

Inhaltsverzeichnis

1. Kategorie:Schnittstellen	2
2. Abwesenheitsschnittstelle	4
3. Automatischer Import des Bedarfs	5
4. Automatischer Import von Kartennummern	6
5. Automatischer Konto-Report-Export	7
6. Bfx.Abstract.Model.WebService.dll	12
7. Dienstplan als PDF im Browser	29
8. Dienstpläne im Kalender anzeigen	32
9. ELDA Schnittstelle	38
10. Eigenschaften importieren	41
11. Export SAP-BI	45
12. Export Tagesinformationen	54
13. Externe Anspruchsverwaltung	57
14. Import Anspruch	61
15. Import Einsatzplan	62
16. Import Monatswerte	63
17. Import Personal	65
18. Import Zeitstempelungen	66
19. Lohnverrechnungsschnittstelle	70
20. Lohnüberleitung	73
21. Präfix für Lohnarten- und Abwesenheitsschnittstelle	78
22. Schnittstellen	79
23. Stundennachweis PDF anzeigen über Webservice	84
24. Webservice	86
25. Webservice Personalstammdaten Export	87
26. Webservice Tagesinformationen	92
27. Wt-Webservice	115

Kategorie:Schnittstellen

Seiten in der Kategorie „Schnittstellen“

Folgende 26 Seiten sind in dieser Kategorie, von 26 insgesamt.

A

- [Abwesenheitsschnittstelle](#)
- [Automatischer Import des Bedarfs](#)
- [Automatischer Import von Kartennummern](#)
- [Automatischer Konto-Report-Export](#)

B

- [Bfx.Abstract.Model.WebService.dll](#)

D

- [Dienstplan als PDF im Browser](#)
- [Dienstpläne im Kalender anzeigen](#)

E

- [Eigenschaften importieren](#)
- [ELDA Schnittstelle](#)
- [Export SAP-BI](#)
- [Export Tagesinformationen](#)
- [Externe Anspruchsverwaltung](#)

I

- [Import Anspruch](#)
- [Import Einsatzplan](#)
- [Import Monatswerte](#)
- [Import Personal](#)
- [Import Zeitstempelungen](#)

L

- [Lohnverrechnungsschnittstelle](#)
- [Lohnüberleitung](#)

P

- [Präfix für Lohnarten- und Abwesenheitsschnittstelle](#)

S

- [Schnittstellen](#)
- [Stundennachweis PDF anzeigen über Webservice](#)

W

- [Webservice](#)

- [Webservice Personalstammdaten Export](#)
- [Webservice Tagesinformationen](#)
- [Wt-Webservice](#)

Abwesenheitsschnittstelle

Was ist das Ziel einer ABW-Schnittstelle?

Das Ziel der Abwesenheitsschnittstelle ist die Übertragung gewünschter Abwesenheiten in eine Datei mit passendem Format, um dann in ein Fremdsystem importiert und damit weiterverarbeitet zu werden.

Wie komme ich zu einer Datei, die Abwesenheiten in einem bestimmten Format beinhaltet?

[Abwesenheitsexport_\(Windows\)](#)

Automatischer Import des Bedarfs

Der Bedarf für [Besetzungsanforderungen](#) und [Aufgaben](#) kann auch automatisch importiert werden.

Dies ist vor allem dann von Nutzen, wenn sich der Bedarf oft ändert.

Wir haben für diesen Zweck eine Standardschnittstelle entwickelt, kontaktieren Sie uns gerne, falls Sie daran Interesse haben.

Ablauf

Damit der Bedarf im System eingespielt werden kann, muss ein File im definierten [Schnittstellenformat](#) in einem definierten Ordner abgelegt werden.

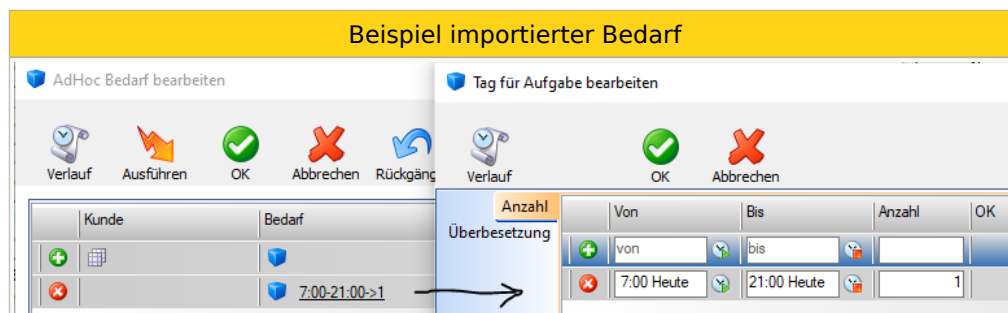
Dieser Ordner wird in einem definierten Intervall auf neue Dateien geprüft - gibt es eine neue Datei, so wird diese eingelesen und als *„processed“* abgespeichert.

Kontrolle im Dienstplan

Der Bedarf wird als "Ad-Hoc Bedarf" am jeweiligen Kalendertag importiert.

Damit man diesen kontrollieren kann, muss man die Besetzungsanforderung / Aufgabe im Besetzungsreport einfügen und am jeweiligen Tag rechte Maustaste -> "Bedarf bearbeiten" auswählen.

Hier ist der importierte Bedarf bzw. auch die importierte Überbesetzung des ausgewählten Tages sichtbar.



Links

- [Parametrierung](#)

Automatischer Import von Kartennummern

Ab Version 2019.10 können Kartennummern über eine Schnittstelle automatisch importiert werden.

Wenn Sie diese Schnittstelle ebenfalls benötigen, kontaktieren Sie uns bitte für ein Angebot.

Automatischer Konto-Report-Export

Der **automatische Konto-Report-Export** ist eine [Programmfunktion](#), die es ermöglicht einen [Konto-Report](#) periodisch am Server auszuführen.

Die durch den Konto-Report erzeugte **Exportdatei** wird auf einem Verzeichnis abgelegt und kann von dort zur weiteren Verarbeitung abgeholt werden.

Dies kann durch

- direkten Zugriff über das Dateisystem oder
- ein [Webservice](#)

erfolgen.

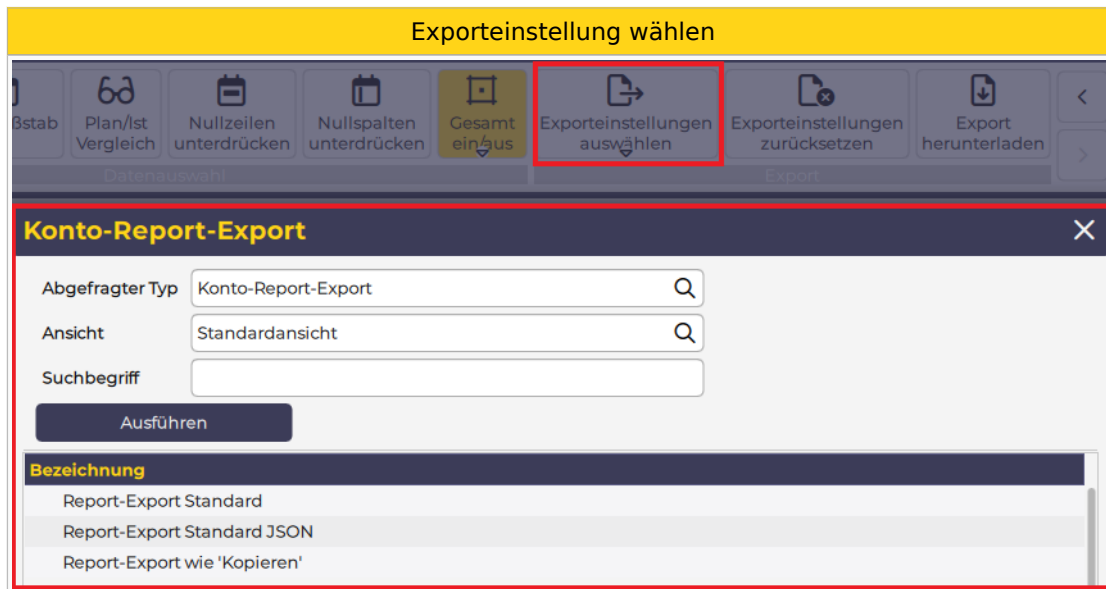
Inhaltsverzeichnis

1 Anlage Konto-Report	8
2 Anlage automatischer Konto-Report Export	8
2.1 Manueller Funktionstest	10
2.2 Automatik testen	10
3 Bit Factory interne Zusatzinfos	11

Anlage Konto-Report

Vorgehensweise zur Anlage eines neuen Konto-Reports:

1. Konto-Report aus Navigator mit grünem Plus erzeugen
2. Konto-Report umbenennen. (Wichtig für Name von automatisch exportierten Dateien)
3. Wertauswahl (Konten, Parameter, ...) durchführen
4. Den gewünschten Zeitmaßstab eintragen (für Konto-Report-Export meist Monat oder Tag).
5. Exporteinstellungen auswählen (einen **Konto-Report-Export** hinterlegen):



Weitere Informationen, wie ein Konto-Report angelegt wird, sind [hier](#) zu finden.

1. Testen Reports

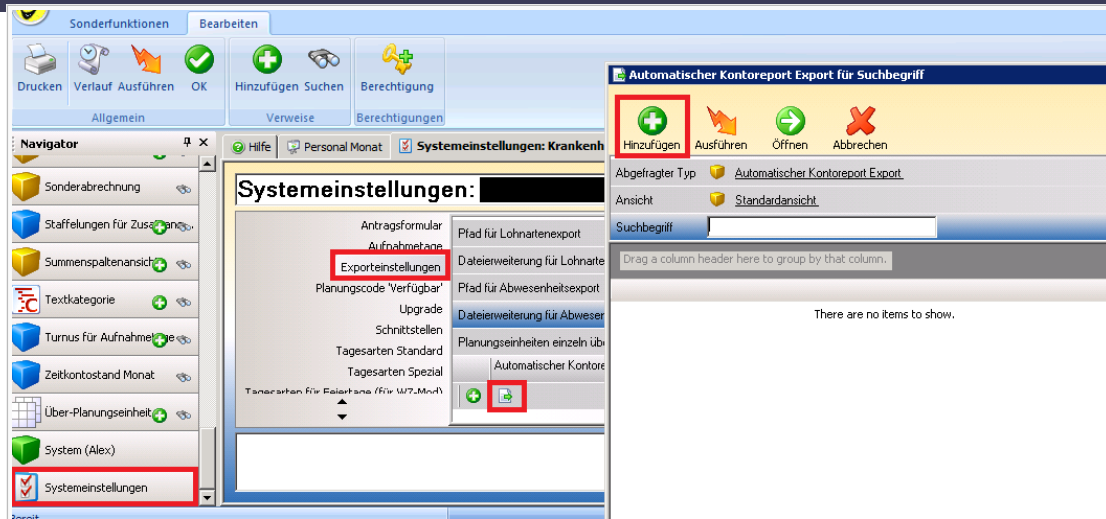
1. Dazu im Reiter "**Export**" das gewünschte Exportfile angeben und mittels "**Export herunterladen**" die Datei auf dem Computer speichern. Es wird eine Zip Datei erstellt, in welcher sich die exportierte Datei befindet.
2. Alternativ kann die Datei direkt abgespeichert werden. Weitere Informationen, wie ein Konto-Report direkt abgespeichert wird, sind [hier](#) zu finden.

Anlage automatischer Konto-Report Export

Um dem Daemon-Prozess die zu exportierenden Konto-Reports mitzuteilen, müssen diese in den Systemeinstellungen eingetragen werden:

In Systemeinstellungen - Gruppe "Exporteinstellungen" - Eigenschaft "Automatische Kontoreport Exports" einen neuen Eintrag machen:

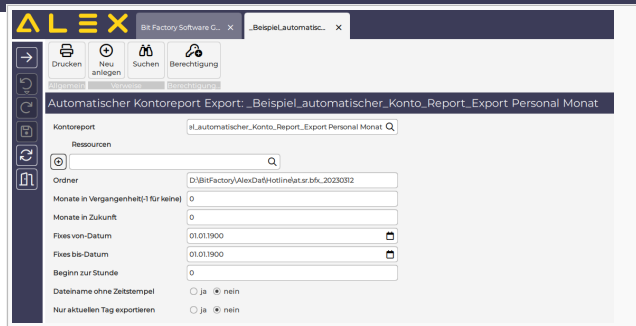
Automatischen Konto-Report Export anlegen



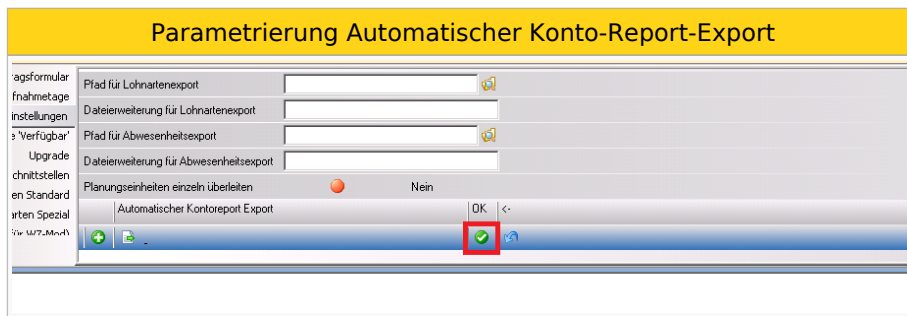
Im erzeugten "Automatischen Konto-Report-Export" können folgende Einstellungen getroffen werden:

Parameter	Wert
Konto-Report	auszuführender Konto-Report
Ressourcen	Ressource (Bereich, Planungseinheit), für welche der Konto-Report ausgeführt wird. Keine Auswahl bedeutet "Systemweit", Personal bringt kein Ergebnis.
Ordner	Ordner für erstellte Dateien, falls nicht angegeben wird Ordner "AutomaticAccountsReportExport" im Ordner des Config-Files angelegt (Dateiname = Name des ausgewählten Konto-Report)
Monat in Vergangenheit (-1 für keine)	Anzahl der zu exportierenden Monate in die Vergangenheit, vom aktuellen Monat ausgehend. (0 für 1 Monat)
Monate in Zukunft	Anzahl der zu exportierenden Monate in die Zukunft, vom aktuellen Monat ausgehend
Fixes von-Datum	Konto-Report wird mit diesem von-Datum ausgeführt
Fixes bis-Datum	Konto-Report wird mit diesem bis-Datum ausgeführt
Beginn zur Stunde	Stunde, zu welcher der Konto-Report ausgeführt wird. Nur ganze Stunden möglich. Konto-Reports werden sequentiell ausgeführt. Dauer eines Konto-Reports kann den Beginn des Nächsten verschieben. Sind mehrere Konto-Reports zu selben Zeit geplant, werden diese nach Erstellungsreihenfolge abgearbeitet.
Dateiname ohne Zeitstempel	Exportierte Datei erhält keinen Zeitstempel und wird täglich überschrieben.
Nur aktuellen Tag exportieren	Konto-Report wird mit Zeitbereich "aktueller Tag" ausgeführt. Nur mit "Zeitmaßstab Tag" sinnvoll. (Verwendung: Export Kartenummer tagesaktuell)

Parametrierung Automatischer Konto-Report-Export

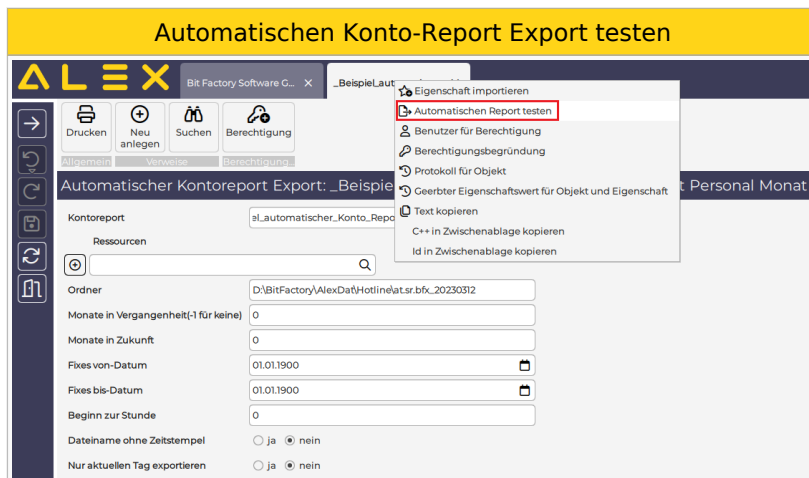


Hinterlegung des "Automatischen Konto-Report-Export" in Systemeinstellungen.



Manueller Funktionstest

Ein **manueller Funktionstest** kann über einen Rechtsklick auf den Reiter des Automatischen Konto-Reports durchgeführt werden. Der Konto-Report wird in dem zuvor definierten Ordner abgelegt.



Mögliche technische Hindernisse können sein:

- Der Benutzer, welcher den Alex-Dienst ausführt, hat andere Berechtigungen als der Benutzer, mit dem in Alex gearbeitet wird
- Die Netzwerkfreigabe erlaubt keinen Schreibzugriff auf den angegebenen Ordner

Automatik testen

Der Konto-Report wird 1x täglich ausgeführt. Man kann daher den Parameter **Beginn zur Stunde** nicht auf eine spätere Uhrzeit stellen, sodass der Report nochmals ausgeführt wird.

Es besteht aber die Möglichkeit, einen weiteren Report anzulegen.

Achtung: Beim Starten des Dienstes wird der Report auch ausgeführt!

Bit Factory interne Zusatzinfos

- [Aktivierung](#)

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	13
1.1 Voraussetzungen	13
1.2 Formatierung der Anfrage	13
1.2.1 Schema für URL-Kodierung	13
1.2.2 Schema für JSON-File	13
1.2.3 objectType	14
1.2.4 indexQuery	14
1.2.5 matchString	14
1.2.6 importType	14
1.2.7 valueString	15
1.2.8 keyDate (optional)	15
1.2.9 toDate (optional)	15
1.2.10 keyString (optional)	15
1.3 Formatierung der Ausgabe	15
2 Beispiele	15
2.1 Beispiel: Ein Personal mit Standardinformationen anlegen	15
2.2 Beispiel: Neues Objekt anlegen	17
2.3 Beispiel: Eigenschaften in Personal ändern	17
2.3.1 Besonderheiten	22
2.3.1.1 Beschäftigungsausmaß	22
2.4 Beispiel: Anspruchszeile	22
2.5 Beispiel: Eigenschaften in Benutzer ändern	22
2.6 Beispiel 1: Email in Personal ändern	23
2.6.1 Ausgangssituation	23
2.6.2 Aufruf der Funktion	23
2.6.3 Ergebnis	24
2.6.4 Datenbank für Beispiel	24
2.7 Cheatsheet	24
3 Zeitbereich Korrektur bei Personal	26
3.1 Beispiel	26
4 Referenzen auf andere Objekte	26
5 File Import	27
6 Interaktiver Test von POST	27
7 Protokollierung	28
8 Fußnoten	28

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 "Stammplanningseinheit"
- 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
• 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	16	
Zuname	Mustermann	
Vorname	Johann	
Kurzname	Hans	
Titel vor Name		
Titel nach Name		
Geschlecht	männlich	
Geburtstag	29.03.1965	
Berufsgruppe	Vertrieb	
Beschäftigt	01.07.1993	
Stamm-Planungseinheit	Steyr	
E-Mail Adresse	johann.muster	

Beispiel: Neues Objekt anlegen

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

Beispiel: Eigenschaften in Personal ändern

Der Einfachheit halber nehmen wir an, es gibt bereits ein Personal mit Personalnummer 1.

