Problemstellung

In SAP Fiori ist es wichtig, sicherzustellen, dass Benutzer nur die Berechtigungen haben, die sie für ihre Aufgaben benötigen, insbesondere wenn es um Änderungen an Daten geht. Eine Möglichkeit, dies zu überprüfen, ist die Verwendung der "**What You See Is What You Get**" (WYSIWYG)-Methodik. Sie ermöglicht es, dass Benutzer in den Apps nur die Aktionen sehen können, für die sie auch berechtigt sind.

Nach unserem Kenntnisstand ist eine derartige Anforderung nicht über die Boardmittel von SAP Fiori umsetzbar. Ein Blick in die UI5 Demo Kit Dokumentation offenbart folgendes: *Moreover, common security mechanisms, which are usually taken for granted, like user authentication, session handling, authorization handling, or encryption are not part of SAPUI5 and need to be handled by the server-side framework and/or custom code of the application.*

Leseberechtigungen auf Daten können in den Fiori (zumeist) unterliegenden CDS-Views (Core Data Services) durch CDS-DCL-Objekte (Data Control Language) problemlos und elegant gesteuert werden. Doch was ist, wenn verschiedene Typen von Anwendern mit verschiedenen Arten von Berechtigungen auf ein und denselben Datensatz zugreifen wollen?



"grant update" und "grant delete" sind in einer DCL nicht vorgesehen

Was sagt SAP zu diesem Problem?

SAP sieht derartige Funktionalität in Fiori-Apps nicht vor. Folgende Alternativ-Lösungen sind möglich:

- A) Jedem Typ von Anwender wird eine eigene Applikation zur Verfügung gestellt
- B) Implementierung von kundeneigenen Lösungen
- C) Verwendung von separaten CDS-Views, welche auf die jeweiligen Änderungsberechtigungen per DCL eingeschränkt wurden

Mit der Lehrmeinung A wollen wir uns an dieser Stelle nicht zufriedengeben. Dieser Blogartikel greift die Variante B auf und demonstriert die Implementierung eines einfachen Web-Service für die Durchführung von PFCG-Berechtigungsprüfungen im UI5-Frontend. Auf Variante C werden wir ggf. in einem zukünftigen Artikel eingehen. An dieser Stelle sei ausdrücklich erwähnt, dass wir uns gern belehren lassen, sollte eine bessere Lösung für die beschriebene Problematik existieren oder mit neuen Releases auf uns warten.

Lösung über einen Function Import

Function Imports in SAP sind spezielle Funktionen in OData-Services, die komplexe Geschäftslogik ausführen. Sie ermöglichen das Abrufen, Bearbeiten oder Löschen von Daten durch definierte Operationen, die über Standard-CRUD-Operationen hinausgehen. Auch wenn es in modernen Releases andere Möglichkeiten gibt, ähnliche Funktionen bereitzustellen, sollte die dargestellte Methode in allen gängigen SAP-Releases ohne zusätzliche Voraussetzungen umsetzbar sein. Wir gehen einzig davon aus, dass im darunterliegenden ABAP-Stack die SAP-Gateway-Komponente zur Verfügung steht.

1. Anlage SEGW Projekt mit Function Import

Zunächst legen wir in der Transaktion SEGW ein neues Gateway Service Builder-Projekt ZUI5_UTILITIES an. Die Einstellungen belassen wir auf Standard.

≡	<u>P</u> roject	<u>E</u> dit	<u>G</u> oto	Extr <u>a</u> s	System	ı <u>H</u> elp		>	/IWBEP/SAPLFC	G_SI	BUI_SB_MAIN	►	₽	_ [⊐ ×
<	SAP					SAP Ga	ateway S	ervice	Builder						
						S4H(1)/1	00 Create P	Project				×			Exit
	<u>r</u>												_		
				Pro	oject: * Z	UI5_UTILI	TIES								
	>			Descrip	otion: * L	JI5 Utilities									
	>														
	Attri	butes													
				Project T	ype: * S	ervice with	SAP Annotati	ions			~				
			Gene	ration Stra	ategy: S	andard					~				
	Obje	ect Dir	ectory l	Entry											
				Pack	age: *	TMP			Ī	D					
			Persor	n Respons	ible: * W	INKLER			u						
										J	Local Object	×			
						$\langle \rangle$					Local Object	~			< >
															Enter

