

Inhaltsverzeichnis

1. Classic Import Satzaufbau	2
2. Bfx.Abstract.Model.WebService.dll	6

Classic Import Satzaufbau

Diese Schnittstelle wird nicht mehr neu in Betrieb genommen, sondern nur noch für bestehende Installationen weitergeführt!

Der neue Standard ist folgende Schnittstelle: [Web-Service: Objekte anlegen und ändern](#)

Der Classic Import dient zum Import von Personalstammdaten im "alten" Format, welches in der Version 6.5 benutzt wurde.

Die Satzarten Z und M können ab der Version 7.8 nicht mehr importiert werden, die Funktionen wurden durch andere Schnittstellenformate abgelöst.

Inhaltsverzeichnis

1 Beschreibung Satzart S (Persönliche Stammdaten)	3
2 Beschreibung Satzart V (Stationszuteilungen, Versetzungen)	3
3 Beschreibung Satzart P (Parameterwerte für die Berechnung)	4
4 Vorgehensweise Import File	4

Beschreibung Satzart S (Persönliche Stammdaten)

Feldinhalt	Länge	Format	Beschreibung
Personalnummer	8	alphanumerisch	Linksbündig
Satzart	1	alphanumerisch	"S" = Stammsatz, muß mind 1x erstellt werden
Zuname	30	alphanumerisch	Zuname
Vorname	30	alphanumerisch	Vorname
Bereich	3	alphanumerisch	Dienstplanbereich (zB "PFL" für Pflege, "ARZ" für Ärzte oder Blank, wenn nur 1 Bereich verwendet wird)
Filler	10		Blanks
Eintrittsdatum	8	JJJJMMTT	Eintrittsdatum
Austrittsdatum	8	JJJJMMTT	Austrittsdatum
Station	5		Blanks
Filler	22		Blanks
Geburtsdatum	8	JJJJMMTT	Geburtsdatum
Geschlecht	1	alphanumerisch	"M" oder "W"
Berufsgruppe	5		Blanks
Magnetkartennummer	15	alphanumerisch	(Bar)Code der Magnetkarte für Zeitstempelung
Abteilungsleiterkennzeichen	1	alphanumerisch	"X" wenn Abteilungsleiter, sonst blank
Filler	19		Blanks
Schülerkennzeichen	1	alphanumerisch	"X" wenn Schüler, sonst leer
Kurzname	30	alphanumerisch	Kurzname (zB Schwesternname)
Abrechnungscode	5		Blanks
Umschlüsselungstabelle	5		Blanks

Beschreibung Satzart V (Stationszuteilungen, Versetzungen)

Feldinhalt	Länge	Format	Beschreibung
Personalnummer	8	alphanumerisch	Linksbündig
Satzart	1	alphanumerisch	"V" = Versetzung, dh Eintritt/Austritt bezieht sich auf Stations; ein "V"-Satz bewirkt, dass der unmittelbar vorangegangene Einsatz beendet wird
Zuname	30		Blanks
Vorname	30		Blanks

Bereich	3		Blanks
Filler	10		Blanks
Eintrittsdatum	8	JJJJMMTT	"Ab-Datum" auf der Station
Austrittsdatum	8	JJJJMMTT	"Bis-Datum" auf der Station
Station	5	alphanumerisch	linksbündig
Filler	22		Blanks
Geburtsdatum	8		Blanks
Geschlecht	1		Blanks
Berufsgruppe	5	alphanumerisch	Berufsgruppencode
Magnetkartennummer	15		Blanks
Abteilungsleiterkennzeichen	1		Blanks
Filler	19		Blanks
Schülerkennzeichen	1		Blanks
Kurzname	30		Blanks
Abrechnungscode	5	alphanumerisch	Abrechnungscode lt. Alex-Stammdaten
Umschlüsselungstabelle	5	alphanumerisch	Umschlüsselungstabelle lt. Alex-Stammdaten

Beschreibung Satzart P (Parameterwerte für die Berechnung)

Feldinhalt	Länge	Format	Beschreibung
Personalnummer	8	alphanumerisch	Linksbündig
Satzart	1	alphanumerisch	"P" = Personalparameter
Parameter	30	alphanumerisch	Parametercode lt. Alex-Stammdaten
Ab Datum	8	JJJJMMTT	Gültigkeitsdatum (Start)
Wert	10	Dezimal	xxxxxxx.xx (B Beschäftigungsgrad)
Text	100	alphanumerisch	bei Textparametern (zB Adresse)

Vorgehensweise Import File

In der Personalsuche Rechtsklick auf einen Mitarbeiter und "Personalstammdaten aus Datei importieren [classic-format]" auswählen:

Neuigkeiten rund um ALEX unser... Personal für Beschäftigt von/bis

Abgefragter Typ **Personal**

Ansicht **STAMM-PLANUNGSEINHEIT, BERUFSGRUPPE, Zuname, Vorname, Personalnummer, Beschäftigt**

Suchbegriff

von

bis

Beschäftigt Ja

Zuname Vorname

Stamm-Planungse
Berufsgruppe:

- Augustin
- Baumgartner
- Eder
- Harasser
- Huber
- Jakubcakova
- Keiler
- Poepperl
- Reich
- Reich
- Ritsch
- Schreder
- Turnusarzt
- Wimmer

Augustin Charlotte, 3902 bearbeiten

Personalstammdaten aus Datei importieren [classic-format]

- Stempel manuell erfassen
- Personalnummer auf Kartenummer kopieren
- Kartenummer konvertieren
- Stundennachweis als E-Mail versenden
- Jahresplan öffnen
- Selbstbedienung lesend anlegen
- Selbstbedienung anlegen
- Stundennachweis
- Jahr neu aufrollen
- Lohnartenexport anlegen
- Abwesenheitsexport anlegen
- Eigenschaft importieren
- Löschen umschalten
- Zeitkontostand für *Personal*
- Besetzungs-Report für ActionReportWatchingResource
- Benutzer für Berechtigung

Webservice - Objekte anlegen und ändern

Das **Webservice - Objekte anlegen und ändern** ist eine **Programmfunktion** zum Anlegen von **Objekten** eines **Objektyps** und zum Ändern von **Eigenschaften** eines **Objekts** über das HTTP-Protokoll.

Es kann dazu verwendet werden

- **Personalstammdaten zu importieren**
- Benutzerstammdaten zu importieren
- **Eigenschaften zu importieren**

Inhaltsverzeichnis

1 Ausgangssituation	7
1.1 Voraussetzungen	7
1.2 Formatierung der Anfrage	7
1.2.1 Schema für URL-Kodierung	7
1.2.2 Schema für JSON-File	7
1.2.3 objectType	8
1.2.4 indexQuery	8
1.2.5 matchString	8
1.2.6 importType	8
1.2.7 valueString	9
1.2.8 keyDate (optional)	9
1.2.9 toDate (optional)	9
1.2.10 keyString (optional)	9
1.3 Formatierung der Ausgabe	9
2 Beispiele	9
2.1 Beispiel: Ein Personal mit Standardinformationen anlegen	9
2.2 Beispiel: Neues Objekt anlegen	11
2.3 Beispiel: Eigenschaften in Personal ändern	11
2.3.1 Besonderheiten	16
2.3.1.1 Beschäftigungsausmaß	16
2.4 Beispiel: Anspruchszeile	16
2.5 Beispiel: Eigenschaften in Benutzer ändern	16
2.6 Beispiel 1: Email in Personal ändern	17
2.6.1 Ausgangssituation	17
2.6.2 Aufruf der Funktion	17
2.6.3 Ergebnis	18
2.6.4 Datenbank für Beispiel	18
2.7 Cheatsheet	18
3 Zeitbereich Korrektur bei Personal	20
3.1 Beispiel	20
4 Referenzen auf andere Objekte	20
5 File Import	21
6 Interaktiver Test von POST	21
7 Protokollierung	22
8 Fußnoten	22

Ausgangssituation

Voraussetzungen

Neben den Voraussetzungen für den Betrieb eines [Webservice](#) muss [das richtige Plugin](#) konfiguriert sein.

Das Webservice stellt dann folgende Zugriffspfade zur Verfügung:

- `/New` : Anlage eines neuen Objekts.
 - kann entfallen wenn der Parameter `Systemeinstellungen: Object Import Einstellungen: Eigenschaften-Import: Neu erstellen falls nicht gefunden?` aktiviert ist. ^[1]
- `/Set` : Ändern eines bestehenden Objekts.

Formatierung der Anfrage

Die Daten werden dem Webservice entweder **in der URL kodiert** als **GET**-Argumentübertragung, oder als **JSON formatierte Zeichenkette** mittels **POST**-Argumentübertragung übergeben.

Eine URL kann in eine JSON-Datei konvertiert werden und eine JSON-Datei in eine oder mehrere URLs.

Ein Datum wird immer als `DD.MM.YYYY` übertragen.