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> /Set ?objectType=Employee &indexQuery=EmployeeIDX &matchString=1 &importType=BESCH_GRAD_IN_PROZENT NT &valueString=62,5 &keyDate=01.03.2022 </pre>	<pre> { "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> /Set ?objectType=Employee &indexQuery=EmployeeIDX &matchString=1 &importType=Employed &toDate=31.08.2022 </pre>	<pre> { "objectType": "Employee", "indexQuery": "EmployeeIDX", "importType": "Employed", "lines": [{ "matchString": "1", "valueString": "", "toDate": "31.08.2022" }] } </pre> [2]
Berufsgruppe auf B1	<pre> /Set ?objectType=Employee &indexQuery=EmployeeIDX &matchString=1 &importType=EmployeeJobGroup &valueString=B1 </pre>	<pre> { "objectType": "Employee", "indexQuery": "EmployeeIDX", "importType": "EmployeeJobGroup", "lines": [{ "matchString": "1", "valueString": "B1" }] } </pre>
		<pre> { "objectType": "Employee", "indexQuery": "EmployeeIDX", "importType": "MasterAllocation", "lines": [</pre>

Änderung	URL Aufruf (codiert)	JSON Aufruf
Stammplanungseinheit auf P1	<code>/Set ?objectType=Employee &indexQuery=EmployeeIDX &matchString=1 &importType=MasterAllocation &valueString=P1 &keyDate=01.09.2022</code>	<pre>{ "matchString": "1", "valueString": "P1", "keyDate": "01.09.2022" }</pre>
Stammplanungseinheit P1 abgrenzen	<code>/Set ?objectType=Employee &indexQuery=EmployeeIDX &matchString=1 &importType=MasterAllocation &valueString=P1 &keyDate=01.09.2022 &toDate=16.09.2022</code>	<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	<code>/Set ?objectType=Employee &indexQuery=EmployeeIDX &matchString=1 &importType=EmployeeQualifications &keyString=AS &keyDate=01.09.2022 &toDate=31.12.2022</code>	<pre>{ "objectType": "Employee", "indexQuery": "EmployeeIDX", "importType": "EmployeeQualifications", "lines": [{ "matchString": "1", "keyString": "AS", "keyDate": "01.09.2022", "toDate": "31.12.2022" }] }</pre>
Anspruch von Urlaub (	<code>/Set ?objectType=Employee &indexQuery=EmployeeIDX &matchString=1 &importType=EmployeeEntitlement_Urlaub &valueString=EmployeeRecEntitlementKeyDate%23%23%23%2302.02.2012%26</code>	<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	<code>/Set ?objectType=AlexUser &indexQuery=UserX &matchString=U1 &importType=UserEmail &valueString=test@example.com</code>	<pre>{ "matchString": "U1", "valueString": "test@example.com" }</pre>
Benutzername auf U2	<code>/Set ?objectType=AlexUser &indexQuery=UserX &matchString=U1 &importType=Username &valueString=U2</code>	<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

Verf

Personalnummer

4

Zuname

Bauer

Vorname

Birgit

Kurzname

BBauer

Titel vor Name

Diplom-Ingenieurin

Titel nach Name

BA

Geschlecht

weiblich

Geburtsdag

01.01.1978

Berufsgruppe

Arbeitstier

Beschäftigt

01.03.2023 31.12.2099

Stamm-Planungseinheit

Plan A

Lohnartenumschlüsselung

Standardumschlüsselung

Selbstbedienungsbenutzer

bauer@bitfactory.com

Kartennummer

Stempelungen verwenden

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.												
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		
Passwort-Richtlinie	Eigenschaften-Import Dateierweiterung		
Pflege			
Planungscode AZG	Fremdschlüssel anzeigen		Nein
Projektkonten Berechnung	Eigenschaften-Import: Referenzen entsprechen 'ObjectID'?		Ja
QA Sperre	Eigenschaften-Import: Neu erstellen falls nicht gefunden?		Nein
QrCode			
Reports	Eigenschaften-Import: Benutzer für Webservice		

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.

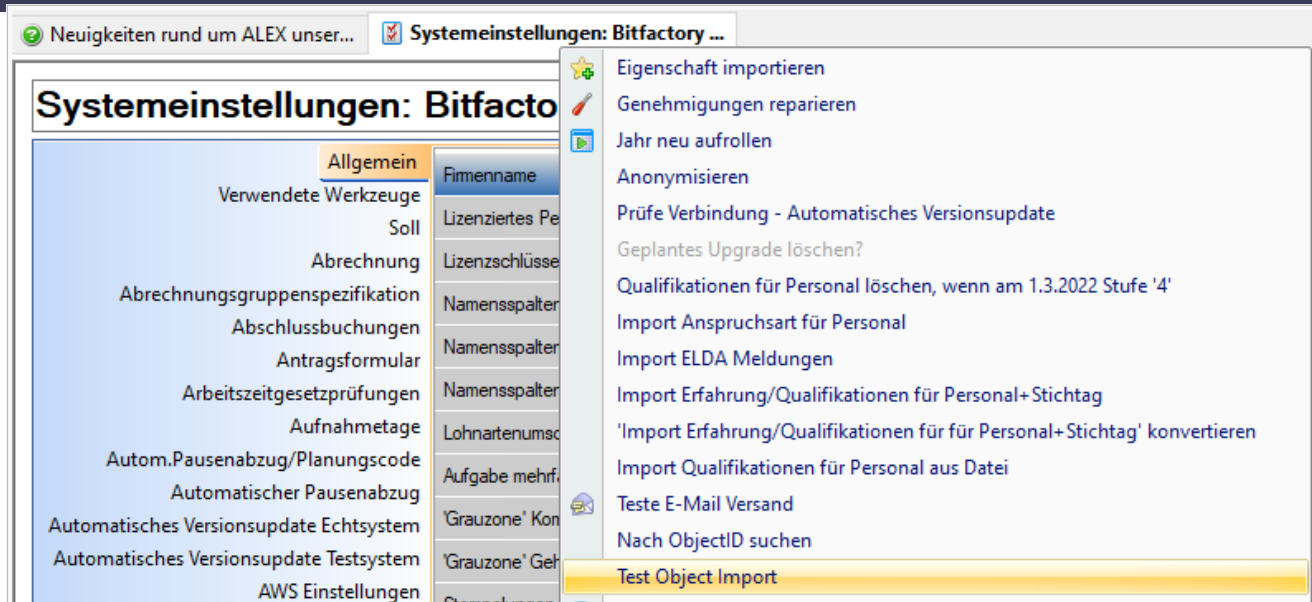
Interaktiver Test von POST

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.

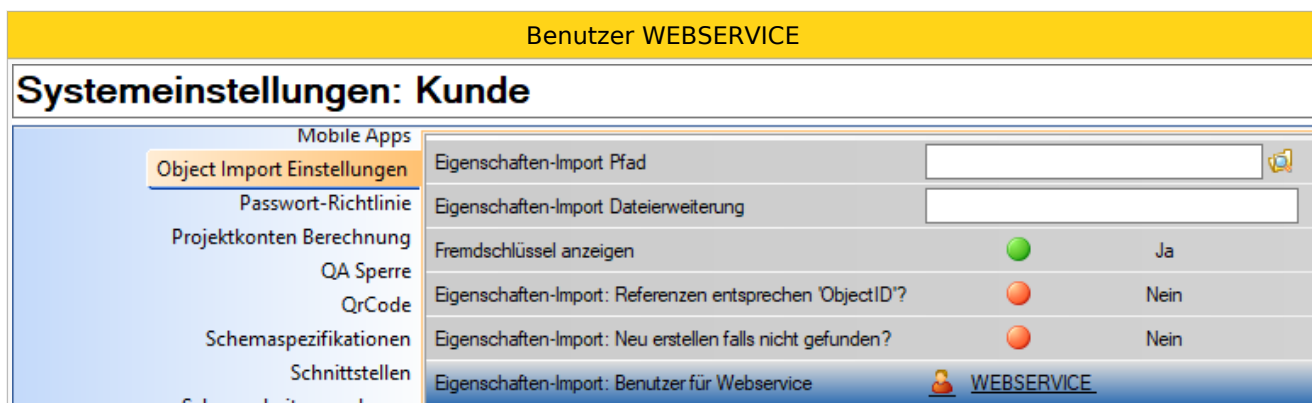
Test Object Import



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.



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](#)

Dienstplan als PDF im Browser

Das [WebUI-Webservice](#) - **Dienstplan PDF** ist eine [Programmfunktion](#) zur Abfrage des Dienstplans pro Personal oder Planungseinheit und Datum im Dateiformat PDF. ^[1]

Inhaltsverzeichnis

1 Ausgangssituation	30
1.1 Voraussetzungen	30
1.2 Formatierung der Anfrage	30
1.2.1 Schema für URL-Kodierung	30
1.3 Formatierung der Ausgabe	31
1.3.1 Status von Dienstplan nicht erfüllt ^[2]	31
2 Fußnoten	31

Ausgangssituation

Voraussetzungen

- Es müssen die Voraussetzungen für den Betrieb des [WebUI-Webservice](#) erfüllt sein sein.
- Es muss ein Benutzer mit den gewünschten Druckeinstellungen konfiguriert werden.
 - Systemeinstellungen: Druck-Einstellungen: Benutzer für Webservice Ausdruck von Monatsplan
- Der Benutzer muss mindestens die Rechte **Dienstplan lesen** besitzen.

Ausdruck Benutzer		
Druckvorschau mit Adobe Reader		Nein
Firmenname 'leer' auf Ausdruck		Nein
Benutzer für Webservice-Ausdruck von Dienstplan	<u>Ausdruck Benutzer</u>	
Minimaler Status für Webservice-Ausdruck von Dienstplan	Planung	▼
Ausdruck: Namensspalte-Icons nicht drucken		Nein
Ausdruck: Informationen zum IST-Wechsel drucken		Nein
Ausdruck: Meldungsfenster Seitenwechsel nach jedem Personal		Nein
Ausdruck: Im Plan Alles mit großem Font anzeigen?		Nein
Ausdruck: Im Plan KEIN 'Gelöscht' zeichnen?		Nein

- Es kann ein **minimaler Status für Webservice-Ausdruck von Dienstplan** ^[2] ausgewählt werden. Danach werden nur Pläne ab diesem Status angezeigt. Falls nichts eingestellt wird, werden alle Status angezeigt.
 - Systemeinstellungen: Druck-Einstellungen: minimaler für Webservice-Ausdruck von Dienstplan

Formatierung der Anfrage

Die Daten werden dem Webservice **in der URL kodiert** als **GET**-Argumentübertragung übergeben.

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:

```
/AlexPrintPlanPeriod
[?planCode=]
[&employeeId=]
[&date=]
```

Parameter	Wertebereich	Beschreibung
planCode	Zeichenkette	Kurzzeichen einer Planungseinheit
employeeId	Zeichenkette	Eine Personalnummer, die Planungseinheit ergibt sich aus der Stammzuteilung
		Datum des Monats des gewünschten Stundennachweises, wenn nicht

Parameter	Wertebereich	Beschreibung
date	Datum	angegeben Heute

Beispiel:

- `http://192.168.0.100:8080/AlexPrintPlanPeriod?date=10.06.2023&employeeId=123`

Formatierung der Ausgabe

Bei einer erfolgreichen Anfrage gibt das Webservice eine **Datei im PDF-Format** zurück

Bei einem Fehler wird ein JSON-String nach folgendem Schema zurückgegeben:

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

Status von Dienstplan nicht erfüllt ^[2]

Falls der **minimaler Status für Webservice-Ausdruck von Dienstplan** nicht erfüllt ist, wird folgende Fehlermeldung als Text im Response zurückgegeben:

```
{
  "request": "/AlexPrintPlanPeriod?date=<date>&employeeId=<id>",
  "status": "error",
  "details": "Dienstplan mit Status >Planung< noch nicht verfügbar."
}
```

Der Response Status wird auf `404 Not Found` gesetzt.

Fußnoten

1. ↑ [#8097](#)
2. ↑ [2.0 2.1](#) [#8715](#)

Dienstpläne im Kalender anzeigen

Mit dieser Funktion können sich Ihre Mitarbeiter den ALEX - Dienstplan im Google- oder Outlook-Kalender anzeigen lassen.

Der Aktualisierungsintervall ist von der eingesetzten Kalender-Software (Google Kalender, Outlook,...) abhängig.

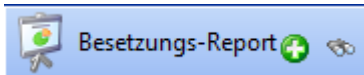
Inhaltsverzeichnis

1 Besetzungsreport anlegen	33
2 iCal-Link per E-Mail versenden oder generieren	34
2.1 iCal-Link per Mail versenden	34
2.2 OPTIONAL: iCal-Link generieren	34
3 Installation im Kalender	35
3.1 Google-Kalender	35
3.2 Outlook-Kalender	35
3.2.1 bis Outlook 2007	36
3.2.2 Outlook 2010	36
3.2.3 Outlook 2013	37
4 Bit Factory interne Zusatzinfos	37

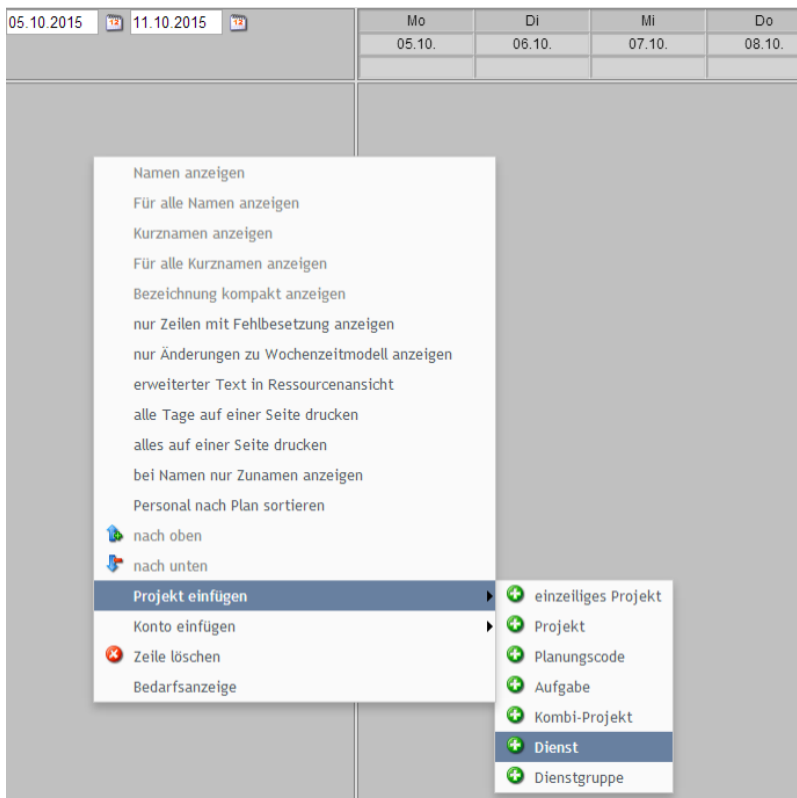
Besetzungsreport anlegen

Um festzulegen, welche Dienste in den Google- oder Outlook-Kalender übertragen werden sollen, ist es notwendig, einen Besetzungsreport anzulegen. Dabei gehen Sie bitte wie folgt vor:

- Im Navigator bei dem Punkt „Besetzungsreport“ auf das grüne Plus klicken (= neuen Report anlegen).



- Den Besetzungsreport umbenennen, indem Sie in der Werkzeugleiste auf  klicken.
- Fügen Sie anschließend jene Dienste ein, die Sie gerne im Google- oder Outlook-Kalender angezeigt haben möchten:
 - Machen Sie dazu einen Rechtsklick im grauen Feld und wählen Sie „Projekt einfügen“ -> „Dienst“ aus.
 - Wählen Sie nun jene Planungseinheit aus, für die Sie den Besetzungsreport anlegen → Planungseinheit mit Doppelklick auswählen.
 - Klicken Sie auf „Ausführen“.
 - Es werden alle Dienste der Planungseinheit angezeigt.
 - Markieren Sie die relevanten Dienste und klicken Sie auf „Öffnen“.



- Wenn Sie auch Planungscode, Aufgaben etc. im Google- oder Outlook-Kalender anzeigen möchten, gehen Sie wie folgt vor:
 - Machen Sie einen Rechtsklick im grauen Feld und wählen Sie „Projekt einfügen“ -> „Planungscode“ / „Aufgabe“ etc. aus.
 - Markieren Sie die gewünschten Planungscode/Aufgaben etc. und klicken Sie anschließend auf „Öffnen“.

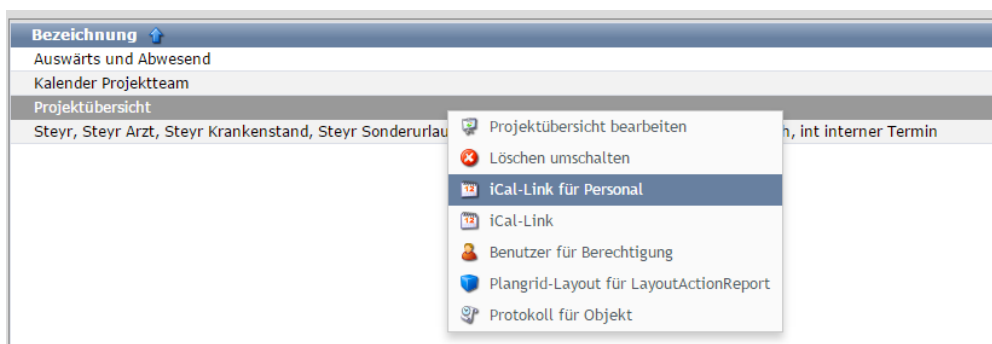
iCal-Link per E-Mail versenden oder generieren

iCal-Link per Mail versenden

- Versenden an einzelnes Personal:
 - im Navigator bei Personal auf die Lupe klicken
 - Mitarbeiter suchen
 - beim Mitarbeiter Rechtsklick -> "iCal-Link für Personal als E-Mail versenden"
- Versenden an gesamtes Personal:
 - im Navigator bei Personal auf die Lupe klicken
 - beim Suchbegriff einen "*" eingeben und die Suche ausführen
 - alle Mitarbeiter markieren -> Rechtsklick -> "iCal-Link für Personal als E-Mail versenden"
- **WICHTIG!:** E-Mail-Adresse eintragen
 - Im Mitarbeiterstammblatt finden Sie im Reiter "Allgemein" das Feld „E-Mail Adresse“. Um den Link senden zu können, muss eine gültige E-Mail-Adresse bei jedem Mitarbeiter eingetragen werden.

