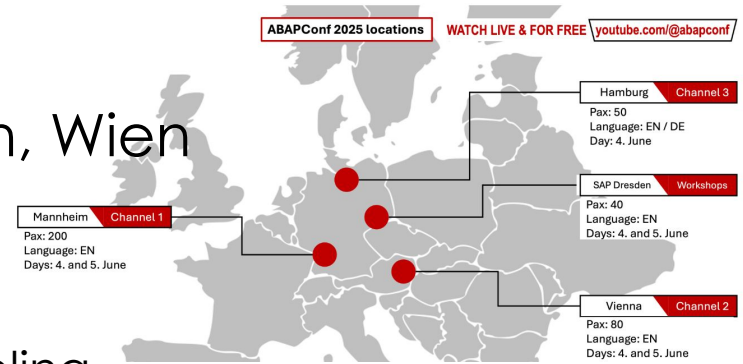


• ABAPConf 2025

- 4. und 5. Juni 2025
- Free | Live | Onsite | Online
 - Mannheim, Hamburg, Dresden, Wien

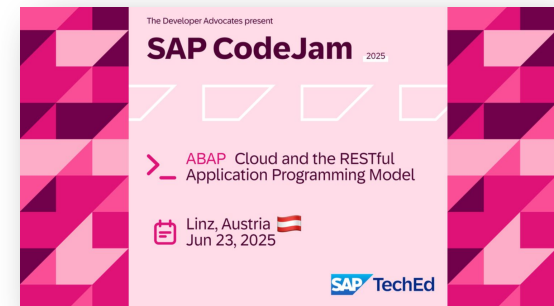
◦ <https://abapconf.org/abapconf2025/>

- Business Configurations, CDS Data Modeling, Design Patterns, Clean Core in ABAP, Building Modern Fiori Elements, XCO, ...





- **SAP CodeJam – ABAP Cloud & RAP**
 - 23. Juni, Linz 09:00 – 16:00 Uhr
 - Host: **Develite IT Solutions** aus Linz
 - Instructor: **Andre Fischer**
 - Adresse: TechCenter Hafenstrasse 47-51
4020 Linz AUSTRIA



• Unsere bisherigen RAP Webinare



Dynamische Entwicklung in ABAP

- **Dynamische ABAP Entwicklung**
 - Data Referenzen und Field Symbols
 - ABAP SQL
 - Methoden und Funktionsbausteinen
 - ABAP EML
 - String Templates
 - RTTS (Runtime Type Services)
 - AMDP, ADBC
 - Subroutinenpool (nur im Ausnahmefall)



ABAP Dynamische Programmierung

29. September 2023

ABAP SQL, Subroutinen Pool, RTT?,
dynamisches Assign, dynamische Aufrufe,

...

- **Dynamisches ABAP RAP**
 - Dynamic Feature Control - Sören
 - Dynamic EML Operations - FöB
 - Type Mapping – Domi

Dynamic Feature Control

Dynamic EML Operations

- **Entity Manipulation Language**

- EML ist ein ABAP Sprachkonstrukt mit dem wir auf RAP-Objekte zugreifen können
 - Lesen, Schreiben, Löschen, Aktionen
- Berücksichtigt alle Entity-Einstellungen
 - Validierungen
 - Berechtigungen
 - ...

```
READ ENTITIES OF /dmo/i_travel_m
  ENTITY travel
  FIELDS ( travel_id agency_id customer_id )
  WITH VALUE #( ( %key-travel_id = 1 )
                ( %key-travel_id = 6 ) )
  RESULT DATA(travels).
```

- **Dynamisches EML**
 - READ ENTITIES
 - GET PERMISSIONS
 - MODIFY ENTITIES
 - CREATE, UPDATE, DELETE, EXECUTE
 - SET LOCKS
 - COMMIT ENTITIES