Schema für URL-Kodierung

Der [Prozent-Encodierte Query-String](#) der URL muss folgendes Format haben. Parameter in eckigen Klammern sind optional:

```
/New
?objectType=
&indexQuery=
&matchString=
[&keyDate=]
```

```
/Set
?objectType=
&indexQuery=
&matchString=
&importType=
&valueString=
[&keyDate=]
[&toDate=]
[&keyString=]
```

Schema für JSON-File

Die, dem Webservice mittels POST-Argumentübertragung, zur Verfügung gestellte JSON-Datei muss folgendem Schema entsprechen:

```
{
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "title": "Set",
  "description": "Format für Anfragen an das Set-Webservice",
  "type": "object",
  "properties": {
    "objectType": {
      "type": "string"
    },
    "indexQuery": {
      "type": "string"
    },
    "importType": {
```

```

        "type": "string"
    },
    "lines": {
        "type": "array",
        "items": {
            "type": "object",
            "properties": {
                "matchString": {
                    "type": "string"
                },
                "valueString": {
                    "type": "string"
                },
                "importType": {
                    "type": "string"
                },
                "keyString": {
                    "type": "string"
                },
                "keyDate": {
                    "type": "string",
                    "pattern": "^(3[01]|[12][0-9]|0[1-9]).(1[0-2]|0[1-9]).[0-9]{4}$"
                }
            },
            "required": [
                "matchString",
                "valueString"
            ]
        }
    }
},
"required": [
    "objectType",
    "indexQuery",
    "lines"
]
}

```

objectType

Der `objectType` enthält den Objekttyp des zu ändernden Objekts.

- Für Personal: `Employee`
- Für Benutzer: `AlexUser`

indexQuery

Der `indexQuery` enthält den Index über den das Objekt gesucht wird.

- Für Personal: `EmployeeIDX`
- Für Benutzer: `UserX`

matchString

Der `matchString` enthält den Wert für die Indexsuche.

- Für Personal die Personalnummer.
- Für Benutzer der Benutzername.

importType

Der `importType` enthält die Eigenschaft die geändert werden soll.

valueString

Der `valueString` enthält den Wert der zu setzenden Eigenschaft als Zeichenkette. Eigenschaften können folgende Werte annehmen:

- Zeichenketten: "Mayr", "Groß", "Ü"
- Ganze Zahlen: "1", "2", "10000"
- Gleitkommazahlen: "1.123", "1.4567"
- Fremdschlüssel: Verweis auf "Station 1" in Eigenschaft "Stammplanungseinheit"
- Komplexe Werte: Kinder für Pflege-Anspruch oder Anspruchszeile
 - Format uncodiert: Property1####Value&Property2####Value ... &PropertyN####Value
 - **Muss in URL codiert werden!**

keyDate (optional)

Das `keyDate` enthält den Wert des **Von**-Datums der zu setzenden Eigenschaft.

toDate (optional)

Das `toDate` enthält den Wert des **Bis**-Datums der zu setzenden Eigenschaft.

keyString (optional)

Der `keyString` enthält den Schlüsselwert der zu setzenden Eigenschaft.

Formatierung der Ausgabe

JSON-String nach dem Schema:

```
{
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "title": "New/Set-Result",
  "description": "Format für Ausgabe des New/Set-Webservice",
  "type": "object",
  "properties": {
    "request": {
      "type": "string",
      "description": "Angeforderter Request als 'Handshake'"
    },
    "status": {
      "type": "string",
      "description": "'ok' oder 'error'"
    },
    "details": {
      "type": "string",
      "description": "Optionale Detailinformationen bei 'error'"
    }
  },
  "required": [
    "request",
    "status"
  ]
}
```

Beispiele

Beispiel: Ein Personal mit Standardinformationen anlegen

Diese Aufrufe des Webservice:

```

/New?objectType=Employee&indexQuery=EmployeeIDX&matchString=16
/Set?
objectType=Employee&indexQuery=EmployeeIDX&matchString=16&importType=MasterAllocation&valueString=SR&keyDate=01.01.2022
/Set?
objectType=Employee&indexQuery=EmployeeIDX&matchString=16&importType=Surname&valueString=Mustermann
/Set?
objectType=Employee&indexQuery=EmployeeIDX&matchString=16&importType=Forename&valueString=Johann
/Set?
objectType=Employee&indexQuery=EmployeeIDX&matchString=16&importType=Nickname&valueString=Hans
/Set?
objectType=Employee&indexQuery=EmployeeIDX&matchString=16&importType=EmployeeTitleBeforeName&valueString=
/Set?
objectType=Employee&indexQuery=EmployeeIDX&matchString=16&importType=EmployeeTitleAfterName&valueString=
/Set?
objectType=Employee&indexQuery=EmployeeIDX&matchString=16&importType=Sex&valueString=1
/Set?
objectType=Employee&indexQuery=EmployeeIDX&matchString=16&importType=DayOfBirth&valueString=29.03.1965
/Set?
objectType=Employee&indexQuery=EmployeeIDX&matchString=16&importType=EmployeeJobGroup&valueString=V
/Set?
objectType=Employee&indexQuery=EmployeeIDX&matchString=16&importType=Employed&keyDate=01.07.1993
/Set?
objectType=Employee&indexQuery=EmployeeIDX&matchString=16&importType=EmployeeEmailAddress&valueString=johann.mustermann@gmail.com
    
```

führen zu diesem Personalstamm:

Personalstammdaten importieren	
Eigenschaften	ALEX [®] -Person
<ul style="list-style-type: none"> • Personalnummer • Zuname • Vorname • Kurzname • Titel vor Name • Titel nach Name • Geschlecht • Geburtstag • Berufsgruppe • Beschäftigungszeitraum • Stamm-Planungseinheit • E-Mail-Adresse 	

Allgemein	Tarifvertrag	Urlaubsansp
Qualifikationen	Schwerarbeitsverordn	
Personalnummer	<input type="text" value="16"/>	
Zuname	<input type="text" value="Mustermann"/>	
Vorname	<input type="text" value="Johann"/>	
Kurzname	<input type="text" value="Hans"/>	
Titel vor Name	<input type="text"/>	
Titel nach Name	<input type="text"/>	
Geschlecht	<input type="text" value="männlich"/>	
Geburtstag	<input type="text" value="29.03.1965"/>	
Berufsgruppe	<input type="text" value="Vertrieb"/>	
Beschäftigt	<input type="text" value="01.07.1993"/>	
Stamm-Planungseinheit	<input type="text" value="Steyr"/>	
E-Mail Adresse	<input type="text" value="johann.muster"/>	

Beispiel: Neues Objekt anlegen

Objektyp	Beschreibung	URL Aufruf	JSON Aufruf
Personal	Neues Personal mit Personalnummer 1	<pre>/New ?objectType=Employee &indexQuery=EmployeeIDX &matchString=1</pre>	nicht möglich
Benutzer	Neuer Benutzer mit Benutzernamen U1	<pre>/New ?objectType=AlexUser &indexQuery=UserX &matchString=U1</pre>	nicht möglich

Beispiel: Eigenschaften in Personal ändern

Der Einfachheit halber nehmen wir an, es gibt bereits ein Personal mit Personalnummer .

Änderung	URL Aufruf (codiert)	JSON Aufruf
		<pre>{ "objectType": "Employee", "indexQuery": "EmployeeIDX", "importType": "EmployeeEmailAddress", "lines": [{</pre>

Änderung	URL Aufruf (codiert)	JSON Aufruf
Email auf test@example.com	<pre> /Set ?objectType=Employee &indexQuery=EmployeeIDX &matchString=1 &importType=EmployeeEmailAddress &valueString=test@example.com </pre>	<pre> "matchString": "1", "valueString": "test@example.com" }] } </pre>
Kartennummer auf 3 ab 01.01.2020	<pre> /Set ?objectType=Employee &indexQuery=EmployeeIDX &matchString=1 &importType=EmployeeCardID &valueString=3 &keyDate=01.01.2020 </pre>	<pre> { "objectType": "Employee", "indexQuery": "EmployeeIDX", "importType": "EmployeeCardID", "lines": [{ "matchString": "1", "valueString": "3", "keyDate": "01.01.2020" }] } </pre>
Zuname auf Mustermann ab 01.10.2020	<pre> /Set ?objectType=Employee &indexQuery=EmployeeIDX matchString=1 &importType=Surname &valueString=Mustermann &keyDate=01.10.2020 </pre>	<pre> { "objectType": "Employee", "indexQuery": "EmployeeIDX", "importType": "Surname", "lines": [{ "matchString": "1", "valueString": "Mustermann", "keyDate": "01.10.2020" }] } </pre>
Vorname auf Max	<pre> /Set ?objectType=Employee &indexQuery=EmployeeIDX &matchString=1 &importType=Forename &valueString=Max </pre>	<pre> { "objectType": "Employee", "indexQuery": "EmployeeIDX", "importType": "Forename", "lines": [{ "matchString": "1", "valueString": "Max" }] } </pre>