OPTIONAL: iCal-Link generieren

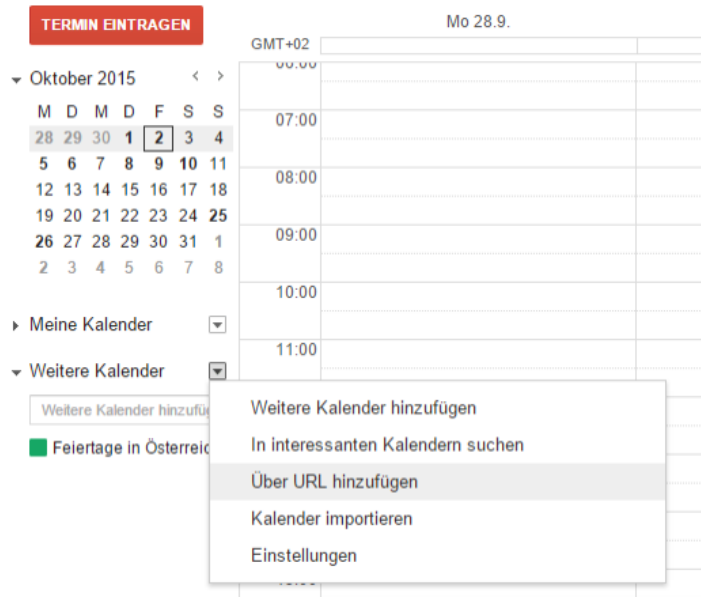
- Dienstplan eines bestimmten Mitarbeiters für den Google- oder Outlook-Kalender freigeben:
 - im Navigator bei Besetzungsreport auf die Lupe klicken
 - gewünschten Besetzungsreport suchen
 - beim Besetzungsreport Rechtsklick -> "iCal Link für Personal"
 - es öffnet sich das Fenster „Personal“ → gewünschten Mitarbeiter suchen und mittels Doppelklick auswählen
 - den nun generierten Link an den Mitarbeiter weitergeben
- Dienstplan des gesamten Personals für den Google- oder Outlook-Kalender freigeben (in diesem Fall wird der gesamte Dienstplan, d.h. von allen Mitarbeitern, in den Google- oder Outlook-Kalender importiert)
 - im Navigator bei Besetzungsreport auf die Lupe klicken
 - gewünschten Besetzungsreport suchen
 - beim Besetzungsreport Rechtsklick -> "iCal Link für Personal"
 - den nun generierten Link an den Mitarbeiter weitergeben



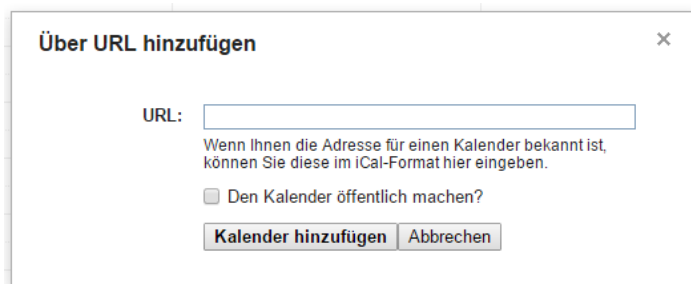
Installation im Kalender

Google-Kalender

- unter „Weitere Kalender“ auf „Über URL hinzufügen“ klicken

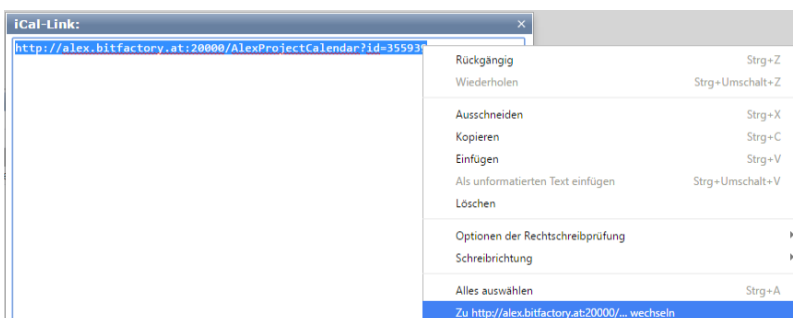


- den Link im Feld „URL“ einfügen und anschließend auf „Kalender hinzufügen“ klicken



Outlook-Kalender

- iCal-Link Datei auf Ihrem PC speichern:
 - im Navigator bei Besetzungsreport auf die Lupe klicken
 - gewünschten Besetzungsreport suchen
 - beim Besetzungsreport Rechtsklick -> "iCal Link für Personal" bzw. „iCal Link“ auswählen
 - Um den generierten Link zu speichern, muss dieser markiert und mit Rechtsklick „Zu <http://alex...>“ ausgewählt werden



- Es öffnet sich ein Fenster, um die Datei zu speichern

Dateiname: alex_calendar.ics

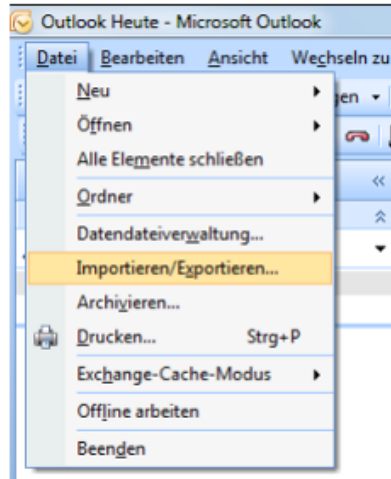
Dateityp: ICS-Datei (.ics)

Ordner ausblenden

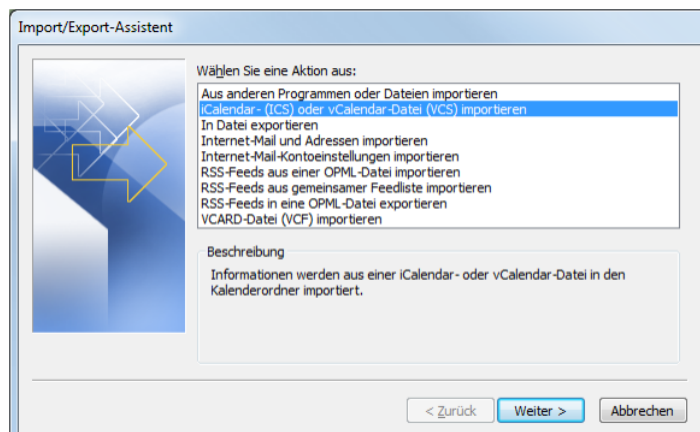
Speichern Abbrechen

bis Outlook 2007

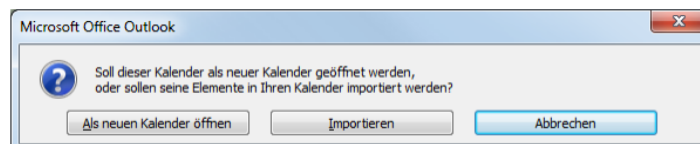
bis Outlook 2007



- Im nächsten Schritt fragt Sie Outlook nach dem zu importierenden Format. Wählen Sie "iCalendar oder vCalendar Date importieren".



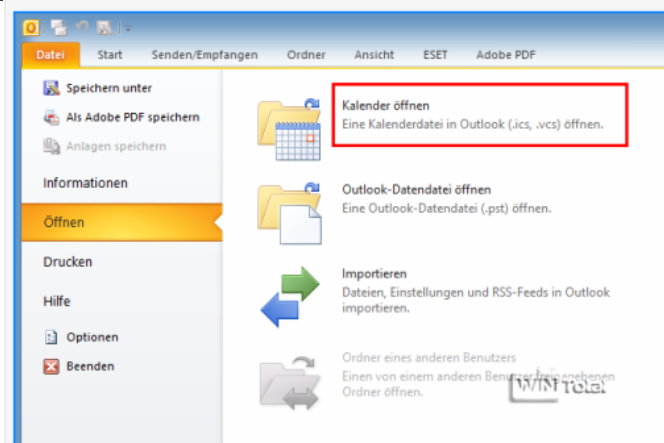
- Wählen Sie nun die zuvor auf Ihrem PC gespeicherte ics-Datei aus (zb: alex_calender.ics).
- Wenn Sie die Datei ausgewählt haben, werden Sie gefragt, ob Sie den Kalender als neuen Kalender öffnen möchten oder ob Sie die Termine in Ihren Kalender integrieren möchten.



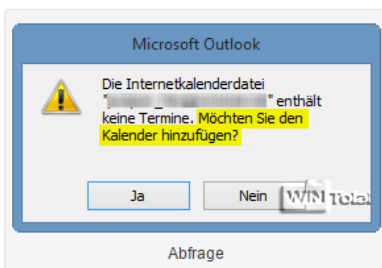
- Als neuen Kalender öffnen: Hier wird ein neuen Kalender im Outlook erstellt.
- Importieren: Hier werden die Termine in Ihren bestehenden Kalender integriert.

Outlook 2010

- Starten Sie Ihr Outlook-Programm. Klicken Sie auf "Datei" – "Öffnen" und auf "Kalender öffnen". Suchen Sie die iCalendar-Datei (*.ics).



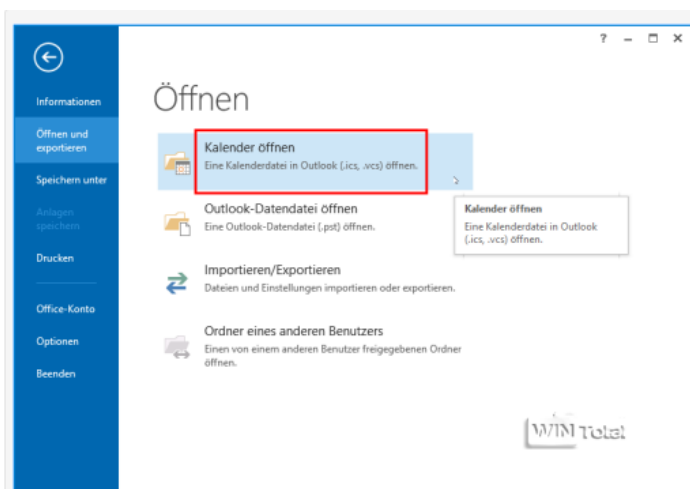
Outlook 2010



Abfrage

Outlook 2013

- Starten Sie Ihr Outlook-Programm, klicken auf "Datei" – "Öffnen und exportieren" und auf "Kalender öffnen". Suchen und öffnen Sie die iCalendar-Datei (*.ics).



Outlook 2013

Bit Factory interne Zusatzinfos

- [Aktivierung](#)

ELDA Schnittstelle

Die ELDA-Schnittstelle ist für den Abgleich mit den bestätigten Krankenständen seitens der Sozialversicherungsträger und den im Alex erfassten Krankenstandsdaten konzipiert.

Inhaltsverzeichnis

1 ELDA Import	39
2 ELDA Report	40
3 Parametrierung	40
4 Links	40

ELDA Import

Die vom Sozialversicherungsträger erhaltenen Daten können laufend oder gesamthaft (z. B. einmal pro Monat) eingespielt werden.

Über Systemeinstellung -> Import ELDA Meldungen kann eine einzelne Datei oder ein ganzer Ordner mit mehreren Dateien auf einmal für den Import ausgewählt werden.



Import-Fehler: Eine Fehlermeldung beim Import entsteht beispielsweise, wenn eine Sozialversicherungsnummer nicht gefunden wird.

Die importierten Daten werden über die Funktion **ELDA Import** dargestellt.

The screenshot shows the 'ELDA Import' window. It has a search bar for 'Planungscode' and a table with two columns: 'Personal' and 'Zeitleiste'.

Personal	Zeitleiste
MIKLAUTZ Gerhard, 374	25.05.2022 - 01.01.2100
WOPINSKI Martin, 355	27.05.2022 - 01.01.2100
HASLINGER Christoph, 188	28.05.2022 - 01.01.2100
BACHMANN Marcel, 16	29.05.2022 - 01.01.2100
STEINER Josef, 458	29.05.2022 - 01.01.2100

Mittels Doppelklick auf eine Importzeile kann der importierte Zeitbereich bei Bedarf verändert bzw. gelöscht werden.

The screenshot shows the 'ELDA Import Korrektur' window. It has a search bar for 'ELDA Import für ELDA Import von...' and a table with columns for 'Allgemein', 'Zeitleiste', 'Von', 'Bis', 'OK', and '<-'. The table shows three rows of data for 'MIKLAUTZ Gerhard, 374 KUR, KR'.

Allgemein	Zeitleiste	Von	Bis	OK	<-
		01.09.2022	31.12.2099		
		06.05.2022	08.05.2022		
		25.05.2022	31.12.2099		

ELDA Report

Der Abgleich der Krankenstandsdaten kann jederzeit über die Funktion **ELDA Report** ausgeführt werden, dabei wird eine Differenzliste erstellt.

ELDA Report		
Abgefragter Typ	ELDA Report	
Ansicht	Standardansichten EmployeeEldaReportResultView	
Suchbegriff		
von	01.07.2022	
bis	31.07.2022	
Personal	ELDA Zeitbereich	aktueller Zeitbereich
Planungscode: K		
AKDUMAN Jürgen, 112	nicht vorhanden	27.07.2022 - 24.08.2022
ALASADI Csaba, 107	nicht vorhanden	11.07.2022 - 11.07.2022
BAKHSHI Bernhard, 424	nicht vorhanden	14.07.2022 - 19.07.2022
BAUER-KANDL Klaus, 630	28.06.2022 - 01.01.2100	28.06.2022 - 22.07.2022
Bauer Thomas, 131	nicht vorhanden	30.07.2022 - 04.08.2022
BERGER Rene, 276	nicht vorhanden	18.07.2022 - 02.08.2022
BÖHMWALDER Paul, 171	06.07.2022 - 01.01.2100	06.07.2022 - 27.07.2022
BRAUN Julian, 306	nicht vorhanden	19.07.2022 - 24.07.2022
CIMEN Gunter, 57	nicht vorhanden	29.07.2022 - 05.08.2022
DAXBÖCK Hubert, 484	11.07.2022 - 01.01.2100	11.07.2022 - 11.07.2022
DELIVUK Martin, 164	11.03.2022 - 09.07.2022	28.06.2022 - 09.07.2022
DING Maurice, 117	nicht vorhanden	30.07.2022 - 31.07.2022
ECKER Peter, 60	nicht vorhanden	29.07.2022 - 31.07.2022
EDELBACHER Josef, 382	30.05.2022 - 01.01.2100	nicht vorhanden

Parametrierung

- [ELDA Schnittstelle/Parametrierung](#)

Links

- [Krankenstands-Dashboard](#)
- [Schnittstellen](#)

Eigenschaften importieren

Mit dieser Erweiterung können alle beliebigen Eigenschaften, wie zum Beispiel E-Mail Adressen oder Benutzernamen, in Alex mit einer .txt Datei importiert werden.

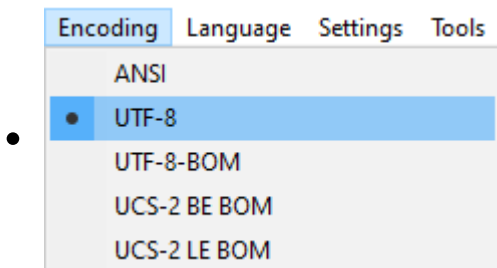
ACHTUNG:

- Falls nicht anders angegeben, ist die Gültigkeit des neuen Datensatzes von 0.0.0 bis 31.12.2099.
- Bei berechnungsrelevanten Eigenschaften wird das "Von"-Datum mit dem letzten abgeschlossenen Plan begrenzt (Abgeschlossen bis)

Satzaufbau

```
<EindeutigeKennungDesObjekts>;<GewünschterWert>;<Stichtag>[;<Schlüsselwert>]
```

- Feldlänge beliebig
- Trennzeichen ";"
- Dezimaltrennzeichen = "." oder ","
- Textdatei (.txt im Dateinamen sowie im gesamten Pfad darf kein Umlaut sein!)
- Der EOL Typ der Datei muss Windows sein (siehe Kapitel "Bekannte Probleme" ganz unten)
- Datei-Kodierung muss "UTF-8" sein (ohne "BOM"). Die Datei-Kodierung kann im Notepad++ wie folgt kontrolliert werden:



Beispiel: E-Mail Adressen importieren: <Personalnummer>;<Email>

```
1;Test.email@icloud.com
11;Test.email@icloud.com
4;Test.email@icloud.com
6;Test.email@icloud.com
```

Beispiel: Nachnamen importieren: <Personalnummer>;<Nachname>;<Stichtag>

```
1;Huber;01.10.2015
11;Hauser;11.12.2015
4;Mayr;01.03.2013
6;Berger;30.05.2012
```

Beispiel: Benutzernamen umschlüsseln: <BenutzernameAlt>;<BenutzernameNeu>

```
user1alt;user1neu
user2alt;user2neu
user3alt;user3neu
user4alt;user4neu
```

Beispiel: Gesundheitsberuferegister - Art der Tätigkeit: <Personalnummer>;<Art der Taetigkeit>;

```
79024928;Freiberuflich
79024929;Angestellt
79024930;Sonstige
```

Beispiel: Gesundheitsberuferegister - Berufsberechtigung: <Personalnummer>;<Berufsberechtigung>;

```
79024928;aktiv
79024929;aktiv
79024930;ruhend
```

Beispiel: Gesundheitsberuferegister - Erstregistrierung: <Personalnummer>;<Erstregistrierung>;

```
79024928;01.02.2003
79024929;01.04.2005
79024930;05.02.2013
```

Beispiel: ausgewählten Parameter auf Wert "Ja" oder "Nein" setzen

```
79024928;ja
79024929;ja
79024930;nein
```

Eigenschaften in Alex importieren

Während der Durchführung des Imports darf kein Benutzer einen Personalstamm sperren, daher sollte unbedingt eine Sperre im System gesetzt werden.

Die Eigenschaften können über die Suche importiert werden. Dazu muss die zu importierende Eigenschaft in einer Ansicht angezeigt werden:

- Im Navigator den Objekttyp wählen, für den Eigenschaften importiert werden sollen.
- In der Werkzeugleiste den Punkt "Neue Ansicht" auswählen
- Über den Punkt "Ansicht umbenennen" kann eine Bezeichnung für die neue Ansicht definiert werden.
- Über den Punkt "Bearbeiten" in der Werkzeugleiste kann nun die gewünschte Eigenschaft eingeblendet werden (z. B. E-Mail-Adresse)
- Rechtsklick im Feld der zu importierenden Eigenschaft und "Eigenschaft importieren" auswählen

Rechte-Maus-Klick in die gewünschte Spalte

Abgefragter Typ: Personal Employee.

Ansicht: Zuname, Vorname, Personalnummer, Beschäftigt, E-Mail Adresse

Suchbegriff:

von: 01.10.2015

bis: 01.12.2015

Beschäftigt: Ja

Drag a column header here to group by that column.

Zuname	Vorname	Personalnummer	Beschäftigt	E-Mail Adresse
Frühling	Martina	11	ab 01.04.2015	
Herbst	Eva	4	ab 01.04.2015	
Max	Männchen	1	ab 01.04.2015	
Mustermann	Susi	6	ab 01.04.2015	
Sommer	Klaus	5	ab 01.04.2015	
Sommer	Sarah	9	ab 01.04.2015	
Winter	Verena	25	ab 01.04.2015	

- Max Männchen, 1 bearbeiten
- Löschen umschalten
- App-Konfiguration als E-Mail versenden
- Jahresplan öffnen
- Selbstbedienung lesen anlegen
- Selbstbedienung anlegen
- Eigenschaft importieren**
- Lohnartensport anlegen
- Abwesenheitsreport anlegen
- Benutzer für Berechtigung
- Protokoll für Objekt
- Jahresplan
- Export (Basis) für Ressource im Entstehungsmonat

- Die Eigenschaft für die eindeutige Kennung des Objekts auswählen (z. B. Personalnummer oder Kartennummer)

Ist die eindeutige Kennung im File die Personalnummer, muss hier die Personalnummer ausgewählt werden.