Anlage SEGW Project

Anschließend wird ein Function Import erstellt und unter "Function Import Parameters" einfache Parameter mit dem Datentyp Edm.String eingetragen. Wir verwenden für diese Demonstration sowohl die einfache Kombination aus genau einem Objekt, eines Feldes und einem Wert, als auch den Import-Parameter autharray, welcher für komplexere Abfragen im JSON-Format gedacht ist.



Import Parameter

Eine komplexe Berechtigungsprüfung könnte der folgenden JSON-Notation entsprechen. Auf die Implementierung einer derartigen generischen Prüfung werden wir in einem Folgeartikel eingehen.

Für die Rückgabe des Function Imports wird eine Rückgabestruktur benötigt. Diese legen wir im Reiter "Entity-Types" an und erstellen gleichzeitig ein Entity-Set, auch wenn es aktuell nicht notwendig ist. Für die Rückgabestruktur verwenden wir in diesem möglichst einfachen Beispiel ZRETURN_MESSAGE mit der Eigenschaft message, welche als ebenfalls als Edm.String zurückgegeben wird.

≡	Project	Edit	Goto	Extr <u>a</u> s	System	<u>H</u> elp						>	/IWBEP/SA	PLFG_S	BUI_SB_	_main [• @ ∣	_ □	×
<	SAP	7						SAP Ga	ateway Ser	vice Builder	r								
				More \checkmark															Exit
	5		•			Function Impo	rt Parameters												
						R 🔊 🖗	ND⊕€		■ ■ 〜	(i) (i) (ii)									
	> 🚱					Name	EDM Core	e Ty Prec	. Scale Max	Unit Referenc	Label	La AB	AP Field	A Se	emantics				
	> 🚳					autharra	v Edm String	а 0	0 0			Т							
H	2 68 3 63							S4	H(1)/100 Crea	te Entity Type			×						
	> 🐼																		
	> 🚱							· (******											
	🗸 🖗 ZU	I5_UTIL	LITIES				Entity Type Nan	ne: • ZRET	URN_MESSAG	티									
	~ 🗗	Data Mo	del																
	[Entity	Types			Optiona	ıl												
H		Assoc	Sate			Cre	ate Related Entity	y Set											
	ا ۲ ~ ۲	Funct	ion Imp	orts			Entity Set Na	me: ZRET	URN_MESSAG	ESet									
		~ == AU	ТНСНЕ	CK															
		> 🗅	Functio	n Import P	arameters														
		Service	Implem	entation															
		Runtime	Artifact	5									✓ ×						
-		Service	Mainter	ance															
_																		_	
																		E	iter
Er	ntitv	Tvp	e fi	ir Ri	icka	abewer	te des F	unct	ion Im	ports									
≡	Project Ed	lit Goto	Extras	System H	elp				-					> /IWB	ep/saplfg	_SBUI_SB_I	MAIN 🕨 (r	□ ×
<	SAP							SAP	Gateway Serv	vice Builder									
			More 🗸																
	3 🖗 1		←⇒	Entity T	ypes														
n i) Ph Seni	ice Impleme	entation	🖬 🖻	1 🕺 🗋				invert 🗸 🖻 🖆										
	> 🗅 Runtir	me Artifacts		Na Na		ABAP Struct	ure Ba	ase Type	Abstr Label	La. Semantics	Thing Media Au	thor	ETag	Publi	ished	As Title Pro	per. Update	d	
	> 🔂 Servi	ice Mainten	ance	- [e	ALTONN_MES	and:													
	> 6	86 ,9753																	
	> (3) > (3)																		
	~ 🕞 ZUI5_U	TILITIES																	
	V 🔂 Ent	tity Types																	
		ZRETURN_	MESSAG																
		Navigatio	on Propert	te															
	C As	sociations																	
	v 🕤 Fu	nction Impo	orts																
		AUTHCHE	CK Import Pa	ira															
	> 🗅 Servi	ice Impleme	entation																
	🕒 Runtir	me Artifacts ice Mainten	ance																
-																			
																			Enter

