

June 6<sup>th</sup>, 2024

abap  
conf2024

# Keynote

```
DATA(ls_struct) = ...  
ASSERT ls_struct IS NOT INITIAL.  
get_sadl( IMPORTING es_sadl = DATA(ls_sadl) ).  
DELETE ls_sadl-attributes USING KEY structure_id WHERE structure_id = ls_struct-id.  
  
LOOP AT it_property_binding ASSIGNING <ls_property_binding> WHERE property_binding IS NOT INITIAL.  
  ls_attribute = VALUE #( ( id = |{ ls_struct-id }-{ ls_struct-id }-data | )  
    name = <ls_property_binding>-alias_text  
    binding = <ls_property_binding>-property_binding  
    structure_id = ls_struct-id ).  
  
  IF <ls_property_binding>-is_key = abap_true.  
    ls_attribute-is_key = if_sadl_type-not_is_key_true.  
  ENDIF.  
  INSERT ls_attribute INTO TABLE ls_sadl-attributes.  
ENDLOOP.
```

# Decisions

```
DATA(ls_structure) = get_structured( ls_getting_get ).
ASSERT ls_structure IS NOT INITIAL.
get_sadl( IMPORTING es_sadl = DATA(ls_sadl) ).
DELETE ls_sadl-attributes USING KEY structure_id WHERE structure_id = ls_structure-id.

LOOP AT it_property_binding ASSIGNING <ls_property_binding> WHERE property_binding IS NOT INITIAL.
  ls_attribute = VALUE #( |<ls_structure-name> |<ls_property_binding-name> |<ls_property_binding-property_binding> |<ls_structure-id> | ).
  IF <ls_property_binding>-is_key = abap_true.
    ls_attribute-is_key = if_sadl_types-not_is_key_true.
  ENDIF.
  INSERT ls_attribute INTO TABLE ls_sadl-attributes.
ENDLOOP.
```

```
DATA(ls_structure) = get_structured( ls_getting_get ).  
ASSERT ls_structure IS NOT INITIAL.  
get_sadl( IMPORTING es_sadl = DATA(ls_sadl) ).  
DELETE ls_sadl-attributes USING KEY structure_id WHERE structure_id = ls_structure-id.  
  
LOOP AT it_property_binding ASSIGNING <ls_property_binding> WHERE property_binding IS NOT INITIAL.  
  ls_attribute = VALUE #( ( id      = (| ls_structure-name |) sy-tabix )  
    ( name    = <ls_property_binding>-name )  
    ( binding = <ls_property_binding>-property_binding )  
    ( structure_id = ls_structure-id ).  
  
  IF <ls_property_binding>-is_key = abap_true.  
    ls_attribute-is_key = if_sadl_types-not_is_key_true.  
  ENDIF.  
  INSERT ls_attribute INTO TABLE ls_sadl-attributes.  
ENDLOOP.
```

```
DATA(ls_structure) = get_structured( ls_getting_get ).
ASSERT ls_structure IS NOT INITIAL.
get_sadl( IMPORTING es_sadl = DATA(ls_sadl) ).
DELETE ls_sadl-attributes USING KEY structure_id WHERE structure_id = ls_structure-id.

LOOP AT it_property_binding ASSIGNING <ls_property_binding> WHERE property_binding IS NOT INITIAL.
  ls_attribute = VALUE #(
    id           = <ls_structure-id> |<ls_property_binding-id>
    name        = <ls_property_binding-name>
    binding     = <ls_property_binding-property_binding>
    structure_id = ls_structure-id ).
  IF <ls_property_binding>-is_key = abap_true.
    ls_attribute-is_key = if_sadl_types-not_is_key_true.
  ENDIF.
  INSERT ls_attribute INTO TABLE ls_sadl-attributes.
ENDLOOP.
```