Eindeutige Kennung entspricht der ersten Spalte im File

Abfragen

Hinzufügen Ausführen Öffnen Abbrechen

Bezeichnung	Type
EmployeeCardIDX	Abfragen
EmployeeSignatureX	Abfragen
Personalnummer	Abfragen
Zusätzliche Kartennummer	Abfragen

- Danach die Datei auswählen. Die gewünschten Eigenschaften werden importiert (z. B. E-Mail-Adressen). Etwaige Fehler werden angezeigt. Nach Kontrolle des Imports kann abgespeichert oder verworfen werden.

Daten werden sofort in der jeweiligen Spalte angezeigt

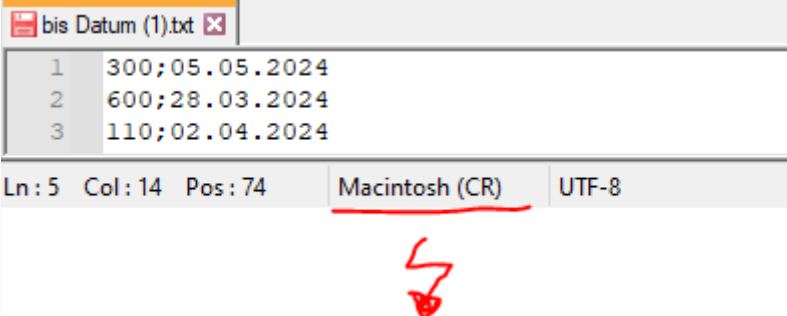
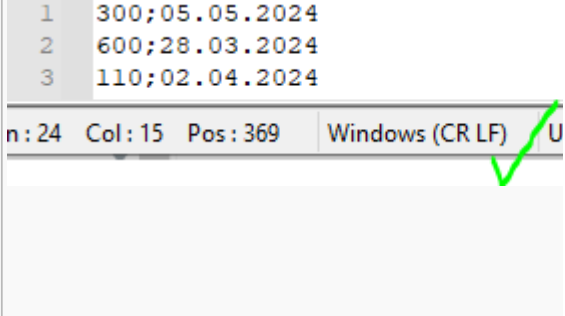
E-Mail Adresse

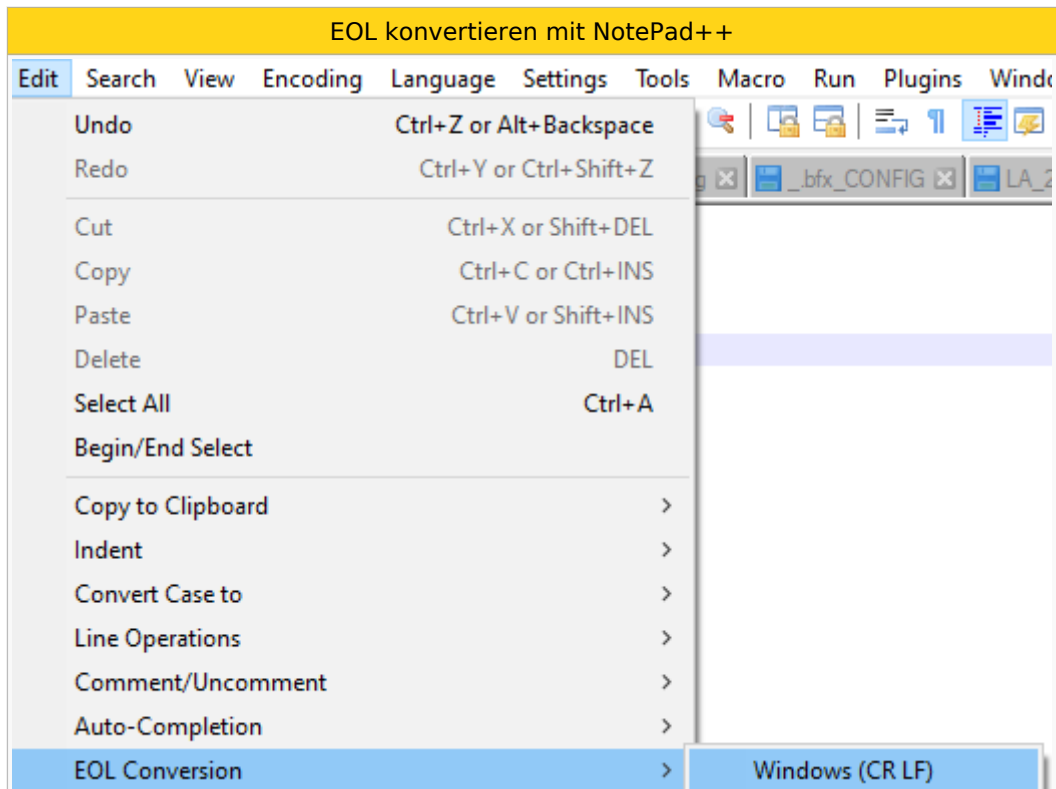
- Test.email@icloud.com
- Test.email@icloud.com
- Test.email@icloud.com
- Test.email@icloud.com
- Test.email@icloud.com
- Test.email@icloud.com
- Test.email@icloud.com

Bekannte Probleme

- Falsche Datei-Codierung: muss UTF-8 sein!
- Falscher EOL Type (=End-of-Line): muss Windows sein, siehe folgendes Beispiel:

Datei im NotePad++ geöffnet, unterste Zeile beachten:

FALSCH: EOL-Conversion darf nicht Macintosh sein		Korrekt: EOL-Conversion Windows	
			



Export SAP-BI

Inhaltsverzeichnis

1 Export von Stammdaten und Zeitkontodaten	46
1.1 Zeitspanne	46
1.2 Personalstammdaten	46
1.2.1 File-Aufbau	46
1.3 Planungseinheiten	47
1.3.1 File-Aufbau	47
1.4 Bewertungsverträge	47
1.4.1 File-Aufbau	47
1.5 PlanungsCodes	48
1.5.1 File-Aufbau	48
1.6 Dienste	48
1.6.1 File-Aufbau	48
1.7 Aufgaben	49
1.7.1 File-Aufbau	49
1.8 Zeitkonto	49
1.8.1 File-Aufbau	49
1.9 Tagesinformationen	49
1.9.1 File-Aufbau	50
1.10 Monatsinformationen	50
1.10.1 File-Aufbau	50
2 Export von PlanungsCodes	51
2.1 File-Aufbau	51
2.2 Übersicht Exporte	52
3 Parametrierung	52
4 Anwendungsbeispiel Krankenstands-Dashboard	53
5 Bit Factory interne Zusatzinfos	53

Export von Stammdaten und Zeitkontodaten

Ziel ist hier eine Datenüberleitung, die mehrmals pro Monat manuell oder automatisiert von Alex® an SAP/BI durchgeführt werden soll.

Es können sowohl Stammdaten wie auch Zeitkontodaten (Tages- und Monatszeitkonten) übergeben werden.

Der Export kann manuell über Systemeinstellungen -> "SAP/BI Export ausführen" angestoßen werden.

Es gibt auch die Möglichkeit, die Reports täglich zur eingestellten Uhrzeit automatisch zu exportieren. (Siehe Aktivierung)

Exportiert werden jeweils alle Files:

- x_TimeRange.csv
- x_Employee.csv
- x_Plan.csv
- x_Plansymbol.csv
- x_Shift.csv
- x_ColorMarker.csv
- x_AccountType.csv
- x_ValuationContract.csv
- x_DayAccounts.csv
- x_MonthAccounts.csv

Wobei x für den Mandanten steht und die Dateiendung .csv auch .txt sein könnte, je nach Parametrierung. (siehe [Parametrierung](#))

Zeitspanne

Das File x_TimeRange.csv liefert die ausgewertete Zeitspanne. Diese kann mit den [Parametern von/bis](#) in den Systemeinstellungen unter "SAP/BI Export" festgelegt werden.

Beispiel: 01.01.2020 - 30.04.2020

Personalstammdaten

Nur die ALEX®-spezifischen Daten werden geschickt.

Der Kunde, welcher diese Schnittstelle beauftragt hat, übernimmt die restlichen Personalstammdaten vom Personalinformationssystem ins SAP-BI.

File-Aufbau

Dateiname: x_Employee.csv
ACHTUNG: Trennzeichen ";" zwischen den Feldern

Feldname:	Beschreibung:
Mandant	Alphanum. Bezeichnung (für jeden Mandanten wird ein eigenes Export-

File erstellt)	Alphanum. Personalnummer
Personalnummer	TT.MM.JJJJ Beginn-Datum (siehe Anmerkung unten)
Gültigkeitsbeginn	TT.MM.JJJJ Ende-Datum (siehe Anmerkung unten)
Gültigkeitsende	Alphanum. Kurzzeichen der Stammpfanungseinheit des Mitarbeiters
Stammpfanungseinheit	Alphanum. Name des Bewertungsvertrages
Bewertungsvertrag	

Beispiel:

A;1237135;01.01.2020;30.04.2020;LAEA;AZG JD

A;1237137;01.01.2020;30.04.2020;KIAD;

A;1237146;01.01.2020;31.03.2020;M1CD;

A;1237146;01.04.2020;30.04.2020;Int;

Die Schnittstelle wurde von 01.01.2020 bis 30.04.2020 ausgewertet. Der Mitarbeiter 1237146 wechselt im Auswertungszeitraum die Stammpfanungseinheit.

ACHTUNG: Es handelt sich hier nicht um Eintritts- oder Austrittsdatum oder um die Dauer der Zuweisung dieser Stammpfanungseinheit, sondern der Auswertungszeitraum ist hier ebenfalls von Belang.

Planungseinheiten

File-Aufbau

Dateiname: x_Plan.csv

ACHTUNG: Trennzeichen ";" zwischen den Feldern

Feldname:	Beschreibung:
Mandant	Alphanum. Bezeichnung (für jeden Mandanten wird ein eigenes Export-
File erstellt)	
Kurzzeichen	Alphanum. Kurzzeichen der Planungseinheit
Gültigkeitsbeginn	TT.MM.JJJJ Beginn-Datum der Planungseinheit
Gültigkeitsende	TT.MM.JJJJ Ende-Datum der Planungseinheit
Bezeichnung	Alphanum. Bezeichnung der Planungseinheit
Ext. Kennzeichen	Alphanum. Externes Kennzeichen der Planungseinheit
Kostenstelle	Alphanum. Kostenstelle der Planungseinheit

Beispiel:

A;CAM;01.01.2010;31.12.2099;Cardiologische Ambulanz;12345;123450

A;CHA4;01.01.1900;31.12.2099;Chirurgie Station 4 A;12346;123460

Bewertungsverträge

Stammdaten der Bewertungsverträge werden exportiert.

File-Aufbau

Dateiname: x_ValuationContract.csv

ACHTUNG: Trennzeichen ";" zwischen den Feldern

Feldname:	Beschreibung:
Mandant	Alphanum. Bezeichnung (für jeden Mandanten wird ein eigenes Export-
File erstellt)	

Kurzzeichen	Alphanum. Kurzzeichen des Vertrages
Bezeichnung	Alphanum. Bezeichnung des Vertrages

Beispiel:

A;AZG BD;Arbeitszeitgesetz Bereitschaftsdienste
A;AZG Opt Out;Arbeitszeitgesetz Opt Out

Planungscodes

Stammdaten der Planungscodes. Die Object-ID ist der der ALEX®-interne Schlüssel, denn es könnten mehrere Planungscodes mit dem gleichen Kurzzeichen angelegt sein.

File-Aufbau

Dateiname: x_PlanSymbol.csv
ACHTUNG: Trennzeichen ";" zwischen den Feldern

Feldname:	Beschreibung:
Mandant File erstellt)	Alphanum. Bezeichnung (für jeden Mandanten wird ein eigenes Export-
Object-ID	Alphanum. ALEX®-interne eindeutige Nummer
Gültigkeitsbeginn	TT.MM.JJJJ Beginn-Datum (Auswertungszeitraum)
Gültigkeitsende	TT.MM.JJJJ Ende-Datum (Auswertungszeitraum)
Plansymbol	Alphanum. Plansymbol des Planungscodes
Bezeichnung	Alphanum. Bezeichnung des Planungscodes
Planungscodart /Reserviert)	Alphanum. Planungscodart (mögliche Werte: Abwesend/Anwesend

Beispiel:
A;21009333;01.01.2020;30.04.2020;Uh;Urlaub in Stunden ;Abwesend
A;4645;01.01.2020;30.04.2020;Be;Besprechung;Anwesend

Dienste

Stammdaten der Dienste. Die Object-ID ist der ALEX®-interne Schlüssel, denn es könnten mehrere Dienste mit dem gleichen Kurzzeichen angelegt sein.

File-Aufbau

Dateiname: x_Shift.csv
ACHTUNG: Trennzeichen ";" zwischen den Feldern

Feldname:	Beschreibung:
Mandant File erstellt)	Alphanum. Bezeichnung (für jeden Mandanten wird ein eigenes Export-
Object-ID	Alphanum. ALEX®-interne eindeutige Nummer
Planungseinheit	Alphanum. Kurzzeichen der Planungseinheit
Gültigkeitsbeginn	TT.MM.JJJJ Beginn-Datum des Dienstes
Gültigkeitsende	TT.MM.JJJJ Ende-Datum des Dienstes
Plansymbol	Alphanum. Plansymbol des Dienstes
Bezeichnung	Alphanum. Bezeichnung des Dienstes
Nachtdienst	Alphanum. "N" wenn es sich um einen Nachtdienst handelt, sonst leer

Beispiel:

```
A;3701;CHS2;01.10.2003;31.12.2099;N;19:00-0:00;N
A;4601;CHS2;01.08.2014;31.12.2099;V8;7:00-12:00 und 12:30-15:30;
A;3964;CHS2;01.08.2014;31.12.2099;D1;8:00-12:00 und 12:30-16:30;
A;9930;CHS2;01.04.2017;31.12.2099;ST;6:45-15:15 mit 30 Min. Pause;
```

Aufgaben

Stammdaten der Aufgaben (früher Farbmarkierungen). Die Object-ID ist der ALEX®-interne Schlüssel, denn es könnten mehrere Aufgaben mit dem gleichen Kurzzeichen angelegt sein.

File-Aufbau

Dateiname: x_ColorMarker.csv

ACHTUNG: Trennzeichen ";" zwischen den Feldern

Feldname:	Beschreibung:
Mandant File erstellt)	Alphanum. Bezeichnung (für jeden Mandanten wird ein eigenes Export-
Object-ID	Alphanum. ALEX®-interne eindeutige Nummer
Planungseinheit	Alphanum. Kurzzeichen der Planungseinheit
Gültigkeitsbeginn	TT.MM.JJJJ Beginn-Datum der Aufgabe
Gültigkeitsende	TT.MM.JJJJ Ende-Datum der Aufgabe
Kurzzeichen	Alphanum. Kurzzeichen der Aufgabe
Bezeichnung	Alphanum. Bezeichnung der Aufgabe

Beispiel:

```
A;5620023;CHS2;01.01.2018;31.12.2099;WuDie;Wunschdienst
A;4989;CHS2;01.01.1900;31.12.2099;PAL;Praxisanleiter
```

Zeitkonto

Stammdaten der Zeitkonten Die Object-ID ist der ALEX®-interne Schlüssel.

File-Aufbau

Dateiname: x_AccountType.csv

ACHTUNG: Trennzeichen ";" zwischen den Feldern

Feldname:	Beschreibung:
Mandant File erstellt)	Alphanum. Bezeichnung (für jeden Mandanten wird ein eigenes Export-
Object-ID	Alphanum. ALEX®-internes eindeutiges Kennzeichen
Label kurz	Alphanum. Kurzbezeichnung des Kontos
Label lang	Alphanum. Kurzbezeichnung des Kontos

Beispiel:

```
A;SOLL_WOCHE;SollW;Sollstunden Woche
A;SOLL_TAG;Soll;Sollstunden Tag
A;AZG_TOTAL;AZG Std;AZG-Stunden
A;AUSZ_UEBSTD_200;Ü200;ÜStd 200%
A;ENTITLEMENT_POSTING_01;U+;Urlaub in Stunden Zubuchung
```

Tagesinformationen

Informationen zu Dauer und Anzahl der geplanten Dienste, Planungscode und Farbmarkierungen. Es kann in der [Parametrierung des Exports](#) noch ein Schemenschema hinterlegt werden, die darin vorhandenen Konten würden dann ebenfalls in diesem File enthalten sein.

File-Aufbau

Dateiname: x_DayAccounts.csv
 ACHTUNG: Trennzeichen ";" zwischen den Feldern

Feldname:	Beschreibung:
Mandant	Alphanum. Bezeichnung (für jeden Mandanten wird ein eigenes Export-File erstellt)
Personalnr	Alphanum. Personalnummer
Datum	TT.MM.JJJJ
Planstatus	Alphanum. Wert entweder Planung oder Ist
Stamplan	Alphanum. Kurzzeichen Stamplanungsseinheit des Mitarbeiters
Dienst auf	Alphanum. Kurzzeichen Planungseinheit, wo der Dienst geleistet wird
ObjectTyp	Alphanum. Shift für Dienst, ColorMarker für Farbmarkierung,
PlanSymbol für Planungscode, AccountType für Zeitkonto	
ObjectID	Alphanum. Object ID
Count/Duration	Alphanum. entweder "Count" oder "Duration", folgendes Feld "Wert" ist dann entweder die Anzahl oder die Dauer
Wert	Numerisch Anzahl oder Dauer des jeweiligen Objektes für diesen Tag für diesen Mitarbeiter

Beispiel:
 Mitarbeiter hat einen V8 Dienst geplant. Im Ist ist ein D1 Dienst eingetragen mit 4 h Planungscode "Besprechung". Über den ganzen Tag ist zusätzlich die Farbmarkierung "Praxisanleiter":
 A;1230462;17.01.2020;Planung;CHS2;CHS2;Shift;4601;Count;1,00
 A;1230462;17.01.2020;Planung;CHS2;CHS2;Shift;4601;Duration;8,00
 A;1230462;17.01.2020;Planung;CHS2;CHS2;ColorMarker;4989;Count;1,00
 A;1230462;17.01.2020;Planung;CHS2;CHS2;ColorMarker;4989;Duration;8,00
 A;1230462;17.01.2020;Ist;CHS2;CHS2;Shift;3964;Count;1,00
 A;1230462;17.01.2020;Ist;CHS2;CHS2;Shift;3964;Duration;8,00
 A;1230462;17.01.2020;Ist;CHS2;CHS2;ColorMarker;4989;Count;1,00
 A;1230462;17.01.2020;Ist;CHS2;CHS2;ColorMarker;4989;Duration;8,00
 A;1230462;17.01.2020;Ist;CHS2;CHS2;PlanSymbol;4645;Count;1,00
 A;1230462;17.01.2020;Ist;CHS2;CHS2;PlanSymbol;4645;Duration;4,00

Monatsinformationen

Es kann in der [Parametrierung des Exports](#) ein Schemenschema hinterlegt werden. Die darin vorhandenen Konten sind in diesem File enthalten.