Entity Type ZRETURN_MESSAGE

Selbstverständlich dient die einfache Gestaltung einer Rückgabe lediglich der Veranschaulichung. In praktischen Anwendungen sollten sinnvolle Rückgabeparameter mit einer Steuerung von HTTP-Status-Codes kombiniert werden.

Ξ	Project Edit Goto Extras Sys	tom Help	> //WBEP/SAPLFG_SBUI_SB_MAIN 🗈 🖉 💶 🗙
<	SAP	SAP Gateway Service Builder	
E			
	Image: Second	Properties Image: Im	A. Semantics
	L Data count		
	Data saved		Enter

Property message des Entity Types ZRETURN_MESSAGE

Ist der Entity Type definiert, kann dieser im Function Import hinterlegt werden. Bitte beachten, dass wir für diese Demonstration lediglich ZRETURN_MESSAGE und die Kardinalität 0..1 angegeben haben. Für praktische Implementierungen wird die "optionale" Kardinalität nicht empfohlen, worauf einen das SAP-System auch gern und häufig hinweisen wird.



Return-Parameter im Function Import

2. Generieren der ABAP-Klassen

Sind die Vorbereitungen erledigt, können wir den Gateway-Service generieren lassen. Dazu drücken wir das "BMW-Symbol" und bestätigen die Vorschläge der zu generierenden Klassen.

≡ S4H(1)/100 SAP Gateway Serv	ice Builder					/IW8EP/SAPLFG_	SBUI_SB_MAIN [8 @ _ ⊟ ×
< SAP			SAP Gateway Service Builder					
<u> </u>	*	S4H(1)/100) Model and Service Definition	×				
	Model Provider Clas	s Class Name: *	• ZCL_ZUIS-UTILITIES_MPC_EXT		d Type Action	n for Entity Ty Lat	el La	ab P
	Data Provider Class	Base Class Name:*	* ZCL_ZUI5_UTILITIES_MPC					
□ ∨ ♥ ZUI5_UTILITIES □ ∨ ♥ Data Model □ ∨ ♥ Entity Types □ ∨ ♥ Entity Types	Generate Classes	Class Name: * Base Class Name: *	* ZCL_ZUI5_UTILITIES_DPC_EXT * ZCL_ZUI5_UTILITIES_DPC					
Properties Properties Properties Properties Properties Properties Properties Properties Properties	perties Service Registration		(100 - 107) 1776 - 100					
	t Parameters	Model Version: " Technical Service Name:	* 1 * 2UI5_UTILITIES_SRV					
Construction Construction		Service Version	1					
				~ ×				
								Enter

ABAP-Klassen generieren

Unter Umständen wird an dieser Stelle ein Fehler geworfen, welcher den Anwender auffordert, die Klassen nochmals zu generieren. Wir tun genau dieses und sollten anschließend lediglich Warnmeldungen erhalten. Ist die Generierung abgeschlossen, navigieren wir zu den Runtime-Artifacts und springen in die ABAP Workbench der * DPC EXT Klasse.

≡	Project Edit Goto Extras System	Help	> /IWBEP/SAPLFG_SBUI_SB_MAIN	• • • - • ×
<	SAP	SAP Gateway Service Builder		
C				
۵		Function imports		
	> <u>6</u>	Return Type Kind Return Type Return Cardinality Return Entity Set HTTP Method Type A	ction for Entity Ty_ Label	Lab
븜		AUTHCHECK Entity Type V ZRETURN MESSA 0.1 V Not specified V		T
	V IS Data Model			
	Sentity Types			
	V III ZRETURN_MESSAGE			
	> 🛅 Properties			
	🛅 Navigation Properties			
	C Associations			
	Entity Sets			
	✓			
	V III AUTHCHECK			
	> D Function Import Parameters			
	> C Service Implementation			
	V C Runtime Artifacts			
	2CL_2UI5_UTILITIES_DPC			
븝	ZCL_ZUI5_UTILITIES_DPC_FXT			
븜		page		
		inga		
	S 711TS LITTI TTTES SDV GO	to ABAP Workbench		
	Service Maintenance De	tails		
				Enter