• READ ENTITIES OPERATIONS

- Zusammenfassen mehrere Leseoperationen
- Tabellentyp für Operations: abp_behv_retrievs_tab

```
DATA read_travel      TYPE TABLE FOR READ IMPORT /dmo/i_travel_m.
DATA op_tab          TYPE abp_behv_retrievals_tab.
DATA read_travel_result TYPE TABLE FOR READ RESULT /dmo/i_travel_m.

read_travel =
  VALUE #( ( %key-travel_id = 1
            %control      = VALUE #( agency_id   = if_abap_behv=>mk-on
                                     customer_id = if_abap_behv=>mk-on ) )
          ).

op_tab = VALUE #( entity_name = '/DMO/I_TRAVEL_M'
                  ( op       = if_abap_behv=>op-r-read
                    instances = REF #( read_travel )
                    results  = REF #( read_travel_result ) ) ).

READ ENTITIES OPERATIONS op_tab FAILED FINAL(read_failed).
```

```

DATA read_travel          TYPE TABLE FOR READ IMPORT /dmo/i_travel_m.
DATA op_tab              TYPE abp_behv_retrievals_tab.
DATA read_travel_result  TYPE TABLE FOR READ RESULT /dmo/i_travel_m.

read_travel =
  VALUE #( ( %key-travel_id = 1
            %control       = VALUE #( agency_id   = if_abap_behv=>mk-on
                                       customer_id = if_abap_behv=>mk-on ) )
          ).

op_tab = VALUE #( entity_name = '/DMO/I_TRAVEL_M'
                  ( op       = if_abap_behv=>op-r-read
                    instances = REF #( read_travel )
                    results   = REF #( read_travel_result ) ) ).

READ ENTITIES OPERATIONS op_tab FAILED FINAL(read_failed).

out->write( op_tab[ 1 ]-results ).

```

```
DATA op_tab TYPE abp_behv_retrievals_tab.
```

```
READ ENTITIES OPERATIONS op_tab FAILED FINAL(read_failed).
```

```
out->write( op_tab[ 1 ]-results ).
```

Operations Tab

Col	Description
op	Operation R or N
entity_name	Entity Name
sub_name	Association, Action
instances	Entity Daten
full	z.B. alle Subentities
results	Ergebnisse
links	Links

EML Dynamic Read

- op_tab enthält alle Details
- read_failed enthält Fehler

```
DATA read_travel      TYPE TABLE FOR READ IMPORT /dmo/i_travel_m.  
DATA op_tab          TYPE abp_behv_retrievals_tab.  
DATA read_travel_result TYPE TABLE FOR READ RESULT /dmo/i_travel_m.
```

```
op_tab = VALUE #( entity_name = '/DMO/I_TRAVEL_M'  
                 ( op       = if_abap_behv=>op-r-read  
                   instances = REF #( read_travel )  
                   results   = REF #( read_travel_result ) ) ).
```

```
READ ENTITIES OPERATIONS op_tab FAILED FINAL(read_failed).
```

```
out->write( op_tab[ 1 ]-results ).
```

Operations Tab

Col	Description
entity_name	Entity Name
operation	R für Read
instances	Entity Daten
results	Ergebnisse

```

DATA read_travel      TYPE TABLE FOR READ IMPORT /dmo/i_travel_m.
DATA op_tab           TYPE abp_behv_retrievals_tab.
DATA read_travel_result TYPE TABLE FOR READ RESULT /dmo/i_travel_m.

read_travel =
  VALUE #( ( %key-travel_id = 1
            %control      = VALUE #( agency_id   = if_abap_behv=>mk-on
                                     customer_id = if_abap_behv=>mk-on ) )
          ).

op_tab = VALUE #( entity_name = '/DMO/I_TRAVEL_M'
                  ( op       = if_abap_behv=>op-r-read
                    instances = REF #( read_travel )
                    results   = REF #( read_travel_result ) ) ).

READ ENTITIES OPERATIONS op_tab FAILED FINAL(read_failed).

out->write( op_tab[ 1 ]-results ).

```

Entity Read Daten

- %key – Keyfelder
- %control – Datenfelder

```

DATA read_travel      TYPE TABLE FOR READ IMPORT /dmo/i_travel_m.
DATA op_tab           TYPE abp_behv_retrievals_tab.
DATA read_travel_result TYPE TABLE FOR READ RESULT /dmo/i_travel_m.

read_travel =
  VALUE #( ( %key-travel_id = 1
            %control       = VALUE #( agency_id   = if_abap_behv=>mk-on
                                       customer_id = if_abap_behv=>mk-on ) )
          ).

op_tab = VALUE #( entity_name = '/DMO/I_TRAVEL_M'
                  ( op       = if_abap_behv=>op-r-read
                    instances = REF #( read_travel )
                    results   = REF #( read_travel_result ) ) ).

READ ENTITIES OPERATIONS op_tab FAILED FINAL(read_failed).

out->write( op_tab[ 1 ]-results ).