File-Aufbau

Dateiname: x_MonthAccounts.csv
 ACHTUNG: Trennzeichen ";" zwischen den Feldern

Feldname:	Beschreibung:
Mandant	Alphanum. Bezeichnung (für jeden Mandanten wird ein eigenes Export-File erstellt)
Personalnr	Alphanum. Personalnummer
Datum	TT.MM.JJJJ
Planstatus	Alphanum. Wert entweder Planung oder Ist
Stamplan	Alphanum. Kurzzeichen Stamplanungsseinheit des Mitarbeiters
ObjectTyp	Alphanum. Shift für Dienst, ColorMarker für Farbmarkierung,

PlanSymbol für Planungscode, AccountType für Zeitkonto
 ObjectID Alphanum. Object ID
 Count/Duration Alphanum. entweder "Count" oder "Duration", folgendes Feld "Wert"
 ist dann entweder die Anzahl oder die Dauer
 Wert Numerisch Anzahl oder Dauer des jeweiligen Objektes für diesen Tag
 für diesen Mitarbeiter

Beispiel:

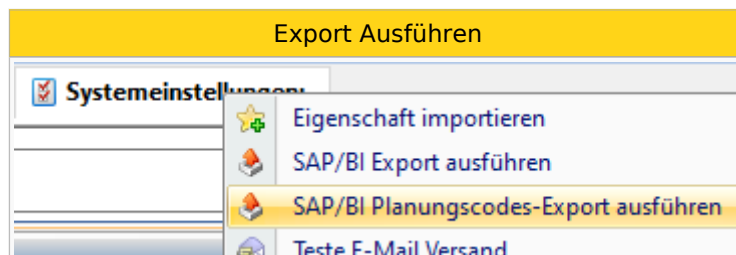
Der Mitarbeiter macht im Jänner 2020 5 ungeplante Plusstunden und hat somit einen neuen Saldo von 8,25 Stunden. Urlaubsreist 295,5 Stunden
 A;1230462;31.01.2020;Planung;CHS2;AccountType;VM_SALDO_GESAMT;Value;3,25
 A;1230462;31.01.2020;Planung;CHS2;AccountType;SALDO_GESAMT;Value;3,25
 A;1230462;31.01.2020;Planung;CHS2;AccountType;SOLL_MONAT;Value;168,00
 A;1230462;31.01.2020;Planung;CHS2;AccountType;TOTAL;Value;168,00
 A;1230462;31.01.2020;Planung;CHS2;AccountType;ENTITLEMENT_BALANCE_01;Value;295,50
 A;1230462;31.01.2020;Ist;CHS2;AccountType;VM_SALDO_GESAMT;Value;3,25
 A;1230462;31.01.2020;Ist;CHS2;AccountType;SALDO_GESAMT;Value;8,25
 A;1230462;31.01.2020;Ist;CHS2;AccountType;SOLL_MONAT;Value;168,00
 A;1230462;31.01.2020;Ist;CHS2;AccountType;PLUSMINUS;Value;5,00
 A;1230462;31.01.2020;Ist;CHS2;AccountType;TOTAL;Value;173,00
 A;1230462;31.01.2020;Ist;CHS2;AccountType;ENTITLEMENT_BALANCE_01;Value;295,50

Export von Planungscodes

Planungscodes können getrennt vom Rest exportiert werden.

Der Export wird als Differenzexport ausgeführt, es werden immer nur die Veränderungen zum zuletzt ausgeführten Export angeführt. Die Vorgehensweise ist unter [Planungscode Differenz Export](#) beschrieben.

Der Export kann manuell über Systemeinstellungen -> "SAP/BI Planungscodes-Export ausführen" angestoßen werden.



Der tägliche automatische Export wird über den Parameter "Planungscodes: automatisch exportieren" aktiviert.

File-Aufbau

Alternativ kann der Fileaufbau von SAP auf IPIS gewechselt werden. (siehe [Parametrierung](#)).

ACHTUNG: Trennzeichen ";" zwischen den Feldern

Feldname:	Kurzzeichen:	Beschreibung:
Mandant	MANDT	Alphanum. Bezeichnung (für jeden Mandanten wird ein eigenes Export-File erstellt)
Logisches System	LOGSYS	AWTYP + MANDT
Referenzvorgang	AWTYP	ALEX
Referenzschlüssel	AWKEY	Eindeutige ID der Export-Zeile (wird zum stornieren benötigt)
Stornokennzeichen	STOKZ	X
Personalnummer	PERNR	Alphanum. Personalnummer
Infotyp	INFYT	2001
Subtyp	SUBTY	Planungscode: Abw-Kennzeichen für Export
Gültigkeitsbeginn	BEGDA	TT.MM.JJJJ Beginn-Datum der Abwesenheit
Gültigkeitsende	ENDDA	TT.MM.JJJJ Ende-Datum der Abwesenheit (siehe Anmerkung unterhalb)

Anmerkung zu Gültigkeitsende:

- Liegt die Abwesenheit nicht vollständig in der Vergangenheit, so wird als das Datum 31.12.9999 eingetragen.
 - Ist beim betreffenden Mitarbeiter bereits eine Inaktivzeit oder ein Austritt erfasst, so wird stattdessen das Datum der letzten Stammzuteilung eingetragen.

Übersicht Exporte

Eine Übersicht der bisherigen Exporte kann über den Navigatorpunkt **"SAP/BI Export Planungscode"** aufgerufen werden. Hier werden alle bisherigen Exporte aufgelistet und können bei Bedarf mittels Rechtsklick und "SAP/BI Planungscode-Export ausführen" nochmals exportiert werden.

Navigator: SAP/BI Export Planungscode

Navigator

- Projekt-Gruppe
- Protokoll
- Qualifikation
- Qualifikationsgruppe
- Qualifikationskreis
- SAP/BI Export Planungscode**
- Saldospezifikation
- Schema Sammlung
- Schemaanpassung(Basis)

Neuigkeiten rund um ALEX unser...

SAP/BI Export Planungscode für...

Abgefragter Typ: SAP/BI Export Planungscode

Ansicht: SAPBIExportPlanSymbolsResultView

Suchbegriff:

erzeugt am	Differenz zu
21.10.2022 11:56:22.35	
21.10.2022 12:00:16.58	203370116
25.10.2022 10:16:51.91	203370122 Differenz zu: 203370116
25.10.2022 10:19:55.03	203370274 Differenz zu: 203370122
22.11.2022 8:43:33.92	203370374 Differenz zu: 203370274

- 203370642 Differenz zu: 203370374 bearbeiten
- Löschen umschalten
- Eigenschaft importieren
- SAP/BI Planungscode-Export ausführen**

Parametrierung

Systemeinstellungen SAP/BI Export

Mandant	123
Verzeichnis	
Dateierweiterung	
von	01.01.2023
bis	31.01.2023
Konten Tage	
Konten Monate	
Beginn zur Stunde	0
Schema für Nachtdienste	
Planungscodes: automatisch exportieren	Nein
Verzeichnis Planungscode	D:\BitFactory\AlexDat\Projekte\Exporte_
Dateierweiterung Planungscode	txt
IPIS: Format verwenden	Nein
IPIS: Property für "LV Bedienernummer für Schnittstelle"	
IPIS: Property für "LV Standort für Schnittstelle"	

Mandant: Pro Mandant wird ein eigenes Export-File erzeugt, die Files werden jeweils mit dem Mandantenkürzel vorne benannt (z. B. 123_DayAccounts.csv)

Verzeichnis: Pfad, an dem die Export-Files abgelegt werden

Dateierweiterung: Gewünschter Dateityp (txt, csv,...)

von / bis: für die Reports DayAccounts und MonthAccounts kann ein fixer Von-Bis-Zeitraum eingestellt werden (ansonsten werden diese beiden Reports jeweils vom 1. des Jahres bis zum Ultimo des aktuellen Monats geliefert).

Konten Tage: hier kann eine Spaltenansicht hinterlegt werden, die enthaltenen Konten werden dem Report "Mandant_DayAccounts" angefügt

Konten Monate: hier kann eine Spaltenansicht hinterlegt werden, die enthaltenen Konten werden dem Report "Mandant_MonthAccounts" angefügt

Beginn zur Stunde: Uhrzeit für den automatischen täglichen Export

Schema für Nachtdienste: Der Report "Shift" weist Nachtdienste mit einem "N" aus, dazu kann hier eine Schemasammlung für die relevanten Nachtdienstschemen hinterlegt werden

Planungscodes: automatisch exportieren: Aktiviert den täglichen automatischen Export

Verzeichnis Planungscodes: Pfad an dem die Export-Files für die Planungscodes abgelegt werden

Dateierweiterung Planungscodes: Gewünschter Dateityp (txt, csv,...) für das File der Planungscodes

IPIS: Format verwenden: Ändert das Export-Format von SAP auf IPIS. Folgende zwei Parameter müssen zusätzlich gesetzt werden:

IPIS Format Parametrierung		
IPIS: Format verwenden	☒	Ja
IPIS: Property für 'LV Bedienernummer für Schnittstelle'		<u>LV Bedienernummer für Schnittstelle</u>
IPIS: Property für 'LV Standort für Schnittstelle'		<u>LV Standort für Schnittstelle</u>

IPIS: Property für 'LV Bedienernummer für Schnittstelle': Bei Verwendung des IPIS-Formats muss hier das Property "LV Bedienernummer für Schnittstelle" hinterlegt werden:

IPIS: Property für 'LV Standort für Schnittstelle': Bei Verwendung des IPIS-Formats muss hier das Property "LV Standort für Schnittstelle" hinterlegt werden:

Anwendungsbeispiel Krankenstands-Dashboard

[Krankenstands-Dashboard](#)

Bit Factory interne Zusatzinfos

[Aktivierung](#)

Export Tagesinformationen

Für diese Schnittstelle muss das Modul "Standard ReportExport DayIntervals" eingespielt werden. Die Module "Basis" und "Basic Procedures" müssen ebenfalls vorhanden sein. Eventuell kann noch zusätzlich das [Bfx.Alex.AccountsReport.DayInterval.Model.dll](#) zur vereinfachten Inbetriebnahme verwendet werden.

Es gibt auch eine [Variante als Web-Service](#), welche vorrangig empfohlen wird, da die hier beschriebene File-Schnittstelle nicht so flexibel adaptierbar ist wie das Web-Service.

Satzbeschreibung der Datei: Export Tagesinformationen

Für Data-Warehouse-Systeme, Planungstools wie Therapieplaner oder ähnliche Werkzeuge werden Detailinformationen pro Mitarbeiter und Tag wie folgt bereitgestellt:

ACHTUNG: Trennzeichen ";" zwischen den Feldern

Feldname:	Feldlänge:	Feldformat:
Personalnummer	var	Alphanum. Personalnummer
Name	var	optional: Alphanum. Name (muss in Systemeinstellungen aktiviert werden)
Vorname	var	optional: Alphanum. Vorname (muss in Systemeinstellungen aktiviert werden)
e-mail-Adresse	var	optional: Alphanum. e-mail-Adresse (muss in Systemeinstellungen aktiviert werden)
Stamplan	var	optional: Alphanum. Bezeichnung der Stamplanungsseinheit (muss in Systemeinstellungen aktiviert werden)
Stamplan-Kurz	var	optional: Alphanum. Kurzzeichen der Stamplanungsseinheit (muss in Systemeinstellungen aktiviert werden)
Für Tag	10	TT.MM.JJJJ Tagesdatum
Berufsgruppe	var	optional: Alphanum. Bezeichnung der Berufsgruppe (muss in Systemeinstellungen aktiviert werden)
Berufsgruppe-Kurz	var	optional: Alphanum. Kurzzeichen der Berufsgruppe (muss in Systemeinstellungen aktiviert werden)
Tagestext	var	optional: Alphanum. Text, welcher am Tag erfasst wurde (muss in Systemeinstellungen aktiviert werden)
Tagesabwesenheit	var	Abwesenheitscode (Url, Krank,...) optional: 'Plansymbol für Export' statt Plansymbol
Bezeichnung Abw.	var	optional: Bezeichnung der Abwesenheit (muss in Systemeinstellungen aktiviert werden)
Dienstkürzel	var	optional: Dienstkürzel vom ersten Tagdienst (muss in Systemeinstellungen aktiviert werden)
Dienstbezeichnung	var	optional: Bezeichnung vom ersten Dienst (muss in Systemeinstellungen aktiviert werden)
Hintergrundfarbe	var	optional: Hintergrundfarbe vom ersten Tagdienst (muss in Systemeinstellungen aktiviert werden)
Textfarbe	var	optional: Textfarbe vom ersten Tagdienst (muss in Systemeinstellungen aktiviert werden)
Aufgabe	var	optional: Bezeichnung der Aufgabe (muss in Systemeinstellungen aktiviert werden)
Planstunden	var	#x,xx (2 Nachkomma)
Iststunden	var	#x,xx (2 Nachkomma)
beliebige Anzahl pro Tag:		
Von Tag/Uhrzeit	16	TT.MM.JJJJ HH:MM Beginn-Tageszeit
Bis Tag/Uhrzeit	16	TT.MM.JJJJ HH:MM Ende-Tageszeit
Dienstkürzel	var	optional: Dienstkürzel für diese Kommt-Geht-Zeit (muss in Systemeinstellungen aktiviert werden)
Planungscode	var	Arzt, Besprechung, Krank, Fortbildung

Beispiele:

```
004711;04.01.2016;;8,00;8,00;04.01.2016 7:30;04.01.2016 12:00;;04.01.2016 12:30;
04.01.2016 16:00;
004712;04.01.2016;;8,00;9,00;04.01.2016 7:30;04.01.2016 12:00;;04.01.2016 12:30;
04.01.2016 16:00;04.01.2016 16:00;04.01.2016 17:00;iFB;
004713;04.01.2016;URL;8,00;8,00;
004714;04.01.2016;URL;8,00;6,00;
1234;Test;Claudia;alex@bitfactory.at;Urologie;UR0;10.04.2017;;;T/V;Tagdienst+verlängerter
Dienst (23,50 Std.);#FFFFFF;#000000;;23,50;23,82
1234;Test;Claudia;alex@bitfactory.at;Urologie;UR0;24.05.2017;ZA genehmigt am 07.12.2016;
ZA;Zeitausgleich (Dienstnehmer);;;;4,80;4,80
```

Zeile 1: Mitarbeiter mit Personalnummer 004711 hat geplant 8.00 Stunden und im Ist 8.00 Std. gearbeitet, von 7:30 - 12:00 und 12:30 - 16:00 Uhr
 Zeile 2: Mitarbeiter mit Personalnummer 004712 wie 004711 aber von 16:00 bis 17:00 Uhr eine 'Interne Fortbildung', daher im Ist 9,00 Stunden
 Zeile 3: Mitarbeiter mit Personalnummer 004713 hat Urlaub den ganzen Tag, Vollzeitkraft daher 8.00 Stunden
 Zeile 4: Mitarbeiter mit Personalnummer 004714 wie 004713 aber Teilzeitkraft mit 75%, daher nur 6.00 Stunden

Einstellungsmöglichkeiten

Unter Systemeinstellungen/Exporteinstellungen können noch folgende Parameter gesetzt werden:

- Report-Export Tage mit Dienst 1 -> es wird auch das Dienstkürzel vom 1. Dienst mitgeschickt (wenn der Dienst ein Kombinationsdienst ist, so werden das 1. und 2. Plansymbol mit einem "/" verbunden (z.B. T/RB))
- Report-Export Tage mit Dienst-Bezeichnung -> es wird auch die Bezeichnung vom Dienst sowie die Bezeichnung der Tagesabwesenheit mitgeschickt
- Report-Export Tage mit Dienst pro K/G -> es wird für jeden Kommt-Geht-Eintrag das Dienstkürzel mitgeschickt
- Report-Export Tage mit Tagtext -> es wird auch der Text, welcher am Tag erfasst wurde, mitgeschickt
- Report-Export mit Stammplan -> es wird die Stammplanungseinheit des Mitarbeiters mitgeschickt
- Report-Export mit Stammplan-Kurzzeichen -> es wird das Kurzzeichen der Stammplanungseinheit des Mitarbeiters mitgeschickt
- Report-Export mit Farbmarkierung -> es werden Aufgaben mitgeschickt
- Report-Export mit Personalnamen -> es werden Namen und Vornamen mitgeschickt
- Report-Export mit Personal-Email -> es wird die E-Mail-Adresse mitgeschickt
- Report-Export mit Dienst-Farbinformationen -> es werden die Nummern der Hinter- und Textfarbe des Dienstes mitgeschickt
- Report-Export mit Berufsgruppe -> es werden die Bezeichnung und das Kurzzeichen der Berufsgruppe mitgeschickt
- Report-Export mit 'Plansymbol für Export' -> es wird das am Planungscode hinterlegte Kennzeichen 'Plansymbol für Export' statt dem Plansymbol verwendet

Konto-Report anlegen:

Falls die [Bfx.Alex.AccountsReport.DayInterval.Model.dll](#) verwendet wird, dann muss kein Konto-Report angelegt werden. Ansonsten:

Für die Schnittstelle muss ein Konto-Report angelegt werden. Im Konto-Report müssen keine Konten eingefügt werden, es müssen lediglich folgende zwei Einstellungen getroffen werden:

- bei "Exporteinstellungen auswählen" : "Report-Export Tage" wählen
- bei Zeitmaßstab "Tag" auswählen

Dieser Konto-Report kann dann als Datei gespeichert werden, indem man bei Export einen Dateinamen + Pfad festlegt.

Besonderheit: Automatisch (täglich) starten:

Mittels eines Automatischen Kontoreports kann diese Schnittstelle täglich im Hintergrund automatisch erstellt werden. Details finden Sie im Kapitel [Automatischer Kontoreport](#).

Externe Anspruchsverwaltung

Inhaltsverzeichnis

1 Allgemeines	58
2 Einmalige Einstellungen	58
2.1 System	58
2.2 Benutzer	58
2.3 Sonderabrechnung	59
3 Monatlicher Ablauf	59
4 Funktion deaktivieren für einzelne Berufsgruppen	59

Allgemeines

Bei externer Anspruchsverwaltung werden die Ansprüche (z.B. Urlaub) nicht in ALEX[®] verwaltet, sondern in einem externen Programm.

Da für die Dienstplanung der Restanspruch jedoch von Bedeutung ist, können die Reststände in ALEX[®] importiert werden.

Folgende Einstellungen sind zu treffen:

- Modul "Externe Anspruchsverwaltung" importieren
- Kein Entitlement.dll
- Standardmäßig Urlaub in Tagen --> wenn Urlaub in Stunden importiert werden soll, dann ist ein Eingriff im Kundenmodul notwendig

Hier gehts zum [Standardsatzaufbau für die Externe Anspruchsverwaltung](#)

Folgende Konten stehen bei externer Anspruchsverwaltung zur Verfügung:

- Urlaub Vormonats Rest
- Urlaub in Stunden / Tagen
- Urlaub Rest

Einmalige Einstellungen

System

Bei den Systemparametern muss angegeben werden, ob der Urlaub in Tagen oder Stunden importiert wird.

Zusätzlich ist beim "Basiskonto für Urlaubskonsum" das Abrechnungsschema des Urlaubsanspruchs hinterlegt werden.