Absprung in die ABAP Workbench

In der Klasse /IWBEP/IF_MGW_APPL_SRV_RUNTIME redefinieren wir die Methode EXECUTE_ACTION, wie im folgenden Bild dargestellt.



Redefinition der Handler-Methode

Die Methode EXECUTE_ACTION führt, vereinfacht gesagt, die Aktion aus, die wir im Function Import als OData Service definiert haben. Über ABAP-Code ist es uns an dieser Stelle möglich, beliebige Logik zu implementieren. Der dargestellte ABAP-Code führt eine einfache ABAP-Berechtigungsprüfung mit genau einem Objekt, Feld und Wert durch und liefert den Return-Code als Message an den OData-Service zurück.



Einfache Berechtigungsprüfung

Anbei der Beispielcode. Bitte beachten, dass wir in einem der nächsten Teile auf den generischen Teil separat eingehen werden.

```
METHOD /iwbep/if mgw appl srv runtime~execute action.
    DATA ls parameter TYPE /iwbep/s_mgw_name_value_pair. " Structure
for parameter name-value pair
    DATA ls entity
                    TYPE zcl zui5 utilities mpc=>ts zreturn message.
" Structure for the return message
    DATA lt entity
                     TYPE zcl zui5 utilities mpc=>tt zreturn message.
н
 Table type for return messages
    DATA lv object TYPE c LENGTH 10. " Variable to hold the object
name (Authorization Object)
                                       " Variable to hold the field
    DATA lv field
                  TYPE c LENGTH 10.
name (Authorization Field)
    DATA lv value TYPE c LENGTH 20. " Variable to hold the field
```

```
value (Authorization Value)
    " Read the parameters from the input table
    READ TABLE it parameter INTO ls parameter WITH KEY name =
'object'.
    lv object = ls parameter-value.
    READ TABLE it parameter INTO ls parameter WITH KEY name = 'field'.
    lv field = ls parameter-value.
    READ TABLE it parameter INTO ls parameter WITH KEY name = 'value'.
    lv value = ls parameter-value.
    " Perform an authority check based on the provided object, field,
and value
    AUTHORITY-CHECK OBJECT lv object
    ID lv field FIELD lv value.
    ls entity-message = sy-subrc. " Store the result of the authority
check in the return message
    " Copy the result to the output reference
    copy data to ref( EXPORTING is data = ls entity
                      CHANGING cr data = er data ).
    er data->* = ls entity. " Final assignment to output data
  ENDMETHOD.
```

3. Registrieren des Services

Um den Service verwenden zu können, müssen wir ihn im ABAP-Gateway registrieren. Dazu navigieren wir in die Transaktion /IWFND/MAINT_SERVICE und klicken auf "Add Services". In dem sich geöffneten Untermenü wählen wir den System Alias "LOCAL" (Achtung: Abweichung möglich je nach Systemkonfiguration) und bestätigen die Auswahl. Anschließend müsste der erstellte Service auswählbar sein. Wir bestätigen die Auswahl und fügen den Service hinzu, ohne zusätzliche Einstellungen vorzunehmen.