Änderung	URL Aufruf (codiert)	JSON Aufruf
Beschäftigungsausmaß auf 62,5% ab 01.03.2022	<pre data-bbox="496 338 1007 584">/Set ?objectType=Employee &indexQuery=EmployeeIDX &matchString=1 &importType=BESCH_GRAD_IN_PROZENT NT &valueString=62,5 &keyDate=01.03.2022</pre>	<pre data-bbox="1023 226 1445 696">{ "objectType": "Employee", "indexQuery": "EmployeeIDX", "importType": "BESCH_GRAD_IN_PROZENT", "lines": [{ "matchString": "1", "valueString": "62,5", "keyDate": "01.03.2022" }] }</pre>
Austritt auf 31.08.2022	<pre data-bbox="496 943 1007 1122">/Set ?objectType=Employee &indexQuery=EmployeeIDX &matchString=1 &importType=Employed &toDate=31.08.2022</pre>	<pre data-bbox="1023 786 1445 1211">{ "objectType": "Employee", "indexQuery": "EmployeeIDX", "importType": "Employed", "lines": [{ "matchString": "1", "valueString": "", "toDate": "31.08.2022" }] }</pre> <p data-bbox="1023 1256 1054 1301">[2]</p>
Berufsgruppe auf B1	<pre data-bbox="496 1480 1007 1659">/Set ?objectType=Employee &indexQuery=EmployeeIDX &matchString=1 &importType=EmployeeJobGroup &valueString=B1</pre>	<pre data-bbox="1023 1368 1445 1749">{ "objectType": "Employee", "indexQuery": "EmployeeIDX", "importType": "EmployeeJobGroup", "lines": [{ "matchString": "1", "valueString": "B1" }] }</pre>
		<pre data-bbox="1023 1839 1445 2074">{ "objectType": "Employee", "indexQuery": "EmployeeIDX", "importType": "MasterAllocation", "lines": [</pre>

Änderung	URL Aufruf (codiert)	JSON Aufruf
Stammplanungseinheit auf P1	<pre> /Get ?objectType=Employee &indexQuery=EmployeeIDX &matchString=1 &importType=MasterAllocation &valueString=P1 &keyDate=01.09.2022 </pre>	<pre> { "matchString": "1", "valueString": "P1", "keyDate": "01.09.2022" } </pre>
Stammplanungseinheit P1 abgrenzen	<pre> /Get ?objectType=Employee &indexQuery=EmployeeIDX &matchString=1 &importType=MasterAllocation &valueString=P1 &keyDate=01.09.2022 &toDate=16.09.2022 </pre>	<pre> { "objectType": "Employee", "indexQuery": "EmployeeIDX", "importType": "MasterAllocation", "lines": [{ "matchString": "1", "valueString": "P1", "keyDate": "01.09.2022", "toDate": "16.09.2022" }] } </pre>
Qualifikation AS von 01.09.2022 bis 31.12.2022	<pre> /Get ?objectType=Employee &indexQuery=EmployeeIDX &matchString=1 &importType=EmployeeQualifications &keyString=AS &keyDate=01.09.2022 &toDate=31.12.2022 </pre>	<pre> { "objectType": "Employee", "indexQuery": "EmployeeIDX", "importType": "EmployeeQualifications", "lines": [{ "matchString": "1", "keyString": "AS", "keyDate": "01.09.2022", "toDate": "31.12.2022" }] } </pre>
Anspruch von Urlaub (<pre> /Get ?objectType=Employee &indexQuery=EmployeeIDX &matchString=1 &importType=EmployeeEntitlement_Urlaub &valueString =EmployeeRecEntitlementKeyDate% 23%23%23%2302.02.2012%26 </pre>	<pre> { "objectType": "Employee", "indexQuery": "EmployeeIDX", "importType": "EmployeeEntitlement_Urlaub", "lines": [{ "matchString": "1", "valueString": "EmployeeRecEntitlementKe </pre>

Änderung	URL Aufruf (codiert)	JSON Aufruf
beachten	EmployeeRecEntitlementCustomPostingKeyDate%23%23%23%2303.02.2012%26EmployeeRecEntitlementValue%23%23%23%23200,00&keyDate=01.10.2020	yDate####02.02.2012&EmployeeRecEntitlementCustomPostingKeyDate####03.02.2012&EmployeeRecEntitlementValue####200,00", "keyDate": "01.10.2020" }] }
Kind 1	/Set ?objectType=Employee &indexQuery=EmployeeIDX &matchString=PNR123 &importType=EmployeeChildren &keyString=0 &valueString=ChildName%23%23%23%23Franz%26ChildBirthday%23%23%23%2303.02.2012	{ "objectType": "Employee", "indexQuery": "EmployeeIDX", "importType": "EmployeeChildren", "lines": [{ "matchString": "1", "keyString": "0", "valueString": "ChildName###Franz&ChildBirthday###03.02.2012" }] }
Mutterschutz von 01.01.2020 bis 01.08.2020	/Set ?objectType=Employee &indexQuery=EmployeeIDX &matchString=1 &importType=MaternityProtection &valueString=1 &keyDate=01.01.2020 &toDate=01.08.2020	{ "objectType": "Employee", "indexQuery": "EmployeeIDX", "importType": "MaternityProtection", "lines": [{ "matchString": "1", "valueString": "1", "keyDate": "01.01.2020", "toDate": "01.08.2020" }] }
Poolzuteilung auf P1 (2387)		{ "objectType": "Employee", "indexQuery": "EmployeeIDX", "importType": "PoolAllocations", "lines": [{ "matchString": "1", "keyString": "2387",