2017

2018

Tobias → Thomas → Sören

2019

Idea ← Föß ← Domi

2020

```
DATA(ls_structure) = get_structured( is_getting_get ).
ASSERT 1 structure IS NOT INITIAL.
get_sadl IMPORTING es_sadl = DATA(ls_sadl ).
DELETE 1 sadl-attributes USING KEY structure_id WHERE structure_id = ls_structure-id.

LOOP AT property_binding ASSIGNING <ls_property_binding> WHERE property_binding IS NOT INITIAL.
  ls_attribute = VALUE #( is_
    name = <ls_property_binding->name.
    binding = <ls_property_binding->property_binding.
    structure_id = ls_structure-id ).
  IF <ls_property_binding>-is_key = abap_true.
    ls_attribute-is_key = if_sadl_type-not_is_key_true.
  ENDIF.
  INSERT ls_attribute INTO TABLE ls_sadl-attributes.
ENDLOOP.
```

2017

2018

Tobias → Thomas → Sören

2019

Idea ← Föß ← Domi

ABAPConf: 09. December 2021

```
DATA(ls_structure) = get_structured( is_getting ).
ASSERT 1 <structure IS NOT INITIAL.
get_sadl IMPORTING es_sadl = DATA(ls_sadl ).
DELETE 1 <sadl-attributes USING KEY structure IF WHERE structure IS = ls_structure.

LOOP AT <property_binding ASSIGNED ls_property_binding WHERE property_binding IS NOT INITIAL.
  ls_attribute = VALUE #(
    name = ls_property_binding->name
    binding = ls_property_binding->binding
    structure = ls_structure
  ).
  IF <ls_property_binding>-is_key = <bool true>.
    ls_attribute-is_key = if_sadl_type-not_is_key_true.
  ENDIF.
  INSERT ls_attribute INTO TABLE ls_sadl-attributes.
ENDLOOP.
```

2017

2018

Sascha

Nick

André

**Tobias** → **Thomas** → **Sören**

Oliver

David

Ingo

2019

2020

**Idea** ← **Föß** ← **Domi**

Karl

**ABAPConf: 09. December 2021**

```
DATA(ls_structure) = get_structured( ls_sadl ).
ASSERT 1 = structure IS NOT INITIAL.
get_sadl IMPORTING es_sadl = DATA(ls_sadl ).
DELETE 1 = adl-attributes USING KEY structure IF WHERE structure IS = ls_structure.

LOOP AT property_binding ASSIGNING ls_property_binding WHERE property_binding IS NOT INITIAL.
  ls_attribute = VALUE #( |<ls_property_binding>|
    name = ls_property_binding->name
    binding = ls_property_binding->binding
    structure = ls_property_binding->structure ).
  IF <ls_property_binding>-is_key = true.
    ls_attribute-is_key = if_sadl_type-not_is_key_true.
  ENDIF.
  INSERT ls_attribute INTO TABLE ls_sadl-attributes.
ENDLOOP.
```



■ NO ■

marketing.  
titles.

hello world.  
buzzword technology.

■ Yes ■

Technology that is important.  
Experience from real projects.  
Decide the roadmap.

```
DATA(ls_structure) = get_structured_ls_getting_started.  
ASSERT ls_structure IS NOT INITIAL.  
get_sadl( IMPORTING es_sadl = DATA(ls_sadl) ).  
DELETE ls_sadl-attributes USING es_sadl-attributes.  
  
LOOP AT it_property_binding ASSIGNING ls_property_binding WHERE property = 'PROPERTY'.  
  ls_attribute = VALUE #( IS_KEY = IF ls_structure->is_key THEN 'X' ELSE '' ).  
  name = 'PROPERTY_' & ls_property_binding-property.  
  bi = 'PROPERTY_' & ls_property_binding-property.  
  st = 'PROPERTY_' & ls_property_binding-property.  
  
  IF <ls_property_binding>-is_key = 'X'.  
    ls_attribute-is_key = if_sadl_type-is_key.  
  ENDIF.  
  INSERT ls_attribute INTO TABLE ls_sadl-attributes.  
ENDLOOP.
```