Systemparameter bearbeiten	
Allgemein	Verhalten bei Planungscode-Eintragung
Erweiterte Organisation	7Up Standeskontrolle: Mehrzeit anzeigen ab [Minuten]
Abrechnungsgruppenspezifikation	Basiskonto für Urlaubskonsum ABW U
Extraspezifikationen	Urlaub in Tage? Ja
Schemaspezifikationen	

Benutzer

Der Import muss während eines Shutdowns durchgeführt werden, damit zu dieser Zeit niemand im Dienstplan arbeitet und so einen Mitarbeiter sperrt.

Damit der Benutzer, welcher die Urlaube importiert, während des Shutdowns einsteigen kann, muss auf diesem Benutzer folgender Parameter auf JA gesetzt sein:

User sperren	
Anmelden während Sperrung	Ja

ACHTUNG! Diesen Parameter darf nur 1 Benutzer im gesamten System hinterlegt haben!!!

Daher wird hier oftmals ein eigener Benutzer z.B. "Urlaubsimport" angelegt.

Sonderabrechnung

Mit Hilfe der Sonderabrechnung "Korr. Urlaubsanspruch" können Sie importierte Urlaube auch im Dienstplan manuell korrigieren.

Die Sonderabrechnung muss dafür jedoch am Bereich unter "Verwendete Sonderabrechnungen" hinterlegt werden:

Monatlicher Ablauf

- 1) Hinterlegung eines Shutdowns
- 2) Einstieg mit User "Urlaubsimport"
- 3) Anlegen eines neuen Imports (erst während der Shutdownzeit): Navigator --> Import Anspruch --> grünes Plus (immer neuen Import anlegen)

- Bezeichnung: Hier vergeben Sie eine beliebige Bezeichnung für den Import. z.B. Jänner 2014
- Importdatei für Restanpassung: Hier wählen Sie die zu importierende Datei aus.

Anschließend bestätigen Sie die Eingaben mit "OK" in der Werkzeugleiste und speichern die Änderungen.

Funktion deaktivieren für einzelne Berufsgruppen

Wenn für bestimmte Berufsgruppen (z. B. Praktikanten) nie ein Urlaubsrest übermittelt wird, kann für diese Berufsgruppe die Funktion deaktiviert werden.

ALEX® sucht nämlich bei jedem Mitarbeiter zur Berechnung vom Konto "Urlaub Rest" die letzte Korrektur vom Urlaubsrest.

Wird bei einem Mitarbeiter keine Urlaubskorrektur gefunden kann es dadurch zu Performanceproblemen kommen.

Deshalb kann die Funktion mit dem Parameter "Externe Anspruchsverwaltung: deaktivieren" ausgeschaltet werden.

Systemeinstellungen: Funktion deaktivieren

Berufsgruppe: Praktikant

Arbeitszeitgesetzprüfungen

Erweiterte Organisation

Fehlermarkierung

Externe Anspruchsverwaltung: deaktivieren



Ja

Externe Anspruchsverwaltung: Arbeitsjahr?



Ja

Externe Anspruchsverwaltung: Startmonat

01.01.1900



Import Anspruch

Weiterleitung nach:

- [Import Anspruch anlegen](#)

Import Einsatzplan

Import von Poolzuteilungen über den "Classic Import". Entspricht der EASY-SOFT Schnittstelle.

Dateiorganisation	sequentielle Datei im CSV-Format, variableSatzlänge
Pfad und Dateiname	wird bei Programmstart abgefragt
Restriktionen	keine
Trennzeichen	keines
Besonderheiten	Achtung! immer am Ende eine Leerzeile --> sonst wird letzter MA nicht importiert!!

Feldinhalt	Länge	Typ	Beschreibung
Personalnummer	8	Text	Eindeutige Personalnummer, die auch in anderen Personalsystemen verwendet wird (z.B. Lohnverrechnung)
Pooleinsatz	1	Text	Konstante "X" für 'Ja'
Leerzeichen	73	blanks	Leerzeichen einfügen
Einsatzbeginn	8	Datum	JJJJMMTT
Einsatzende	8	Datum	JJJJMMTT
Planungseinheit	5	Text	Dienstplaneinheit (Abteilung, Station, ...) lt. "Kurzzeichen" in den Alex®- Stammdaten

Beispiel

```
433456 X
2014100120150930INT_A
451122 X
2015081720150930INT_B
987776 X
2015050420150719CH_I
998877 X
2015072020150816CH_I
```

Zu Beachten

Bei Einspielung der Importdatei werden immer alle aktuellen Pooleinsätze überschrieben. Die Daten jener Pläne die noch im Status "Ist" sind, können folglich durch den Import verändert werden. Daher ist es empfohlen, alle aktuellen und zukünftigen Pooleinsätze zu importieren.

Beispiel: April 2017 im Ist -> es wurden Pooleinsätze ab Oktober 2017 eingespielt -> bestehende Pooleinsätze von April bis Oktober wurden herausgelöscht, da diese nicht in der Importdatei vorhanden waren

Import Monatswerte

technische Voraussetzungen

- Modul "Import_Extra_Monat.BfxMODULES" muss importiert sein (Zusätzlich muss auch das Modul Basic_Procedures importiert sein)
- zu importierende Datei (aus Fremdsystem) muss vorhanden sein --> [Satzaufbau für Import Monatswerte](#)
 - pro Sonderabrechnung muss es eine eigene Importdatei geben

Ablaufbeschreibung

- Einstieg in ALEX mit einem Benutzer mit Supervisor-Berechtigung
Es ist möglich, eine Anmelde-Sperre für ALEX (="Shutdown") einzugeben, um sicherzustellen, dass kein Personal durch andere ALEX-Benutzer "gesperrt" ist.
In diesem Fall muss bei dem ALEX-Benutzer, mit dem der Import ausgeführt werden soll, der Parameter "Einstieg während Sperre" einmalig auf JA gesetzt werden.
- im Navigator bei "Import Monatswerte" auf "Grünes Plus" klicken
 - "Bezeichnung"
freier Text
 - "Importdatei"
hier Dateipfad auswählen (muss zur nachfolgend gewählten Sonderabrechnung passen)
 - "Monats-Sonderabrechnung"
gewünschte Monats-Sonderabrechnung auswählen
 - "Datum für Monats-Sonderabrechnung"
hier das Datum des Monatsplans eingeben, auf den die Werte gesetzt werden sollen
 - Hinweis: Pläne müssen zum gewählten Zeitpunkt im Status "nicht erzeugt", "Planung" oder "Ist" befinden.
Sollte versucht werden, einen Datensatz für ein Personal einzuspielen, dessen Plan sich bereits im Status "Abgeschlossen" befindet, so kommt ein Hinweis im Import-Protokoll, dass dieser Datensatz nicht importiert werden konnte.
 - "Import-Protokoll"
Hier ist nichts zu tun - dieses Feld wird nach Klick auf "Ausführen" automatisch befüllt.
 - "Ausführen"
 - Neben "Import-Protokoll" muss der Text "Import OK" angezeigt werden, anderenfalls ist ein Problem aufgetreten.
(Das Import-Protokoll wird auch als Text-Datei in jenem Ordner abgelegt, in dem sich die zuvor gewählte Importdatei befindet.)
- Änderungen können bei Bedarf sofort in den Monatsplänen kontrolliert werden.
- "Speichern" bzw. "Alles verwerfen" (alle Änderungen durch den Import werden wieder verworfen)

Beispiel für erfolgreichen Import:

Import Monatswerte: Test

Import Monatswerte: Test

Bezeichnung	Test
Importdatei	Z:\ImportF10\F10_201912.txt
Monats-Sonderabrechnung	<u>Einspringer F10 (für 'Import Monatswerte')</u>
Datum für Monats-Sonderabrechnung	01.12.2019
Import-Protokoll	Import OK

Beispiel für Import-Protokoll Datei:

> Dieser PC > AlexDat (\\nas\bfx) (Z:) > ImportF10			
Name	Änderungsdatum	Typ	
F10_201912.txt	06.02.2020 09:10	TXT-Datei	
F10_201912.txt_Import-Protokoll_06-02-2020_9-15-32-46.txt	06.02.2020 09:15	TXT-Datei	

... diese Protokolldatei wird immer neu erzeugt. Bei einem fehlerfreien Import ist diese leer (0 KB).

Import Personal

Weiterleitung nach:

- [ASCII-Standardsatzaufbau für den Personalimport](#)

Import Zeitstempelungen

Stempelungen müssen nicht zwingend über die ALEX[®]-Stempeluhr erfasst werden.

Für den Import von anderen Zeiterfassungssystemen gibt es aktuell fünf unterschiedliche Import-Schnittstellen:

- ALEX[®]-Standard
- ALEX[®]-Standard Semicolon
- Gantner
- EMVS
- Timecom

Inhaltsverzeichnis

1 Satzbeschreibung der Zeitstempel-Importschnittstelle Format "Standard"	67
1.1 Muster-Beispiele	67
2 Satzbeschreibung der Zeitstempel-Importschnittstelle Format "Standard Semicolon"	67
2.1 Muster-Beispiele	68
3 Parametrierung	68
3.1 Systemeinstellungen	68
3.2 Sonstige	69
4 Weiterführende Links	69

Satzbeschreibung der Zeitstempel-Importschnittstelle Format "Standard"

Die Dateiendung muss ".exp" lauten. Die Datei muss mit der Codierung "ANSI" abgespeichert werden.

Satzbeschreibung bei Zeitstempel-Import Format = "Standard"

Feldname	Feldlänge	Feldformat	Entspricht im Alex:
Nr	variabel	alphanum., Länge ist einstellbar (default=20)	Personal - Kartenummer
DatumZeit	16	JJJJ-MM-TT HH:MM	Datum und Zeit der Stempelung
Type	1	numerisch (nur 0 oder 1 möglich)	0 bedeutet "von"-Stempelung, 1 bedeutet "bis"-Stempelung
ABWCode	5	alphanum, z.B.: AZ für Arztbesuch oder B für Besprechung	Planungscode-Zeiterfassung/Stempeluhr - Plansymbol für Import; wenn leer -> Planungscode -Plansymbol
Text	80	alphanum (optionaler freiwählbarer Text)	Kommt-Geht-Text
PlanCode	20	alphanum	Planungseinheit für Kommt-Geht (Externes Kennzeichen, sonst Kurzzeichen)

Muster-Beispiele

So muss der Satz in der Schnittstelle aussehen, wenn der Mitarbeiter...

... um 9:00 zum normalen Dienst gekommen ist und als Text "hallo" erfasst hat:

```

4711          2006-01-01 09:000      hallo
1234567890123456789012345678901234567890123456789012345678901234567890 ( -> nur
Hilfestellung - nicht in die Schnittstelle schreiben )
1          2          3          4          5          6          7          8          ( -> nur
Hilfestellung - nicht in die Schnittstelle schreiben )

```

... um 9:00 vom Arzt gekommen ist:

```

4711          2006-01-01 09:000AZ
1234567890123456789012345678901234567890123456789012345678901234567890
1          2          3          4          5          6          7          8

```

... um 9:00 zur Besprechung gegangen ist:

```

4711          2006-01-01 09:001B
1234567890123456789012345678901234567890123456789012345678901234567890
1          2          3          4          5          6          7          8

```

... um 9:00 nach Hause gegangen ist:

```

4711          2006-01-01 09:001
1234567890123456789012345678901234567890123456789012345678901234567890
1          2          3          4          5          6          7          8

```

Satzbeschreibung der Zeitstempel-Importschnittstelle Format "Standard Semicolon"

Die Dateiendung muss ".exp" lauten. Die Datei muss mit der Codierung "ANSI" abgespeichert werden. Die Felder werden durch einen Strichpunkt ";" getrennt

Satzbeschreibung bei Zeitstempel-Import Format = "Standard semicolon"

Feldname	Feldformat	Entspricht im Alex:
Nr	alphanum.	Personal - Kartenummer
DatumZeit	JJJJ-MM-TT HH:MM	Datum und Zeit der Stempelung
Type	numerisch (nur 0 oder 1 möglich)	0 bedeutet "von"-Stempelung, 1 bedeutet "bis"-Stempelung
ABWCode	alphanum, z.B.: AZ für Arztbesuch oder B für Besprechung	Planungscode-Zeiterfassung/Stempeluhr - Plansymbol für Import; wenn leer -> Planungscode -Plansymbol
Text	alphanum (optionaler freiwählbarer Text)	Kommt-Geht-Text
PlanCode	alphanum	Planungseinheit für Kommt-Geht (Externes Kennzeichen, sonst Kurzzeichen)
ColorMark	alphanum	Aufgabe für Kommt-Geht (Object ID)

Muster-Beispiele

So muss der Satz in der Schnittstelle aussehen, wenn der Mitarbeiter...

... um 9:00 zum normalen Dienst gekommen ist und als Text "hallo" erfasst hat:

```
4711;2006-01-01 09:00;0;;hallo
```

... um 9:00 vom Arzt gekommen ist:

```
4711;2006-01-01 09:00;0;AZ
```

... um 9:00 zur Besprechung gegangen ist:

```
4711;2006-01-01 09:00;1;B
```

... um 9:00 nach Hause gegangen ist:

```
4711;2006-01-01 09:00;1
```

Parametrierung

Systemeinstellungen

In **Systemeinstellungen - Zeiterfassung/Stempeluhr** sind folgende Parameter zu setzen:

ALEX[®]-Parametrierung

Zeitstempel-Import Format	Standard	▼
Zeitstempel-Import Pfad	\\SERVER\StempelimportOrdner 	
Zeitstempel-Import Häufigkeit (120)[sec]		120
Zeitstempel-Import Kartenummer Anzahl der Stellen(20)		20
Zeiterfassung: 'ComputationAgent' Intervall (180)[min]		

- Zeitstempel-Import-Format zur Auswahl des Formats:
 - Standard
 - Standard Semicolon
 - Gantner
 - EMVS
 - Timecom
- Zeitstempel-Import Pfad: Pfad hinterlegen, wo die Import-Files von der Stempeluhr abgelegt werden.
 - Achtung! Der Server bzw. der Benutzer (Windowsbenutzer), mit welchem der ALEX[®]-Dienst ausgeführt wird, muss auf diesem Pfad eine Schreib-Berechtigung haben, wenn sich der Ordner nicht am Server befindet.
- Zeitstempel-Import Häufigkeit: Wie oft (in Sekunden) sollen die Stempel importiert werden?
- Zeitstempel-Import Kartenummer Anzahl der Stellen: Default-Wert ist 20. Dieses Feld ist für das "Standard"-Format zu befüllen, falls die Länge dieses Feldes angepasst werden soll.
- Zeiterfassung 'ComputationAgent' Intervall (180)[min]:
 - Standardmäßig werden die Zeitkonten des Personals alle 180 Minuten neu durchgerechnet.
 - Der Gesetz Wert wird nur bei aktiver Zeiterfassung herangezogen.

Sonstige

- Personalstammdaten
 - Kartenummer
 - Stempelungen verwenden = JA
- Bei Bedarf auch:
- Planungscode
 - Plansymbol für Import
 - div. andere Einstellungen zur Regelung, wie sich der Planungscode beim Stempeln verhält ([mehr dazu hier](#))
- Planungseinheit
 - Externes Kennzeichen
- Aufgabe
 - Kurzzeichen

Weiterführende Links

[Aktivierung](#)

Lohnverrechnungsschnittstelle

Mit der Lohnverrechnungsschnittstelle werden ausgewählte variable Werte in ein Schnittstellenfile geschrieben, welches in die [externe Lohnverrechnung übergeleitet](#) werden kann.

Aufgrund der unterschiedlichen Personalsysteme wird die Schnittstelle individuell an die Anforderungen angepasst.

Im Schnittstellenfile werden zumindest folgende Details angeführt:

- [Personalnummer](#)
- [Lohnart](#)
- [Zeitkonto](#)

Die Schnittstellen für die Personalverrechnung (Lohnarten und Abwesenheiten) sind lizenzpflichtig, d. h. optional erhältlich.

Kann eine Schnittstelle aus technischen Gründen nicht erzeugt werden, so ist eine Kontrolle der [Lohnarten](#) wie folgt möglich:

- [Lohnarteninformation](#)
- [Lohnartenliste](#)

Inhaltsverzeichnis

1 Erstellen einer Lohnverrechnungsschnittstelle	71
2 Differenz zu bestehender Lohnverrechnungsschnittstelle	72
3 Parametrierung	72
4 Bit Factory interne Zusatzinfos	72

Erstellen einer Lohnverrechnungsschnittstelle

Voraussetzungen:

- Dienstplan befindet sich im Status 'Abgeschlossen'
- Benutzer verfügt über ausreichende Berechtigungen
- Pfad wurde am System hinterlegt
- Lohnverrechnungsmodul wurde eingespielt, d.h. die Anforderungen an die Schnittstelle wurden bereits abgeklärt

Um eine Lohnartenliste zu erstellen, wird im Navigator "Lohnartenexport anlegen" ausgewählt. In erster Linie ist nun die Auswahl der gewünschten Ressource relevant:

- System: wird keine Auswahl getroffen, so wird das Schnittstellenfile für das gesamte Haus erstellt
- Bereich
- Planungseinheit
- Mitarbeiter

Durch Auswahl des Symbols "Lohnarten in Datei exportieren" in der Multifunktionsleiste wird nun die Schnittstelle erzeugt und am angegebenen Pfad abgelegt.

Der Name der Datei beinhaltet immer das Erstellungsdatum sowie den Namen des Benutzers.

Mittels der Suchfunktion "Lohnartenexport für Entstehungsmonat" ist ein einfacher Überblick möglich, welche Schnittstellen oder [Lohnartenliste](#) erzeugt wurden. Folgende Informationen werden angeführt:

- Ressource
- Abrechnungsmonat
- Wann wurde die Liste von welchem Benutzer erstellt?
- Sollte ein Schnittstelle erstellt worden sein: Wann wurde eine Schnittstelle von welchem Benutzer erstellt?

Entstehungsmonat	Abrechnungsmonat	Erstellung	Exportierung
01.09.2008	01.09.2008	09.09.2008 11:24:57.45, SUPERVISOR	

In der Statusübersicht wird angeführt, für welche Planungseinheit bereits eine Schnittstelle erzeugt wurde.

!!!ACHTUNG!!!

Wird die Schnittstelle mit einem Doppelklick und Auswahl von "Lohnarten in Datei exportieren" geöffnet, so kann die Datei unter Umständen andere Daten enthalten.

In diesem Fall werden die Daten neu aus dem Dienstplan bezogen - es wird ein neuer Datenauszug erstellt. Wurde der Dienstplan verändert, so enthält auch die Schnittstelle andere Daten als bisher.

Differenz zu bestehender Lohnverrechnungsschnittstelle

Wird, wie oben erklärt, der Dienstplan im Status 'Ist' nochmals verändert, so besteht die Möglichkeit, eine neue Schnittstelle zu erzeugen, die NUR die Differenz zur Originalschnittstelle beinhaltet.

Die Schnittstelle wird erneut im Navigator mit "Lohnartenexport anlegen" erstellt. Diesmal ist unter "Differenz zu Export" die ursprüngliche Schnittstelle auszuwählen.

Durch Auswahl des Symbols "Lohnarten in Datei exportieren" in der Multifunktionsleiste wird nun die neue Schnittstelle abgespeichert.