Änderung	URL Aufruf (codiert)	JSON Aufruf
	<pre> /Set ?objectType=Employee &indexQuery=EmployeeIDX &matchString=1 &importType=PoolAllocations &keyString=2387 &valueString= &keyDate=01.09.2022 </pre>	<pre> "valueString": "", "keyDate": "01.09.2022" }] } </pre>

Besonderheiten

Beschäftigungsausmaß

Die Eigenschaft **Beschäftigungsausmaß in Prozent** wird je nach Abrechnungsart unterschiedlich angesprochen:^[3]

- nur_7up: `EmploymentFactorPercent`
- nur Standard: `BESCH_GRAD_IN_PROZENT`
- gemischt: beides erlaubt

Beispiel: Anspruchszeile

```

/Set?
objectType=Employee&indexQuery=EmployeeIDX&matchString=PNR123&importType=EmployeeEntitlement
nt_Urlaub&valueString=EmployeeRecEntitlementKeyDate%23%23%23%2302.02.2012%
26EmployeeRecEntitlementCustomPostingKeyDate%23%23%23%2303.02.2012%
26EmployeeRecEntitlementValue%23%23%23%23200,00&keyDate=01.10.2020
                    
```

EmployeeEntitlement_Urlaub → hier muss nach dem _ der Name der Anspruchsart gesendet werden

EmployeeRecEntitlementKeyDate → Stichtag der Anspruchszeile

EmployeeRecEntitlementCustomPostingKeyDate → Zubuchungstag der Anspruchszeile

EmployeeRecEntitlementValue → Zusatzanspruch der Anspruchszeile

keyDate → Beginn-Datum der Anspruchszeile - meist Eintrittsdatum oder 1. des Eintrittsmonats (hier gilt auch die oben beschriebene Logik bzgl. dem Maximum zwischen keyDate und "Ist bis")

ACHTUNG: Anspruchszeilen sollten nur einmalig bei der Initialanlage des Personals eingespielt werden. Alle folgenden Änderungen (z.B. Gültigkeit der Anspruchszeile erhöhen) müssen direkt in Alex[®] eingegeben werden.

Beispiel: Eigenschaften in Benutzer ändern

Der Einfachheit halber nehmen wir an, es gibt bereits einen Benutzer mit Benutzernamen `U1`.

Änderung	URL Aufruf (codiert)	JSON Aufruf
		<pre> { "objectType": "AlexUser", "indexQuery": "UserX", "importType": "UserEmail", "lines": [</pre>

Änderung	URL Aufruf (codiert)	JSON Aufruf
Email auf test@example.com	<pre>/Set ?objectType=AlexUser &indexQuery=UserX &matchString=U1 &importType=UserEmail &valueString=test@example.com</pre>	<pre>{ "matchString": "U1", "valueString": "test@example.com" }</pre>
Benutzername auf U2	<pre>/Set ?objectType=AlexUser &indexQuery=UserX &matchString=U1 &importType=Username &valueString=U2</pre>	<pre>{ "objectType": "AlexUser", "indexQuery": "UserX", "importType": "Username", "lines": [{ "matchString": "U1", "valueString": "U2" }] }</pre>

Beispiel 1: Email in Personal ändern

Ausgangssituation

Im folgenden Beispiel verfügt die Mitarbeiterin mit der Personalnummer 4, Birgit Bauer über eine hinterlegte E-Mail-Adresse