	ALIENTON A 44 Colored Constant				
-			Add Selected Services		
			Add Selected Selvices		
	✓ 6∂ Get Services More ✓				
Filter					
	Syntem Alian: LOCAL		Co-Deployed		
1			ce	x	
	Service				
Sele	Technical Service Name:* Z	UI5_UTILITIES_SRV			
ଭ	Service Version: 1				
	Description: U				
	External Service Name: Z	2015_UTILITIES_SRV			
밑	Namespace:				
出	External Mapping ID:				
	External Data Source Type: 0				
붬	Model				
	Model Version: 1				
	modet version.				
H	Creation Information				
	Package Assignment: \$	TMP			
믬		Local Object			
H					
	ICF Node				
	Standard Mode	O None			
믬	Set Current Client as Default Client in ICF Node				
	OAuth enablement				
	Enable OAuth for Service				
믭					Ĵ
					nter

Registrierung des Services in /IWFND/MAINT_SERVICE

Ist der Service registriert, suchen wir ihn in der /IWFND/MAINT_SERVICE und öffnen im Kontext Menü unten den Link "Configure SICF-Service". Im SICF-Service öffnen wir die "GUI-Configuration" und tragen dort den Parameter ~CHECK_CSRF_TOKEN mit dem Wert 0 ein. Diese Einstellung deaktiviert die CSRF-Token Abfrage und ist notwendig, um den Service einfacher mit einem Tool wie Postman testen zu können.

Ξ Service <u>E</u> dit S <u>y</u> stem <u>H</u> elp				> SICF 🕨 🖻 🗌 _	- 🗆 ×
< SAP		S4H(5)/100 Maintain Service Parameters	×		
					Eva
¥ %	Parameter Name	Value			EXIL A
Path: /default_host/sap/op	~CHECK_CSRF_TOKEN		Ŷ		Ĭ
Service Name: S4H10000000967					
Lang.: English					
Description					
Description 1: ZUI5_UTILITI					
Description 2:					
Description 3:					
Service Data Logon Data H					
Service Options					
Web Service					
I ignore innerited settings					
Load Balancing:					
Session Timeout: 00,00,0					
Compression: Not spec					
Interactive Options					
G					
Support	\mathbf{O}	c)	\$		
		± ⊕ ⊝ [͡ᢒ]	×		
					0
				Store	Cancel

SICF CSRF-Prüfung deaktivieren

4. Testen des Services

Für den Test verwenden wir das Tool Postman, welches in der Basis-Version frei verfügbar heruntergeladen werden kann. Als Abfrage-Methode wählen wir GET und als Ziel-URL: <server-addresse>/sap/opu/odata/ZUI5_UTILITIES_SRV, welche im Normalfall gültig sein sollte. Falls dies nicht funktioniert, kann die korrekte Addresse über die /IWFND/MAINT_SERVICE oder SICF in Erfahrung gebracht werden (...).

⇒ ←	→ Ho	me Workspaces - API Network -	Q Search Postman	😕 Invite 🕸 🗘 🎯 Upgrade	· – 🗆	>
Å	6	Overview GET CUIS_UT	LITIES +	~ 🕅 N	o environment 🗸 🗸	E
Collections	सारे दा	UIS_UTILITIES / /sap/opu/odata/t	sap/ZUI5_UTILITIES_SRV/	Sav	share	E
P.	GET	✓ /sap/opu/odata/sap/2	ZUI5_UTILITIES_SRV/?object='F_BKPF_BUK'&field='ACTVT'&value='0.	2'	Send 🗸	G
4) History	Params Query P	Authorization Headers (10) Body Scripts Test - arams	s Settings		Cookies	4
00	0	Кеу	Value	Description	+++ Bulk Edit	1
8+	0	autharray	"{{"Object":"F_BKPF_BUK","Fields":{{"Field":"BUKRS","Value":"100	0		(
		object	'F_BKPF_BUK'			
	⊻	field	'ACTVT'			
		value	'02'			
		Key	Value	Description		
	Respon	50	<u>*</u>		v	
			° (c)			

Test-Setup einfache Prüfung

Als Query-Parameter legen wir für das einfache Szenario eine Prüfung auf F_BKPF_BUK mit der ACTVT 02 fest. Bitte beachten, dass alle Übergabeparameter mit dieser Vorgehensweise über einfache Anführungsstriche als String deklariert werden müssen.