In dieser Schnittstelle ist nun ersichtlich, welche Auswirkung die Änderung des Dienstplanes auf die variablen Bezüge hat. In der oberen, ersten, Grafik werden 3 Nachtdienste sowie 16 Sonntagsstunden angeführt. In der neu erstellten "Differenzschnittstelle" werden die Sonntagsstunden nicht mehr angeführt, da es hier keine Änderungen gab. In der Schnittstelle ist ersichtlich, dass der Mitarbeiter einen zusätzlichen Nachtdienst leistete (4 anstatt der vorhin eingetragenen 3 Nachtdienste).

Parametrierung

Voraussetzung ist, dass das entsprechende Modul eingespielt wurde.

In den Systemeinstellungen / Exporteinstellungen können unter anderem folgende Parameter eingestellt werden:

- Pfad für Lohnartenexport: dort wird dann das File gespeichert.
- Dateierweiterung für Lohnartenexport: Achtung, manche Schnittstellen haben dies ausprogrammiert, dann wird dieser Parameter zur Gänze ignoriert! Die Dateierweiterung sollte OHNE Punkt eingegeben werden.
- **Datei-Präfix für Lohnartenexport:** Zum Definieren der Anfangsbuchstaben des Filenamens
- Pfad für Abwesenheitsexport: dort wird dann das File gespeichert.
- Dateierweiterung für Abwesenheitsexport: Achtung, manche Schnittstellen haben dies ausprogrammiert, dann wird dieser Parameter zur Gänze ignoriert! Die Dateierweiterung sollte OHNE Punkt eingegeben werden.
- **Datei-Präfix für Abwesenheitsexport:** Zum Definieren der Anfangsbuchstaben des Filenamens

Bit Factory interne Zusatzinfos

- [Details zu den Schnittstellen](#)

Lohnüberleitung

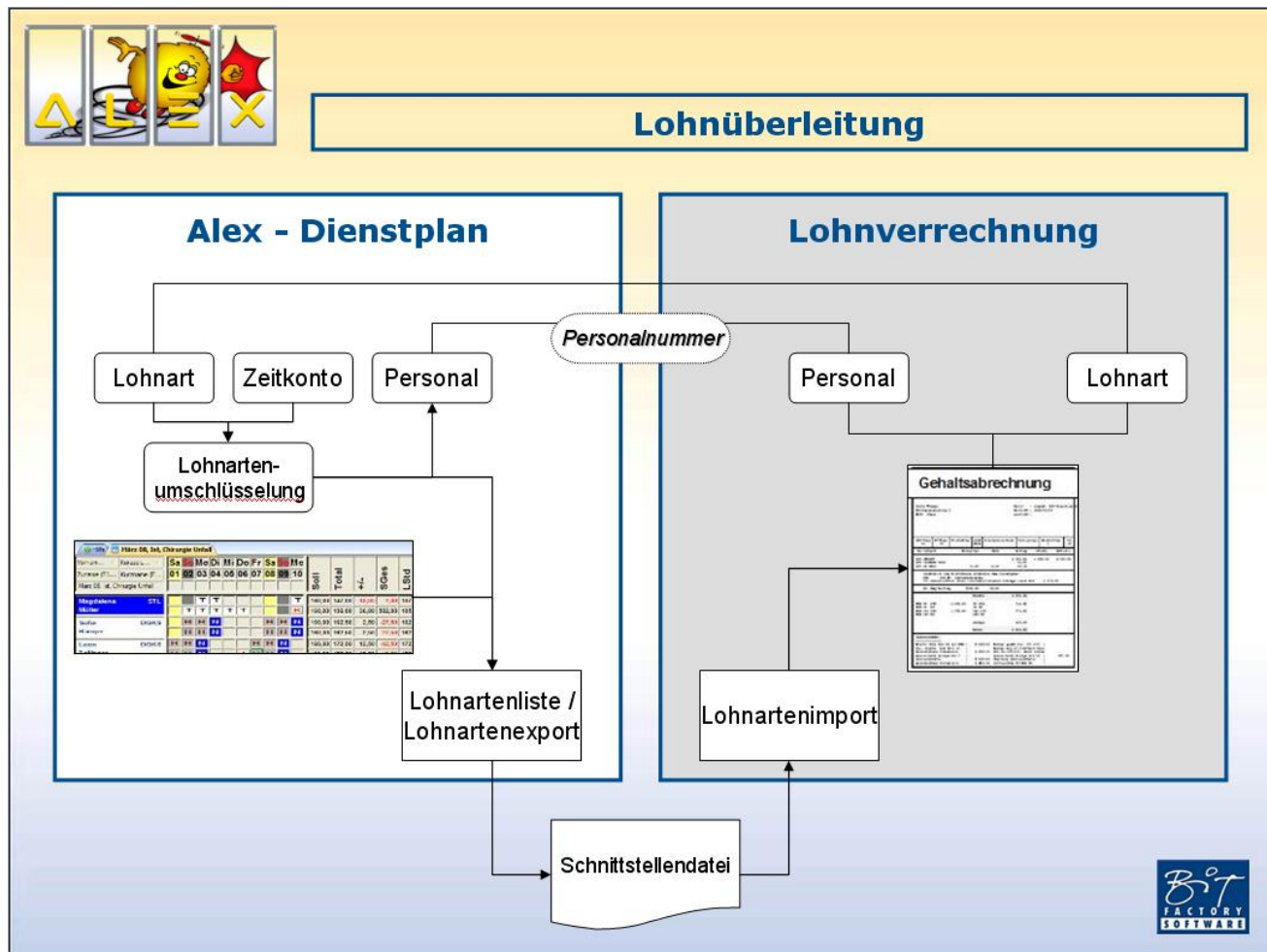
Die **Lohnüberleitung** ermittelt die variablen Bezüge eines Mitarbeiters in einem Monat und stellt diese optional einer **externen Lohnverrechnung** zur weiteren Verarbeitung zur Verfügung.

Die variablen Bezüge sind Teile des monatlichen Bruttogehalts eines Mitarbeiters, die durch spezielle Arbeitsleistungen entstehen.

Beispiele hierfür sind:

Sonntagszulagen Nachtdienstzulagen Überstundenzuschläge

Alex kann eine Vielzahl derartiger variabler Bezüge ermitteln.



Inhaltsverzeichnis

1 Betriebswirtschaftliche Zielsetzung	75
2 Umsetzung in Alex	75
2.1 Einstellungen	75
2.1.1 Lohnarten	75
2.1.2 Lohnartenumschlüsselung	75
2.1.3 Personal	75
2.1.4 Lohnverrechnungsschnittstelle	75
2.2 Operative Maßnahmen	75
2.2.1 Lohnartenliste / Lohnartenexport	75
2.2.2 Übernahme in Lohnverrechnungssystem	75

2.3 Kontrollmöglichkeiten	76
2.3.1 Monatsplan	76
2.3.2 Lohnverrechnungsschnittstelle	76
3 Schwierigkeiten	76
3.1 Handhabung	76
3.2 Fehlermöglichkeiten	76
4 Wechselwirkungen	76
5 Privilegienverwaltung	76
6 Vollständigkeit und geplanter Ausbau	77

Betriebswirtschaftliche Zielsetzung

Die durch die Dienstplanung und der hier integrierten Ist-Zeit-Erfassung gesammelten Werte sollen automatisch in die [Lohnverrechnung](#) überführt werden.

Umsetzung in Alex

Einstellungen

Folgende Einstellungen sind zu treffen, um eine Überleitung in die [externe Lohnverrechnung](#) zu ermöglichen:

Lohnarten

Alle für die [externe Lohnverrechnung](#) relevanten [Lohnarten](#) werden in Alex angelegt. Existieren für ein [Zeitkonto](#) mehrere [Lohnarten](#), so sind alle zu definieren.

Die [Lohnarten](#) werden in weiterer Folge in der Lohnüberleitung mit den [Zeitkonten](#) verknüpft.

Lohnartenumschlüsselung

Nachdem die [Lohnarten](#) festgelegt sind, die Alex in die [externe Lohnverrechnung](#) übernehmen soll, wird die Verbindung zu den [Zeitkonten](#) mithilfe der [Lohnartenumschlüsselung](#) hergestellt.

Für jede Mitarbeitergruppe, die die abzurechnenden Bezüge mit einer anderen [Lohnart](#) bereitgestellt bekommt, ist eine eigene [Lohnartenumschlüsselung](#) anzulegen.

Personal

Im [Personalstamm](#) sind folgende Eingaben nötig:

- die Personalnummer: zur eindeutigen Identifikation des Mitarbeiters in ALEX und der [externen Lohnverrechnung](#)
- die [Lohnartenumschlüsselung](#): als Information dafür, welches [Zeitkonto](#) mit welcher [Lohnart](#) übergeleitet werden soll

Lohnverrechnungsschnittstelle

Um eine automatische Überleitung der variablen Bezüge zu ermöglichen, wird ein individuell angepasstes Modul in Alex eingespielt.

Mit diesem Modul werden die im Vorfeld definierten Anforderungen an die [Lohnverrechnungsschnittstelle](#) umgesetzt.

Operative Maßnahmen

Lohnartenliste / Lohnartenexport

Nachdem alle oben angeführten Einstellungen getroffen wurden, kann nun eine [Lohnverrechnungsschnittstelle](#) erzeugt werden.

Besteht die Möglichkeit einer automatischen Überleitung der variablen Bezüge nicht, so steht nun eine [Lohnartenliste](#) zur Verfügung.

Übernahme in Lohnverrechnungssystem

Die erzeugte [Lohnverrechnungsschnittstelle](#) kann in weiterer Folge in das verwendete [Lohnverrechnungssystem](#) eingespielt werden.

Kontrollmöglichkeiten

Monatsplan

Die bei den einzelnen Mitarbeitern entstandenen variablen Bezüge und die damit verbundenen [Lohnarten](#) können direkt im Monatsplan mittels Kontextmenü '[Info Lohnarten](#)' überprüft werden.

Zur Kontrolle der [Lohnarten](#) der gesamten Planungseinheit steht ebenfalls eine [Lohnarteninformation](#) zur Verfügung. Die Auswahl erfolgt über "Druckvorschau".

Lohnverrechnungsschnittstelle

Im [Schnittstellenfile](#) werden alle Informationen der ausgewählten Ressourcen angeführt.

Schwierigkeiten

Handhabung

- Bei einem untermonatigen Stationswechsel werden die Lohnarten jener Planungseinheit zugeordnet, auf die der Mitarbeiter mit Monatsersten zugeteilt war. Welche [Lohnarten](#) übergeleitet werden, kann mit Hilfe der [Lohnarteninformation](#) kontrolliert werden.
- Eine [Lohnverrechnungsschnittstelle](#) sowie eine [Lohnartenliste](#) können nur im Status 'Abgeschlossen' erzeugt werden. Die [Lohnarteninformation](#) im Monatsplan steht jedoch auch in 'Planung' und 'Ist' zur Verfügung. Achtung: Bei [Wochenplanung](#) muss die ganze letzte, hinaushängende Woche abgeschlossen werden, damit die Lohnarten exportiert werden können.
- ALEX führt sowohl Gesamtüberleitungen als auch Rollungen durch. Wird das [Schnittstellenfile](#) für ein Monat ein zweites Mal erzeugt, so beinhaltet das File entweder alle abzurechnenden Bezüge des ausgewählten Bereichs oder nur die Differenz zur ersten Schnittstelle. Nähere Informationen in Bezug auf die Differenzlogik finden Sie unter [Lohnverrechnungsschnittstelle](#) oder [Lohnartenliste](#).

Fehlermöglichkeiten

- Wird das [Schnittstellenfile](#) für ein Monat ein zweites Mal erzeugt und in das [Lohnverrechnungssystem](#) eingespielt, so werden entweder alle abzurechnenden Bezüge des ausgewählten Bereichs angeführt oder nur die Differenz. Es ist darauf zu achten, dass bei Erstellung der [Lohnverrechnungsschnittstelle](#) die korrekte Auswahl getroffen wird. Weiters ist zu berücksichtigen, dass die [Externe_Lohnverrechnung](#) die Daten korrekt verarbeiten kann.
- Bei Erstellung des [Schnittstellenfiles](#) ist auf den korrekten Dateinamen zu achten. Nur so kann gewährleistet werden, dass das korrekte File in die [externe Lohnverrechnung](#) übernommen wird.
- Wechselt ein Mitarbeiter untermonatig die Station - und beide Stationen haben unterschiedliche Bereiche - so kann es beim Erstellen der [Lohnartenliste](#) zu einem Fehler kommen: Werden die Bereiche einzeln übergeleitet, so werden die variablen Bezüge des Mitarbeiters auf beiden Listen angeführt. Er erhält also seine Zulagen doppelt. Werden alle Bereiche gesammelt übergeleitet, kommt es zu keinen Problemen.

Wechselwirkungen

- Die Erstellung einer [Lohnverrechnungsschnittstelle](#) ist nur im Status 'Abgeschlossen' möglich. Um einen Plan in diesen Status zu setzen, muss die [Statustreppe](#) eingehalten werden.
- Analog zu den [Lohnarten](#) können auch alle Abwesenheiten mittels [Abwesenheitsschnittstelle](#) in das verwendete Personalsystem übernommen werden.

Privilegienverwaltung

Um eine [Lohnverrechnungsschnittstelle](#) erzeugen zu können, sind bestimmte Privilegien nötig. Alle Benutzer mit den Rechten "Supervisor" oder der standardisierten Rolle "Lohnverrechner" können eine Schnittstelle erzeugen. Nähere Informationen sind unter [Privilegienverwaltung](#) zu finden.

Vollständigkeit und geplanter Ausbau

Der Prozess ist im Sinne der Zielsetzung noch nicht vollständig automatisiert. Bei gegebener technischer Voraussetzung auf Seite der Lohnverrechnungssoftware kann zukünftig die Kommunikation ohne manuellen Eingriff erfolgen.

Präfix für Lohnarten- und Abwesenheitsschnittstelle

Das Präfix im Namen der Lohn- und Abwesenheitsdatei muss nicht händisch abgeändert werden, sondern kann auch im Programm festgelegt werden.

Zentrale Anforderung

Damit die Einspielung der Lohn- und Abwesenheitsdatei problemlos und ohne weiterer händischer Bearbeitung funktioniert, sollen die Anfangsbuchstaben des Dateinamens im Programm einstellbar sein.

Funktionsbeschreibung/Parametrierung

In der Gruppe Exporteinstellungen kann man unter folgenden Parametern die gewünschten Präfixe definieren:

- Datei-Präfix für Lohnartenexport
- Datei-Präfix für Abwesenheitsexport

Wird nichts hinterlegt, dann werden die üblichen Präfixe LA_ und PS_ genommen.

Links

- [Lohnverrechnungsschnittstelle](#)

Schnittstellen

Wird ALEX[®] im Verbund mit anderen HR-Informationssystemen betrieben, erfolgt der Datenaustausch über Schnittstellen. Jedes dieser Systeme benötigt Personalstammdaten und produziert im Rahmen der eigenen Arbeitsprozesse Daten, die von anderen Systemen (zum Teil) zur Weiterverarbeitung benötigt werden.

Inhaltsverzeichnis

1 HR-Informationssysteme	80
2 Datenaustausch mit Alex [®]	80
3 Personalstammdaten	80
3.1 Schnittmenge	80
3.2 Differenzmenge	81
3.3 Datensynchronisation	81
3.3.1 Fremdschlüssel	81
3.3.2 Stichtagsbezogene Bereitstellung	81
3.3.3 Auslösung von Aufrollungen	81
4 Technologische Umsetzung	82
5 Betriebswirtschaftliche Betrachtung	82

HR-Informationssysteme

Im HR-Management können unterschiedliche IT-Systeme zur Anwendung kommen:

- ALEX[®]-Dienstplanung
- Elektronische Zeiterfassung (als externes System; kann auch von ALEX[®] als integriertes Modul verwendet werden)
- Lohn- und Gehaltsabrechnung
- Personalinformation
- Business-Intelligence Systeme
- Schulverwaltung (z. B. für Pflegeschüler)
- Aus- und Weiterbildungsmanagement
- Dokumentationsmanagement
- Mobile Einsatzplanung mit Klientenabrechnung
- Pflegedokumentation
- usw.

Datenaustausch mit ALEX[®]

Jedes System kann aufgrund seiner Spezialisierung ein anderes System mit Daten versorgen:

- ALEX[®] liefert monatlich variable Bezüge (Zulagen, Zuschläge) für die Lohnverrechnung
- ALEX[®] liefert monatliche Abwesenheiten für die Lohnverrechnung und/oder das Personalinformationssystem
- ALEX[®] liefert geplante Arbeitszeiten und Abwesenheiten für die mobile Einsatzplanung
- ALEX[®] liefert Arbeitszeiten und erhält Aus- und Weiterbildungsdaten im Rahmen dieser Arbeitszeit
- Ein externes Lohnverrechnungs- oder Personalinfo-System liefert Personalstammdaten
- Ein externes Zeiterfassungssystem liefert Zeitstempelungen an ALEX[®]
- Eine Pflegeschulverwaltung liefert Einsatzzeiträume und Ferienzeiten von Pflegeschülern
- Ein Personalinformationssystem liefert die jährlichen Urlaubsansprüche an ALEX[®]

Personalstammdaten

Jedes HR-System benötigt Personalstammdaten, wobei die Schnittmenge nur einen kleinen Teil ausmacht. Der wesentlich größere Teil ist die Differenzmenge, also jene speziellen Personalstammdaten, die nur im eigenen System Bedeutung haben. Bei den Personalstammdaten sprechen wir von 'Eigenschaften des Mitarbeiters'.

Schnittmenge

Die Schnittmenge ist der kleinere Teil der Personalstammdaten, wie

- Personalnummer
- Vorname
- Zuname
- Geburtsdatum
- Eintrittsdatum
- E-Mail-Adresse

wobei die Personalnummer als eindeutige Identifikation systemübergreifend zur Anwendung kommt.

Differenzmenge

Jedes System benötigt seine eigenen Personalstammdaten, die im Fremdsystem völlig unbedeutend sind (Sozialversicherungs- oder Lohnsteuerschlüssel sind ausschließlich für das Lohnverrechnungssystem selbst relevant). ALEX[®] weist folgende Differenzmenge an Personalstammdaten auf, um die Planungs- und Abrechnungsprozesse abbilden zu können:

- Beschäftigungsausmaß in Prozent
- Beschäftigungsausmaß pro Tag (siehe [Sollstundenberechnung](#))
- Berufsgruppe
- Qualifikationen mit/ohne Bewertung
- Stammpfanungseinheit (Abteilung)
- Bewertungskriterien (siehe [Abrechnung Österreich](#) / [Abrechnung Deutschland](#))
- [Lohnartenumschlüsselung](#)
- Rahmenplanmitarbeiter
- Passwort für Selbstbedienungsbenutzer
- Urlaubsanspruch (mit Zusatzanspruch für Invalidität, Stichtage für Staffelung bei höheren Ansprüchen)
- Wochenzeitmodelle
- uvm.