```

Demo

- **MODIFY ENTITIES OPERATIONS**

- Dynamische Version von MODIFY ENTITIES
 - Erstellen, Ändern, Löschen, Aktionen
- Mehrere Modify-Operationen in einem Statement
 - Internen Tabelle vom Typ ABP_BEHV_CHANGES_TAB

Demo

- **COMMIT ENTITIES**

- Triggert die RAP save sequence
 - Tabelle: abp_entity_name_tab

```
FINAL(dyn_tab) = VALUE abp_entity_name_tab( ( conv #( '/DMO/I_TRAVEL_M' ) ) ).  
COMMIT ENTITIES RESPONSES of dyn_tab  
  REPORTED FINAL(commit_reported)  
  FAILED   FINAL(commit_failed).
```


• GET PERMISSOINS

- Infos zu Berechtigungen zu Operationen/Felder
- Table: ABP_BEHV_PERMISSIONS_TAB
- Global / Instanz
 - Berechtigungen
 - Feature Controls

```
DATA perm_key_root TYPE TABLE FOR PERMISSIONS KEY /dmo/i_travel_m.
DATA perm_req_root TYPE STRUCTURE FOR PERMISSIONS REQUEST /dmo/i_travel_m.
DATA perm_res_root TYPE STRUCTURE FOR PERMISSIONS RESULT /dmo/i_travel_m.
DATA perm_tab      TYPE abp_behv_permissions_tab.

perm_key_root = VALUE #( ( %key-travel_id = 13 ) ).

perm_req_root = VALUE #( %field-total_price = if_abap_behv=>mk-on
                          %field-created_at  = if_abap_behv=>mk-on
                          %field-description = if_abap_behv=>mk-on ).

perm_res_root = VALUE #( ).

perm_tab = VALUE #( ( entity_name = 'DMO/I_TRAVEL_M'
                      instances  = REF data( perm_key_root )
                      request    = REF data( perm_req_root )
                      results    = REF data( perm_res_root ) ) ).

GET PERMISSIONS ONLY INSTANCE FEATURES OPERATIONS perm_tab
FAILED FINAL(permissions_failed)
REPORTED FINAL(permissions_reported).
```

• SET LOCKS

- Interne Tabelle vom Typ ABP_BEHV_LOCKS_TAB
 - entity_name
 - instances – TYPE TABLE FOR KEY OF ...

```
DATA instance TYPE TABLE FOR KEY OF /dmo/i_travel_m.  
  
instance = VALUE #( ( travel_id = 1 ) ).  
  
FINAL(lock) = VALUE abp_behv_locks_tab(  
    ( entity_name = '/DMO/I_TRAVEL_M'  
      instances = REF #( instance ) ) ).  
  
SET LOCKS lock FAILED FINAL(lock_failed)  
            REPORTED FINAL(lock_reported).
```

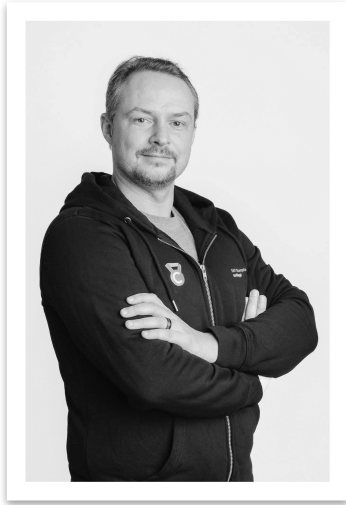
- **Informationen**

- Links

- [ABAP Keyword Dokumentation](#)
 - [ABAP for RAP \(abap-cheat-sheets\)](#)
 - [RAP EML Dynamic Operations: Dynamic EML Requests](#)

Type Mapping for RAP

Danke!



@SoSchlegel87



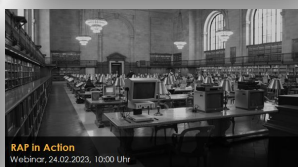
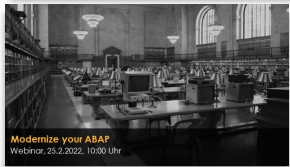
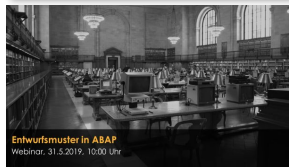
johann.foessleitner@cadaxo.com

@foessleitnerj



dominik.bigl@cadaxo.com

@DomiBigSAP



<http://www.cadaxo.com/blog/>

Unsere vergangenen Webinare

dynamische RAP Entwicklung