- E-Mail-Adresse: bauer@bitfactory.com
- Personalnummer: 4

Mitarbeiter Birgit Bauer

Allgemein	Soll	Abrechnung	Arbeitszeitgesetzprüfungen	Automatischer Pausenabzug	Erweit
Qualifikationen	Schwerarbeitsverordnung	Steuerung	Zeitzuschläge div. KV's	Überstunden	Vert
Personalnummer	<input type="text" value="4"/>				
Zuname	<input type="text" value="Bauer"/>				
Vorname	<input type="text" value="Birgit"/>				
Kurzname	<input type="text" value="BBauer"/>				
Titel vor Name	<input type="text" value="Diplom-Ingenieurin"/>				
Titel nach Name	<input type="text" value="BA"/>				
Geschlecht	<input type="text" value="weiblich"/>				
Geburtstag	<input type="text" value="01.01.1978"/>				
Berufsgruppe	<input type="text" value="Arbeitstier"/>				
Beschäftigt	<input type="text" value="01.03.2023"/> <input type="text" value="31.12.2099"/>				
Stamm-Planungseinheit	<input type="text" value="Plan A"/>				
Lohnartenschlüsselung	<input type="text" value="Standardumschlüsselung"/>				
Selbstbedienungsbenutzer	<input type="text" value=""/>				
E-Mail Adresse	<input type="text" value="bauer@bitfactory.com"/>				
Kartennummer	<input type="text" value=""/>				
Stempelungen verwenden	<input type="checkbox"/>				

Aufruf der Funktion

Um die E-Mail-Adresse von einem Mitarbeiter mit der Personalnummer 4 zu ändern, muss das folgende JSON-Format als Zeichenkette mit der POST-Methode übertragen werden.

```
{
  "objectType": "AlexUser",
  "indexQuery": "UserX",
  "importType": "UserEmail",
  "lines": [
```

```
{
  "matchString": "4",
  "valueString": "birgit.bauer@bitfactory.at"
}
]
```

Ergebnis

JSON Datei

```
{
  "request": "/Set",
  "status": "ok"
}
```

Datenbank für Beispiel

[DokuDB Bsp 1](#)

Cheatsheet

Eigenschaft	importType	valueString
Stamm-Planungseinheit	MasterAllocation	wie konfiguriert
Zuname	Surname	Zeichenkette
Vorname	Forename	Zeichenkette
Kurzname	Nickname	Zeichenkette
Titel vor Name	EmployeeTitleBeforeName	Zeichenkette
Titel nach Name	EmployeeTitleAfterName	Zeichenkette
Geschlecht	Sex	0 = weiblich 1 = männlich
Geburtstag	DayOfBirth	Datum
Berufsgruppe	EmployeeJobGroup	wie konfiguriert
Beschäftigt	Employed	leer
E-Mail Adresse	EmployeeEmailAddress	Zeichenkette
Lohnartenumschlüsselung	EmployeeAccountToWageTypeMap	wie konfiguriert
Kartenummer	EmployeeCardID	Zeichenkette
Inaktivzeiten	MaternityProtection	0 = Schwanger 1 = Mutterschutz 2 = Karenz 3 = Sonstige Inaktivzeit weitere kundenspezifische möglich
Telefonnummer	EmployeePhoneNumber	Zeichenkette

Eigenschaft	importType	valueString
Sozialversicherungs-Nr.	EmployeeSocialInsurance	Zeichenkette
Straße	EmployeeStreet	Zeichenkette
Handzeichen für Pflegedoku	EmployeeSignature	Zeichenkette
Einstufung	EmployeeJobEvaluation	Zeichenkette
Beschäftigungsausmaß in Prozent ^[3]	BESCH_GRAD_IN_PROZENT EmploymentFactorPercent	Gleitkommazahl
Sollstunden Montag	SOLL_MO	Gleitkommazahl
Sollstunden Dienstag	SOLL_DI	Gleitkommazahl
Sollstunden Mittwoch	SOLL_MI	Gleitkommazahl
Sollstunden Donnerstag	SOLL_DO	Gleitkommazahl
Sollstunden Freitag	SOLL_FR	Gleitkommazahl
Sollstunden Samstag	SOLL_SA	Gleitkommazahl
Sollstunden Sonntag	SOLL_SO	Gleitkommazahl
Sollstunden FEIERTAG	SOLL_FT	Gleitkommazahl
Fixes Monatssoll in Stunden	SOLL_MONAT_FIX	Gleitkommazahl
Arbeitstage je Woche	WorkingDaysPerWeek	Gleitkommazahl
Anspruchsart	EmployeeEntitlement_NAME	EmployeeRecEntitlementKeyDate: Stichtag EmployeeRecEntitlementCustomPostingKeyDate: Zubuchungstag EmployeeRecEntitlementValue: Zusatzanspruch
Kinder für Pflegefreistellung	EmployeeChildren	ChildName: Name ChildBirthday: Geburtstag
Poolzuteilungen	PoolAllocations	leer

Zeitbereich Korrektur bei Personal

Im Personalstamm gibt es Eigenschaften, die zeitlich abgegrenzt werden können, zum Beispiel:

- Stammpfanungseinheit
- Beschäftigungsgrad
- Sollstunden

Für deren Änderung muss ein Zeitbereich angegeben werden.