\equiv \leftarrow	→ Hor	me Workspaces 🛩 API Network 🗸	Q Search Postman	🐥 Invite 🕸 🗘 🎯 Upgrade	• – D	>		
^	60	Overview GET • 🗍 ZUI5_UT	ILITIES × +	~ 国 N	o environment 🗸 🗸	E		
Collections	लाई टा	II5_UTILITIES / 'sap/opu/odata/	sap/ZUI5_UTILITIES_SRV/	🖾 Sav	e 🗸 Share	1		
e.	GET	✓ /sap/opu/odata/sap/	/sap/opu/odata/sap/ZUI5_UTILITIES_SRV/?autharray='[["Object":"F_BKPF_BUK";"Fields":[["Field":"BUKRS";"Value":"1000"], ("Field":"ACTVT";"Value Send					
4) History	Params Query P	Authorization Headers (10) Body Scripts Test arams	is Settings		Cookies	4		
00		Key	Value	Description	+++ Bulk Edit	1		
0÷		autharray	'[{"Object":"F_BKPF_BUK","Fields":[{"Field":"BUKRS","Value":"1000.			(
	0	object	'F_BKPF_BUK'					
		field	ACTVT'					
	0	value	'02'					
		Key	Value	Description				
	Respons	ie			v			
			Glick Send to get a response					
O Online	Q. Find an	nd replace 🖾 Console		O Postbot Runner Start Proxy Cookies	🕫 Vault 🗻 Trash 🖾	A.		

Test-Setup komplexe Prüfung

Damit keine CSRF-Fehler auftreten, ist es außerdem wichtig den Header X-Requested-With mit einem beliebigen Wert z.B. "X" mitzusenden.

\equiv \leftarrow	ightarrow Home Workspaces $ ightarrow$ API Network $ ightarrow$	Q Search Postman	🙏 Invita 🕸 🗘 🎯 Upgrade 🗸 — 🗆 🗲
ĉ	C Overview GET	□ ZUIS_UTILITIES +	✓ IN No environment ✓ IN
0 Collections	👼 Zuis_utilities /	'sap/opu/odata/sap/ZUI5_UTILITIES_SRV/	🍘 Save 🐱 Share
D. Environments	GET ~ /	sap/opu/odata/sap/ZUI5_UTILITIES_SRV/?autharray='[{"Object":"F_BKPF_BU	UK";"Fields":{{"Field":"BUKRS";"Value::"1000"},("Field":"ACTVT";"Value
1) History	Params Authorization Headers (10) Body Headers Planders Plande	Scripts Tests Settings	Cookies </td
	Key	Value	Description Bulk Edit Presets
0+	X-Requested-With	×	G
	Key	Value	Description
	Response		
		Click Send to get a response	
E O Online	a Q Find and replace Console		🔞 Postbot 💽 Runner 🧬 Start Proxy 🚯 Cookies 🕫 Vault 🧻 Trash 🔛

X-Requested-With Header zur Umgehung der CSRF-Prüfung

Sind alle genannten Schritte korrekt durchgeführt worden, sollten wir jetzt in der Lage sein, einen Request abzufeuern, welcher im Backend eine einfache Berechtigungsprüfung dessen Return-Code an das Frontend weiterleitet.



Return Message als XML

Die Anwendung dieser Technik demonstrieren wir in Teil 2.

Über den Autor



Hendrik Winkler ist Partner der consiness und Lead Architekt für Identity und Access Management Lösungen. Er kann auf umfangreiche Expertise in SAP ABAP, GRC, Cloud-Technologien und SAP Identity Management zurückgreifen. Mit über zehn Jahren in der IT-Branche hat er sich auf die Entwicklung und Implementierung von komplexen IAM-Systemen spezialisiert, wobei er stets ein Auge auf Sicherheit, Benutzerfreundlichkeit und Compliance hat.

Der Artikel ist auch bei Linkedin erschienen:

https://www.linkedin.com/pulse/berechtigungen-zur-laufzeit-sap-fiori-apps-prüfen-hendrik-winkler-t52we/?trackingId=wDswEpGqSTKQNxEUIC%2BX7g%3D%3D