# Experience

```
DATA(ls_structure) = get_structured( <ls_entity> );
ASSERT ls_structure IS NOT INITIAL.
get_sadl( IMPORTING es_sadl = DATA(ls_sadl) ).
DELETE ls_sadl-attributes USING KEY structure_id WHERE structure_id = ls_structure-id.

LOOP AT it_property_binding ASSIGNING <ls_property_binding> WHERE property_binding IS NOT INITIAL.
  ls_attribute = VALUE #( ( <ls_structure> name ( <ls_data> ) )
    name = <ls_property_binding>-name_path
    binding = <ls_property_binding>-property_binding
    structure_id = ls_structure-id ).
  IF <ls_property_binding>-is_key = abap_true.
    ls_attribute-is_key = if_sadl_type-not_is_key_true.
  ENDIF.
  INSERT ls_attribute INTO TABLE ls_sadl-attributes.
ENDLOOP.
```



A conference with content WE like to see

Focus on topics WE have to work with

Content by DEVELOPERS that work with this DAILY

Technology that is important NOW

```
DATA(ls_structure) = get_structure( ... )
ASSERT ...
get_sadl( IMPORTING ls_sadl = ... )
DELETE ls_sadl-attributes USING KEY structure ...

LOOP AT it_property_binding ASSIGNING <ls_property_binding> WHERE property_binding ...
  ls_attribute = VALUE #( ... )
  name = <ls_property_binding>-name
  binding = <ls_property_binding>-binding
  structure = ...
  IF <ls_property_binding>-is_key = ...
    ls_attribute-is_key = if_sadl_type-is_key ...
  ENDIF.
  INSERT ls_attribute INTO TABLE ls_sadl-attributes
ENDLOOP.
```



# An event made by developers.

```
DATA(ls_structure) = get_structured( is_getting_get ).  
ASSERT ls_structure IS NOT INITIAL.  
get_sadl( IMPORTING es_sadl = DATA(ls_sadl) ).  
DELETE ls_sadl-attributes USING KEY structure_id WHERE structure_id = ls_structure-id.  
  
LOOP AT it_property_binding ASSIGNING <ls_property_binding> WHERE property_binding IS NOT INITIAL.  
  ls_attribute = VALUE #( is_ = |{ ls_structure-name }| <ls_data> |{>  
    name      = <ls_property_binding-name> <ls_data>  
    binding   = <ls_property_binding-property_binding>  
    structure_id = ls_structure-id ).  
  
  IF <ls_property_binding>-is_key = abap_true.  
    ls_attribute-is_key = if_sadl_types-not_is_key_true.  
  ENDIF.  
  INSERT ls_attribute INTO TABLE ls_sadl-attributes.  
ENDLOOP.
```



# An event made by developers. For developers.

```
DATA(ls_structure) = get_structured( is_getting_get ).  
ASSERT ls_structure IS NOT INITIAL.  
get_sadl( IMPORTING es_sadl = DATA(ls_sadl) ).  
DELETE ls_sadl-attributes USING KEY structure_id WHERE structure_id = ls_structure-id.  
  
LOOP AT it_property_binding ASSIGNING <ls_property_binding> WHERE property_binding IS NOT INITIAL.  
  ls_attribute = VALUE #( is_ = |{ ls_structure-name } { ls_property_binding-name }|  
    name = <ls_property_binding-name>_field  
    binding = <ls_property_binding-property_binding>  
    structure_id = ls_structure-id ).  
  
  IF <ls_property_binding>-is_key = abap_true.  
    ls_attribute-is_key = if_sadl_types-not_is_key_true.  
  ENDIF.  
  INSERT ls_attribute INTO TABLE ls_sadl-attributes.  
ENDLOOP.
```



# An event made by developers. For developers.

```
DATA(ls_structure) = get_structured( is_getting_get ).
ASSERT ls_structure IS NOT INITIAL.
get_sadl( IMPORTING es_sadl = DATA(ls_sadl) ).
DELETE ls_sadl-attributes USING KEY structure_id WHERE structure_id = ls_structure-id.

LOOP AT it_property_binding ASSIGNING <ls_property_binding> WHERE property_binding IS NOT INITIAL.
  ls_attribute = VALUE #( is_
    = |{ ls_structure-name }| <ls_property_binding>|
    name = <ls_property_binding>-name_field.
    binding = <ls_property_binding>-property_binding.
    structure_id = ls_structure-id ).
  IF <ls_property_binding>-is_key = abap_true.
    ls_attribute-is_key = if_sadl_types-not_is_key_true.
  ENDIF.
  INSERT ls_attribute INTO TABLE ls_sadl-attributes.
ENDLOOP.
```