Diese Eigenschaften beeinflussen die Berechnung der Zeitkonten und dürfen sich auf abgeschlossenen Plänen nicht ändern.

Darum können keine Eigenschaften vor dem **Abgeschlossen-Bis**-Datum der aktuellen Stammpfanungseinheit des Personals durch das Webservice geändert werden.

Der Zeitbereich der Änderung wird nach folgenden Kriterien angenommen:

- wird ein `keyDate` übergeben, wird dieses verwendet.
- wird kein `keyDate` an die Schnittstelle übergeben, wird automatisch der **1. des aktuellen Monats** angenommen.
- wird ein `toDate` übergeben, wird dieses verwendet.
- wird kein `toDate` an die Schnittstelle übergeben, wird automatisch **unendlich** angenommen.

Der entstandene Zeitbereich wird nun mit dem **Abgeschlossen-Bis**-Datum begrenzt.

Beispiel

Abgeschlossen-Bis	*										
1. aktueller Monat	0	1	2	3	4	5	6	7	8	9	inf
Zeitbereich							*	*	*	*	
Zeitbereich korr.							*	*	*	*	
Zeitbereich	*	*	*	*	*	*	*	*	*	*	
Zeitbereich korr.	*	*	*	*	*	*	*	*	*	*	
Zeitbereich						*	*	*	*	*	->
Zeitbereich korr.						*	*	*	*	*	->
Zeitbereich	*	*	*			*	*	*	*	*	->
Zeitbereich korr.	*	*	*			*	*	*	*	*	->

Referenzen auf andere Objekte

Der Import Referenzen auf andere Objekte ist möglich über

- die interne `ObjectID` von ALEX.
- externe Kennzeichen, die zuvor initialisiert werden müssen.

Sind Externe Kennzeichen gesetzt kann trotzdem über die `ObjectID` importiert werden:

- Parameter `Systemeinstellungen: Object Import Einstellungen: Eigenschaften-Import: Referenzen entsprechen 'ObjectID'?`

Systemeinstellungen für Objekt Import

Object Import Einstellungen	Eigenschaften-Import Pfad	<input type="text"/>	
Passwort-Richtlinie	Eigenschaften-Import Dateierweiterung	<input type="text"/>	
Pflege	Fremdschlüssel anzeigen	●	Nein
Planungscode AZG	Eigenschaften-Import: Referenzen entsprechen 'ObjectID'?	●	Ja
Projektkonten Berechnung	Eigenschaften-Import: Neu erstellen falls nicht gefunden?	●	Nein
QA Sperre	Eigenschaften-Import: Benutzer für Webservice		
QrCode			
Reports			

Fremdschlüssel stehen für folgende Objekttypen zur Verfügung:

- Berufsgruppe
- Planungseinheit
- Qualifikation
- Vertrag
- Lohnartenumschlüsselung

File Import

Zusätzlich zum Webservice können Dateien im JSON-Format auch über einen Daemon-Prozess importiert werden.

Der Daemon muss im Configfile konfiguriert werden.

Der Pfad und die Endung der Dateien kann in den Systemeinstellungen eingestellt werden.

- Systemeinstellungen: Object Import Einstellungen: Eigenschaften-Import Pfad
- Systemeinstellungen: Object Import Einstellungen: Eigenschaften-Import Dateierweiterung

Dateien werden alle 2 Minuten importiert.

Interaktiver Test von POST

Unter Systemeinstellung -> rechte Maustaste auf Reiter -> Test Object Import kann ein Eingabefenster geöffnet werden.

In dieses Eingabefenster können Dateien im JSON-Format kopiert werden.

Damit kann das Format des POST Mechanismus auf Richtigkeit getestet werden.

Test Object Import

Protokollierung

Damit Änderungen über die Webservice-Schnittstelle protokolliert werden, muss beim Parameter "Eigenschaften-Import: Benutzer für Webservice" ein Benutzer hinterlegt werden.

Der hinterlegte Benutzer benötigt SUPERVISOR-Berechtigung für das ganze System.

Benutzer WEBSERVICE

Protokoll Personal

Geändert um	Geänderte Eigenschaft	Alter Wert	Neuer Wert	Geändert von	Geändert auf Computer
Änderung auf: Huber Christa, 367					
09.09.2022 10:24:56.09	E-Mail Adresse		test@example.com	WEBSERVICE	Bfx.HTTP.Server.dll

Fußnoten

1. ↑ #7363
2. ↑ #8070
3. ↑ ^{3,0} ^{3,1} #8393