0 0 0 0

```
DATA(ls_structure) = get_structured(ls_property_binding);
ASSERT ls_structure IS NOT INITIAL;
get_sadl( IMPORTING es_sadl = DATA(ls_sadl) );
DELETE ls_sadl-attributes USING KEY structure_id WHERE structure_id = ls_structure-id;

LOOP AT it_property_binding ASSIGNING ls_property_binding WHERE property_binding IS NOT INITIAL AND NOT DELETED;
  ls_attribute = VALUE #( id = |{ls_structure-name}| {ls-property-binding-id} |
    name = <ls_property_binding-name> |
    binding = <ls_property_binding-property_binding> |
    structure_id = ls_structure-id );

  IF <ls_property_binding>-is_key = abap_true.
    ls_attribute-is_key = if_sadl_types-not_is_key_true.
  ENDIF.
  INSERT ls_attribute INTO TABLE ls_sadl-attributes.
ENDLOOP.
```

6 8 1 5

```
DATA(ls_structure) = get_structured(ls_property_binding);
ASSERT ls_structure IS NOT INITIAL;
get_sadl( IMPORTING es_sadl = DATA(ls_sadl) );
DELETE ls_sadl-attributes USING KEY structure_id WHERE structure_id = ls_structure-id;

LOOP AT it_property_binding ASSIGNING ls_property_binding WHERE property_binding IS NOT INITIAL;
  ls_attribute = VALUE #(
    id      = |{ls_structure-name}| {ls-property-binding-id}
    name    = <ls_property_binding-name>
    binding = <ls_property_binding-property_binding>
    structure_id = ls_structure-id );

  IF <ls_property_binding>-is_key = abap_true.
    ls_attribute-is_key = if_sadl_types-not_is_key_true.
  ENDIF.
  INSERT ls_attribute INTO TABLE ls_sadl-attributes.
ENDLOOP.
```

4 2 1 5

```
DATA(ls_structure) = get_structured(ls_property_binding);
ASSERT ls_structure IS NOT INITIAL;
get_sadl( IMPORTING ee_sadl = DATA(ls_sadl) );
DELETE ls_sadl-attributes USING KEY structure_id WHERE structure_id = ls_structure-id;

LOOP AT it_property_binding ASSIGNING ls_property_binding WHERE property_binding IS NOT INITIAL AND NOT DELETED;
  ls_attribute = VALUE #(
    id = |{ls_structure-name}|{ls-property-binding-id}|
    name = <ls_property_binding-name>_field
    binding = <ls_property_binding-property_binding>
    structure_id = ls_structure-id );

  IF <ls_property_binding>-is_key = abap_true.
    ls_attribute-is_key = if_sadl_types-not_is_key_true.
  ENDIF.
  INSERT ls_attribute INTO TABLE ls_sadl-attributes.
ENDLOOP.
```



4 2 4 8

```
DATA(ls_structure) = get_structured(ls_property_binding);
ASSERT ls_structure IS NOT INITIAL;
get_sadl( IMPORTING ee_sadl = DATA(ls_sadl) );
DELETE ls_sadl-attributes USING KEY structure_id WHERE structure_id = ls_structure-id;

LOOP AT it_property_binding ASSIGNING ls_property_binding WHERE property_binding IS NOT INITIAL AND NOT DELETED;
  ls_attribute = VALUE #( id = |{ls_structure-name}| {ls-property-binding-id} |
    name = <ls_property_binding-name> |
    binding = <ls_property_binding-property_binding> |
    structure_id = ls_structure-id );

  IF <ls_property_binding>-is_key = abap_true.
    ls_attribute-is_key = if_sadl_types-not_is_key_true.
  ENDIF.
  INSERT ls_attribute INTO TABLE ls_sadl-attributes.
ENDLOOP.
```

6 4 0 0

```
DATA(ls_structure) = get_structured(ls_property_name);
ASSERT ls_structure IS NOT INITIAL;
get_sadl( IMPORTING es_sadl = DATA(ls_sadl) );
DELETE ls_sadl-attributes USING KEY structure_id WHERE structure_id = ls_structure-id;

LOOP AT it_property_binding ASSIGNING ls_property_binding WHERE property_binding IS NOT INITIAL AND NOT DELETED;
  ls_attribute = VALUE #( id = |{ls_structure-name}| {ls-property-binding-id} |
    name = <ls_property_binding-name> |
    binding = <ls_property_binding-property_binding> |
    structure_id = ls_structure-id );

  IF <ls_property_binding>-is_key = abap_true.
    ls_attribute-is_key = if_sadl_types-not_is_key_true.
  ENDIF.
  INSERT ls_attribute INTO TABLE ls_sadl-attributes.
ENDLOOP.
```

# Registrations

6 4 0 0 +

1800+ subscriptions

1.000+ live viewers

YouTube

73.000+ views

29.000+ hours of content

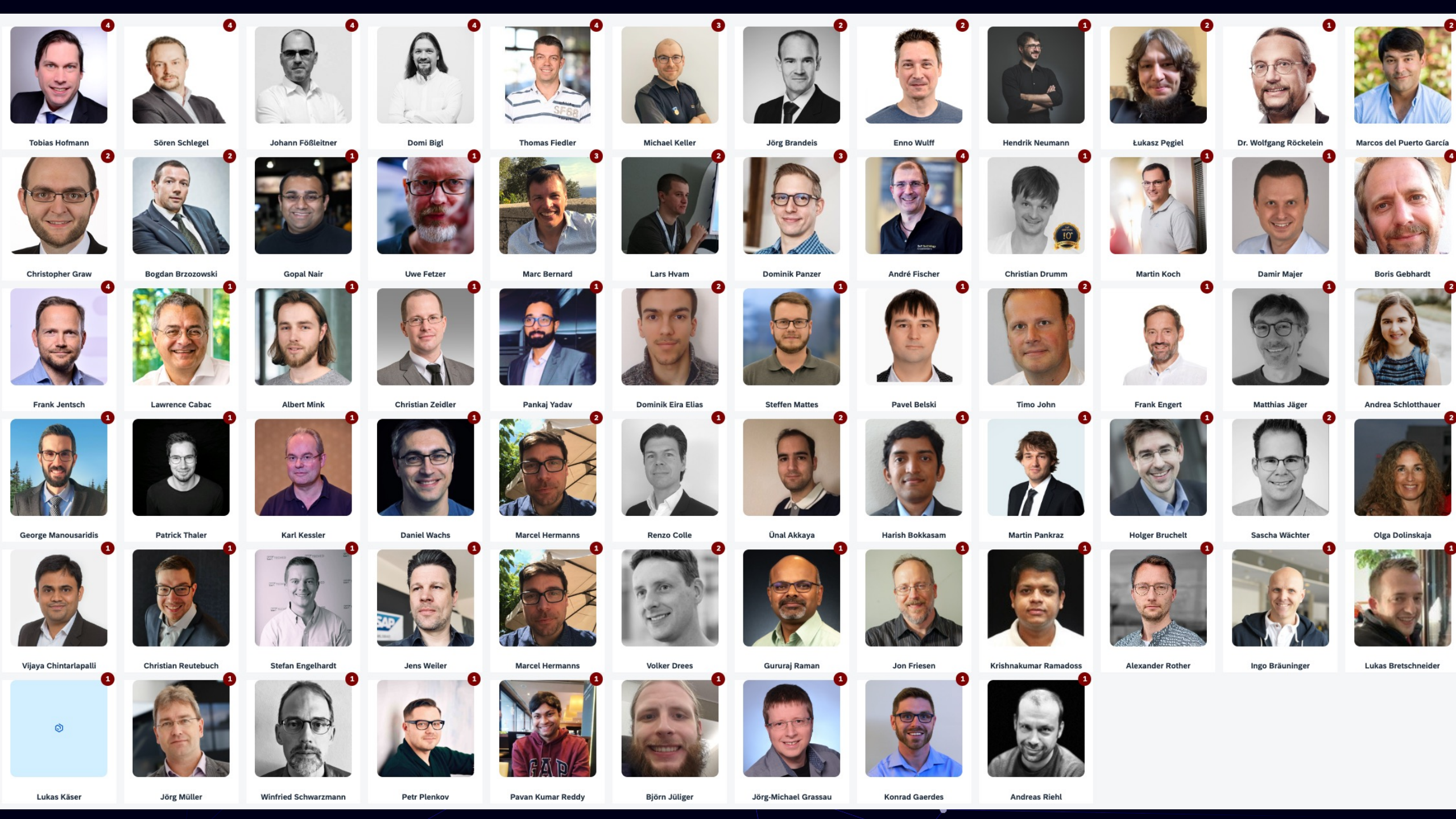
```
DATA(ls_structure) = get_structs
INSERT INTO TABLE ls_sadl IS NOT INITIAL
get_structs(ls_sadl) = DATA(ls_sadl)
DELETE ls_sadl-attributes USING KEY structure_id WHERE structure_id = ls_structure-id

LOOP AT it_property_binding ASSIGNING ls_property_binding
  ls_attribute = VALUE #(
    name = <ls_property_binding-property_name>
    binding = <ls_property_binding-property_binding>
    structure_id = ls_structure-id )
  IF <ls_property_binding>-is_key = abap_true.
    ls_attribute-is_key = if_sadl_type-not_is_key_true.
  ENDIF.
  INSERT ls_attribute INTO TABLE ls_sadl-attributes
ENDLOOP.
```



# ABAPConf LEGENDS

```
DATA(ls_structure) = get_structured( ls_getting_get ).  
ASSERT ls_structure IS NOT INITIAL.  
get_sadl( IMPORTING es_sadl = DATA(ls_sadl) ).  
DELETE ls_sadl-attributes USING KEY structure_id WHERE structure_id = ls_structure-id.  
  
LOOP AT it_property_binding ASSIGNING <ls_property_binding> WHERE property_binding IS NOT INITIAL.  
  ls_attribute = VALUE #( ( id = |{ ls_structure-id }| <ls_property_binding> ) )  
  name = <ls_property_binding>-name.  
  binding = <ls_property_binding>-property_binding.  
  structure_id = ls_structure-id ).  
  
IF <ls_property_binding>-is_key = abap_true.  
  ls_attribute-is_key = if_sadl_types-not_is_key_true.  
ENDIF.  
INSERT ls_attribute INTO TABLE ls_sadl-attributes.  
ENDLOOP.
```



Tobias Hofmann

Sören Schlegel

Johann Föbtleitner

Domi Bigl

Thomas Fiedler

Michael Keller

Jörg Brandeis

Enno Wulff

Hendrik Neumann

Łukasz Pegiel

Dr. Wolfgang Röckelein

Marcos del Puerto García

Christopher Graw

Bogdan Brzozowski

Gopal Nair

Uwe Fetzer

Marc Bernard

Lars Hvam

Dominik Panzer

André Fischer

Christian Drumm

Martin Koch

Damir Majer

Boris Gebhardt

Frank Jentsch

Lawrence Cabac

Albert Mink

Christian Zeidler

Pankaj Yadav

Dominik Eira Elias

Steffen Mattes

Pavel Belski

Timo John

Frank Engert

Matthias Jäger

Andrea Schlotthauer

George Manousaridis

Patrick Thaler

Karl Kessler

Daniel Wachs

Marcel Hermanns

Renzo Colle

Ünal Akkaya

Harish Bokkasam

Martin Pankraz

Holger Bruchelt

Sascha Wächter

Olga Dolinskaja

Lukas Käser

Jörg Müller

Winfried Schwarzmann

Petr Plenkov

Pavan Kumar Reddy

Björn Jüliger

Jörg-Michael Grassau

Konrad Gaerdes

Andreas Riehl

Lukas Käser

Jörg Müller

Winfried Schwarzmann

Petr Plenkov

Pavan Kumar Reddy

Björn Jüliger

Jörg-Michael Grassau

Konrad Gaerdes

Andreas Riehl

# ABAPConf Legends

<https://www.abapconf.org/legends>

```
DATA(ls_structure) = get_structured( get_structured ).  
ASSERT ls_structure IS NOT INITIAL.  
get_sadl( IMPORTING es_sadl = DATA(ls_sadl) ).  
DELETE ls_sadl-attributes USING KEY structure_id WHERE structure_id = ls_structure-id.  
  
LOOP AT it_property_binding ASSIGNING <ls_property_binding> WHERE property_binding IS NOT INITIAL.  
  ls_attribute = VALUE #( ls_attribute |>  
    name      = <ls_property_binding>-name |>  
    binding   = <ls_property_binding>-property_binding |>  
    structure_id = ls_structure-id ).  
  IF <ls_property_binding>-is_key = abap_true.  
    ls_attribute-is_key = if_sadl_types-not_is_key_true.  
  ENDIF.  
  INSERT ls_attribute INTO TABLE ls_sadl-attributes.  
ENDLOOP.
```







# SAP Development Tech Radar

2024.06-draft

## UI

### ADOPT

- 1. FPM
- 2. WDI5
- 3. Web Components

### USE

- 4. Fiori Launchpad
- 5. Fiori UX
- 6. MDK
- 7. Personas
- 8. UI Flexibility

### HOLD

- 9. Mobile Cards
- 10. SAP BTP SDK for Android
- 11. SAP BTP SDK for iOS
- 12. UI5 freestyle
- 13. WDA

### STOP

- 14. BSP
- 15. CRM-WebClient-UI
- 16. Dynpro
- 17. J2EE
- 18. WDJ

## Frameworks

### ADOPT

- 32. ABAP Cloud
- 33. RAP managed draft

### USE

- 34. ABAP Unit
- 35. BRF+
- 36. CAP JS
- 37. Clean ABAP
- 38. Custom Fields & Logic
- 39. Fiori Elements
- 40. OpenUI5
- 41. RAP managed
- 42. RAP unmanaged
- 43. SAPUI5

### HOLD

- 44. BOPF Fiori
- 45. CAP Java

### STOP

- 46. BOPF
- 47. Reports

## Technology

### ADOPT

- 19. BTP
- 20. OData v4
- 21. Workzone

### USE

- 22. CDS View Entity
- 23. FES embedded
- 24. OData v2

### HOLD

- 25. CDS Views
- 26. SOAP

### STOP

- 27. FES hub
- 28. Neo
- 29. Portal
- 30. XSA
- 31. XSC

## Tools

### ADOPT

- 48. Cloud ALM
- 49. gCTS
- 50. UI5 linter

### USE

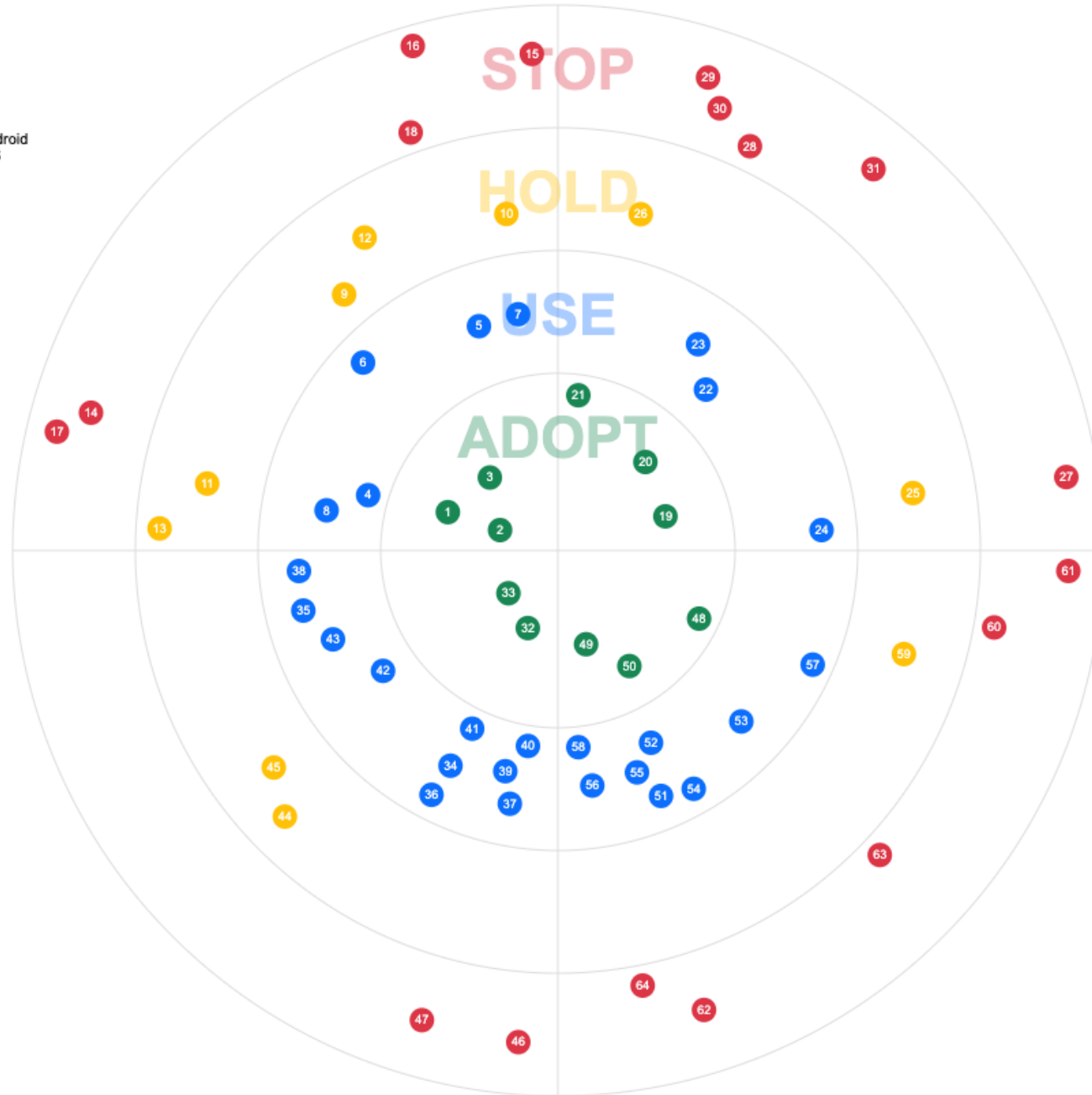
- 51. ABAP Cleaner
- 52. abapGit
- 53. ADT
- 54. ATC
- 55. BAS
- 56. eCATT
- 57. UI5 tooling
- 58. VS Code

### HOLD

- 59. SolMan

### STOP

- 60. Kapsel
- 61. SAP GUI
- 62. SE80
- 63. SEGW
- 64. WebIDE





# ABAPConf

LIVE | ONLINE | ON-SITE | FREE

```
DATA(ls_structure) = get_structured( ls_getting_get ).
ASSERT ls_structure IS NOT INITIAL.
get_sadl( IMPORTING es_sadl = DATA(ls_sadl) ).
DELETE ls_sadl-attributes USING KEY structure_id WHERE structure_id = ls_structure-id.

LOOP AT it_property_binding ASSIGNING <ls_property_binding> WHERE property_binding IS NOT INITIAL.
  ls_attribute = VALUE #( |<ls_structure-name> |<ls_structure-id> |<ls_property_binding-name> |<ls_property_binding-property_binding> |<ls_structure-id> | ).
  IF <ls_property_binding>-is_key = abap_true.
    ls_attribute-is_key = if_sadl_types-not_is_key_true.
  ENDIF.
  INSERT ls_attribute INTO TABLE ls_sadl-attributes.
ENDLOOP.
```