Datensynchronisation

Bei der zentralen Verwaltung der Stammdaten zur automatischen Synchronisation sind die im folgenden beschriebenen Aspekte zu berücksichtigen:

Fremdschlüssel

Einige Eigenschaften der Differenzmenge sind ALEX[®]-Fremdschlüssel:

- Berufsgruppe
- Stammpfanungseinheit
- Beschäftigungsausmaß, falls mit Sollverträgen gearbeitet wird
- [Lohnartenumschlüsselung](#)

Diese Keys müssen ebenfalls synchronisiert werden, um die eigentlichen Personalstammdaten zentral verwalten zu können.

Stichtagsbezogene Bereitstellung

ALEX[®] verwaltet die Mitarbeitereigenschaften fast ausschließlich (Ausnahme: Vorname, Geburtstag) stichtagsbezogen, da der Stichtag (Monatserster, ggf. auch untermonatig) maßgebend die Prozesse beeinflusst.

Beispiel:

Stammpfanungseinheit ab 15.11. nicht mehr "Abteilung A" sondern "Abteilung B" --> der Mitarbeiter steht der Dienstplanverantwortlichen der Abteilung "B" zur Verfügung

Beschäftigungsausmaß ab 01.12. nicht mehr "100%" sondern "75%" --> ab 01.12. werden sämtliche Berechnungen **SOFORT** mit Eigenschaftsänderung korrigiert.

Auslösung von Aufrollungen

Mit Ausnahme von Personalnummer und Name können alle Eigenschaften eine unmittelbare Auswirkung auf die Berechnung haben:

- Geburtstag: AZG-Prüfungen

- Geschlecht: Schwerarbeitsverordnung
- Eintritt: Sollstunden, Anspruch
- usw.

Durch Erfassen dieser Eigenschaften in Fremdsystemen können Erfassungsfehler nicht sofort erkannt werden; bei Erfassung dieser Daten im Dienstplanformular ergibt sich eine sofortige Plausibilitätsprüfung der erfassten Situation durch den Planer.

Eine Korrektur von Eigenschaften im Fremdsystem führt beim automatischen Import zu Problemen, da möglicherweise die [Statustreppe](#) verletzt wird und Dokumente mit falschen Ergebnissen in Umlauf sind.

Technologische Umsetzung

ALEX[®] liefert eine Reihe von Import- und Export-Schnittstellen, wobei derzeit zwei unterschiedliche Technologien zur Anwendung kommen:

- im ASCII-Format
- als Webservice

Der Großteil des Datenaustausches erfolgt derzeit in Form von sequentiellen Files - mittelfristig werden alle Schnittstellen auch als Webservice zur Verfügung stehen:

- **Personalstammdaten:**
 - [Import von Personalstammdaten \(Webservice\)](#)
 - [Import von Personalstammdaten - altes Format bis 2014](#)
 - [Import von Personalstammdaten \(einmalig beim Systemaufbau\)](#)
 - [Export von Personalstammdaten \(Webservice\) - ALEX ist führendes Personalsystem](#)
- [Import von Anspruchsdaten \(altes Format\)](#)
- [Import von Restansprüchen bei externen Anspruchsverwaltung](#)
- [Import von speziellen Monatswerten](#)
- [Import von Zeitstempelungen externer Zeiterfassungsterminals](#)
- [Import von Schüler-Einsatzplänen \(Krankenpflegeschüler\)](#)
- [Import von Krankenstandsdaten vom SV-Träger](#)
- [Export von variablen Bezügen \(Zuschläge, Zulagen, ...\)](#)
- [Export von Abwesenheiten \(Urlaube, Krankenstände, ...\)](#)
- [Export von Tagesinformationen \(Arbeitszeiten von/bis, Abwesenheiten\)](#)

Eine automatische Übermittlung der variablen Bezüge per Mail ist bereits verfügbar.

Betriebswirtschaftliche Betrachtung

Der Einsatz einer Schnittstelle ist nur dann zielführend, wenn keine manuellen Nachbearbeitungen erfolgen müssen. So erzielen beispielsweise folgende Schnittstellen eine sehr hohe Wertschöpfung:

- [Export von variablen Bezügen \(Zuschläge, Zulagen, ...\)](#)
- [Export von Abwesenheiten \(Urlaube, Krankenstände, ...\)](#)
- [Import von Zeitstempelungen externer Zeiterfassungsterminals](#)
- [Import von Schüler-Einsatzplänen \(Krankenpflegeschüler\)](#)
- Kommunikation mit Lernmanagement-Systemen

Einen kritischen Blick MUSS man auf die Inbetriebnahme einer Personalstammdatenschnittstelle werfen:

Jede Eigenschaft, die NICHT oder FALSCH elektronisch zur Verfügung gestellt wird, erfordert eine manuelle Nachbearbeitung.
Es ist somit zu prüfen, ob die Daten so zur Verfügung gestellt werden können, dass kein manueller Eingriff erforderlich ist.

Folgende Überlegungen müssen bei der Entscheidungsfindung über die Inbetriebnahme berücksichtigt werden:

- Zu welchem Zeitpunkt wird der Mitarbeiter im Gesamtsystem benötigt? (ALEX[®]: zum Planungszeitpunkt, Lohnverrechnung: zum 'Eintrittstag')
- Wie lange dauert die Anlage von Neuzugängen? (ALEX[®]: ca. 1 Minute pro MA --> bei 10 Neuzugängen im Monat: 10 Minuten Arbeitszeit).
- Wie lange dauert die Änderung von Mitarbeiterstammdaten? (welche sind das, wie oft)
- Rechtfertigt das Ergebnis den (organisatorischen / technologischen / kommerziellen) Aufwand?

Stundennachweis PDF anzeigen über Webservice

Das [WebUI-Webservice](#) - **Stundennachweis PDF** ist eine [Programmfunktion](#) zur Abfrage des Stundennachweis pro Personal und Datum im Dateiformat PDF.

Inhaltsverzeichnis

1 Ausgangssituation	85
1.1 Voraussetzungen	85
1.2 Formatierung der Anfrage	85
1.2.1 Schema für URL-Kodierung	85

Ausgangssituation

Voraussetzungen

Es müssen die Voraussetzungen für den Betrieb des [WebUI-Webservice](#) erfüllt sein sein.

Formatierung der Anfrage

Die Daten werden dem Webservice **in der URL kodiert** als **GET**-Argumentübertragung übergeben.

Ein Datum wird immer als übertragen.

Schema für URL-Kodierung

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

```
/AlexPrintTimeSheet  
?id=  
[&date=]
```

Parameter	Wertebereich	Beschreibung
id	Zeichenkette	Eine Personalnummer
date	Datum	Datum des Monats des gewünschten Stundennachweises, falls nicht angegeben Heute

Webservice

Ein **Webservice** ist eine [Programmfunktion](#) zur Bereitstellung von Schnittstellen über das HTTP-Protokoll.

Voraussetzungen

Um die Programmfunktion nutzen zu können

- muss der [Webservice-Daemon](#) konfiguriert sein.
- muss ein gültiger [WebApi-Port](#) konfiguriert sein.
- kann ein Zertifikat zur [Verschlüsselung](#) verwendet werden.
- kann ein [Sicherer Link-Aufruf](#) konfiguriert werden.

Das Webservice stellt dann Zugriffspfade an der konfigurierten [Webservice-Adresse](#) zur Verfügung.

Der Bediener des Webservices muss mit dem **HTTP-Protokoll** und seinen **Möglichkeiten zur Argumentübertragung** ^{[1][2]} vertraut sein.

Der Bediener muss das Format der Anfrage und Ausgabe kennen.

http-client-example

[Live-Demo](#)

[Sourcecode](#)

Dieses Beispielprogramm in HTML und JavaScript dient zur Veranschaulichung der Bedienung eines ALEX[®]-Webservice.

Mit diesem Beispielprogramm kann

- ein HTTP-GET Request an ein Webservice gesendet werden (Body-Eingabefeld bleibt leer)
- ein HTTP-POST Request an ein Webservice gesendet werden (Body-Eingabefeld ist gefüllt)
- ein API-Key gesendet werden

Im Response-Ausgabefeld wird das Ergebnis des Aufrufs angezeigt.

Mit diesem Beispielprogramm können alle ALEX[®]-Webservices bedient werden.

http-client-example

API-Key: 92834g8f-942e-231e-8943-308c2147790db

URL: http://localhost:4401/Set

Body:

```
{
  "objectType": "Employee",
  "endQuery": "Employee200",
  "reportType": "EmployeeEmailAddress",
  "lines": [
    {
      "watchString": "a",
      "valueString": "jürgit.bauer@bitfactory.at"
    }
  ]
}
```

Send

Response

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

Fußnoten

1. [↑ https://www.w3schools.com/tags/ref_httpmethods.asp](https://www.w3schools.com/tags/ref_httpmethods.asp)
2. [↑ https://de.wikipedia.org/wiki/Hypertext_Transfer_Protocol#Argument%C3%BCbertragung](https://de.wikipedia.org/wiki/Hypertext_Transfer_Protocol#Argument%C3%BCbertragung)

Webservice Personalstammdaten Export

Das [Webservice](#) - **Personalstammdaten Export** ist eine [Programmfunktion](#) zur Abfrage von Personalstammdaten für einen Zeitbereich.

Es kann dazu verwendet werden die Personalstammdaten Fremdsystemen zur maschinellen Weiterverarbeitung zur Verfügung zu stellen.

Personal welches mit **Löschen umschalten** versteckt wurde ist nicht im Ergebnis enthalten.

Unter `Systemeinstellungen: WebApiServer Einstellungen: zu exportierende Personalstamm-Eigenschaften` können zusätzliche Personal-Eigenschaften in einer Ansicht gesammelt werden, welche auch exportiert werden. ^[1]

Inhaltsverzeichnis

1 Ausgangssituation	88
1.1 Voraussetzungen	88
1.2 Formatierung der Anfrage	88
1.2.1 Schema für URL-Kodierung	88
1.3 Formatierung der Ausgabe	88
2 Beispielanfragen	89
3 Beispielausgabe	90
4 Fußnoten	91

Ausgangssituation

Voraussetzungen

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

Das Webservice stellt dann folgende Zugriffspfade zur Verfügung:

- `/GetEmployees`

Formatierung der Anfrage

Die Daten werden dem Webservice **in der URL kodiert** als **GET**-Argumentübertragung übergeben.

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:

```
/GetEmployees  
[?from=]  
[&end=]
```

Parameter	Wertebereich	Beschreibung
<code>from</code>	Datum	Beginn des abzufragenden Zeitbereichs, falls nicht angegeben minus unendlich
<code>end</code>	Datum	Ende des abzufragenden Zeitbereichs, falls nicht angegeben plus unendlich

Formatierung der Ausgabe

Personalstammdaten im [JSON-Format](#) UTF8 codiert. Das JSON-Objekt hat folgendes [JSON-Schema](#):

JSON-Schema Datei

```
{  
  "$schema": "https://json-schema.org/draft/2020-12/schema",  
  "title": "GetEmployees",  
  "description": "format for the GetEmployees-Webservice",  
  "type": "object",  
  "properties": {  
    "Employees": {  
      "type": "array",  
      "description": "Array der Personen",  
      "items": {  
        "type": "object",  
        "properties": {  
          "EmployeeID": {  
            "type": "string",  
            "description": "Personalnummer"  
          },  
          "Surname": {  
            "type": "string",  
            "description": "Zuname"  
          },  
          "Forename": {  
            "type": "string",  
            "description": "Vorname"  
          },  
          "DayOfBirth": {  
            "type": "string",  
            "description": "Geburstag"  
          }  
        }  
      }  
    }  
  }  
}
```



```

    },
    "EmployeeSignature": {
      "type": "string",
      "description": "Handzeichen für Pflegedoku"
    },
    "Sex": {
      "type": "string",
      "description": "Geschlecht"
    },
    "EmployeeJobGroup": {
      "type": "string",
      "description": "Berufsgruppe (externe Referenz)"
    },
    "MasterAllocation": {
      "type": "string",
      "description": "Stammplanungseinheit (externe Referenz)"
    },
    "EmployeeQualifications": {
      "type": "array",
      "description": "Array der Qualifikationen",
      "items": {
        "type": "string",
        "description": "Qualifikation (externe Referenz)"
      }
    },
    "EmployeeEmailAddress": {
      "type": "string",
      "description": "Email"
    },
    "EmployeePhoneNumber": {
      "type": "string",
      "description": "Telefonnummer"
    },
    "EmployeeSocialInsurance": {
      "type": "string",
      "description": "Sozialversicherungsnummer"
    },
    "EmployeeStreet": {
      "type": "string",
      "description": "Straße"
    },
    "EmployeeCity": {
      "type": "string",
      "description": "Stadt"
    },
    "PostalCode": {
      "type": "string",
      "description": "Postleitzahl"
    },
    "ForNation": {
      "type": "string",
      "description": "Land"
    },
    "EmployeeLastEmployedFrom": {
      "type": "string",
      "description": "Eintritt"
    },
    "EmployeeLastEmployedTo": {
      "type": "string",
      "description": "Austritt"
    }
  }
}

```

Beispielanfragen

Beschreibung	URL
gesamte Personalstammdaten	/GetEmployees

Beschreibung	URL
Personalstammdaten von 05.12.2017 bis einschließlich 07.12.2017	/GetEmployees?id=20&from=05.12.2017 &end=07.12.2017

Beispielausgabe

JSON-Datei einer Ausgabe

```
{
  "Employees": [
    {
      "EmployeeID": "123",
      "Surname": "Mayr",
      "Forename": "Max",
      "DayOfBirth": "01.01.1981",
      "EmployeeSignature": "MYM",
      "Sex": "männlich",
      "MasterAllocation": "S1",
      "EmployeeJobGroup": "Berufsgruppe1",
      "EmployeeQualifications": [
        "SP",
        "V"
      ],
      "EmployeeEmailAddress": "mayr@mail.com",
      "EmployeePhoneNumber": "07252 / 82 68 5",
      "EmployeeSocialInsurance": "1234",
      "EmployeeStreet": "Gleinkergasse 50",
      "EmployeeCity": "Steyr",
      "PostalCode": "4400",
      "ForNation": "Österreich",
      "EmployeeLastEmployedFrom": "01.08.2020",
      "EmployeeLastEmployedTo": "31.12.2099"
    },
    {
      "EmployeeID": "345",
      "Surname": "Huber",
      "Forename": "Franz",
      "DayOfBirth": "01.02.1983",
      "EmployeeSignature": "HUF",
      "Sex": "männlich",
      "MasterAllocation": "S1",
      "EmployeeJobGroup": "Berufsgruppe1",
      "EmployeeQualifications": [
        "SP"
      ],
      "EmployeeEmailAddress": "huber@mail.com",
      "EmployeePhoneNumber": "07252 / 4400 12345",
      "EmployeeSocialInsurance": "4321",
      "EmployeeStreet": "Fischergasse 22",
      "EmployeeCity": "Steyr",
      "PostalCode": "4400",
      "ForNation": "Österreich",
      "EmployeeLastEmployedFrom": "01.08.2020",
      "EmployeeLastEmployedTo": "31.12.2099"
    },
    {
      "EmployeeID": "567",
      "Surname": "Müller",
      "Forename": "Hilde",
      "DayOfBirth": "09.03.1985",
      "EmployeeSignature": "MUH",
      "Sex": "weiblich",
      "MasterAllocation": "S2",
      "EmployeeJobGroup": "Berufsgruppe1",
      "EmployeeQualifications": [
        "V"
      ],
      "EmployeeEmailAddress": "mueller@mail.com",
      "EmployeePhoneNumber": "0732 / 4010 98765",
      "EmployeeSocialInsurance": "9999",

```

```
"EmployeeStreet": "Kirchengasse 57",  
"EmployeeCity": "Linz",  
"PostalCode": "4010",  
"ForNation": "Österreich",  
"EmployeeLastEmployedFrom": "01.08.2020",  
"EmployeeLastEmployedTo": "31.12.2099"  
  }  
}  
}
```

Fußnoten

1. [↑ #7697](#)

Webservice Tagesinformationen

Das **Webservice - Tagesinformationen** ist eine **Programmfunktion** zur Abfrage von Tagesinformationen pro Personal und Zeitbereich.

Es kann dazu verwendet werden die Tagesinformationen für einen Zeitbereich und Personal Fremdsystemen zur maschinellen Weiterverarbeitung zur Verfügung zu stellen.

Inhaltsverzeichnis

1 Ausgangssituation	93
1.1 Voraussetzungen	93
1.2 Formatierung der Anfrage	93
1.2.1 Schema für URL-Kodierung	93
1.3 Formatierung der Ausgabe	93
2 Beispiele	97
2.1 Beispiel 1: gesamtes Personal	97
2.1.1 Ausgangssituation	97
2.1.2 Aufruf der Funktion	98
2.1.3 Ergebnis	98
2.1.4 Datenbank für Beispiel	101
2.2 Beispiel 2: Personal mit Personalnummer	101
2.2.1 Ausgangssituation	101
2.2.2 Aufruf der Funktion	102
2.2.3 Ergebnis	102
2.2.4 Datenbank für Beispiel	103
2.3 Beispiel 3: Personal auf Plan	103
2.3.1 Ausgangssituation	103
2.3.2 Aufruf der Funktion	103
2.3.3 Ergebnis	104
2.3.4 Datenbank für Beispiel	106
2.4 Beispiel 4: Beispiel Personal auf Plan und Personal mit Personalnummer	107
2.4.1 Ausgangssituation	107
2.4.2 Aufruf der Funktion	107
2.4.3 Ergebnis	107
2.4.4 Datenbank für Beispiel	110
2.5 Beispiel 5: Alle Status	110
2.5.1 Ausgangssituation	110
2.5.2 Aufruf der Funktion	110
2.5.3 Ergebnis	110
2.5.4 Datenbank für Beispiel	112
3 Beispielanfragen	112
4 Beispielausgabe	112
5 Fußnoten	114

Ausgangssituation

Voraussetzungen

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

Das Webservice stellt dann folgende Zugriffspfade zur Verfügung:

- `/EmployeeDayIntervals`

Formatierung der Anfrage

Die Daten werden dem Webservice **in der URL kodiert** als **GET**-Argumentübertragung übergeben.

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:

```
/EmployeeDayIntervals
[?allEmployees=]
[&id=]
[&planCode=]
[&resourcesBson=]
[&begin=]
[&end=]
[&allstates=]
```

Parameter	Wertebereich	Beschreibung
<code>allEmployees</code>	<code>true</code> oder <code>false</code>	Wenn <code>true</code> , gesamtes Personal im Zeitbereich
<code>id</code>	Zeichenkette	Eine Personalnummer
<code>planCode</code>	Zeichenkette	Kurzcode eines Plans
<code>resourcesBson</code>	Zeichenkette	JSON-String: eine Zusammenfassung von Plan Kurzzeichen und Personalnummern zum Beispiel: <code>resourcesBson={"employeeIDs":["E1","E2","E3"],"planCodes":["P1","P2","P3"]}</code>
<code>begin</code>	Datum	Beginn des abzufragenden Zeitbereichs, falls nicht angegeben Heute
<code>end</code>	Datum	Ende des abzufragenden Zeitbereichs, falls nicht angegeben Morgen
<code>allstates</code>	<code>true</code> oder <code>false</code>	Wenn <code>true</code> , Statusdaten für alle aktuellen Status: Planung -> Planung, Ist -> Planung und Ist

Formatierung der Ausgabe

Tagesinformation im [JSON-Format](#) UTF8 codiert. Das JSON-Objekt hat folgendes [JSON-Schema](#):

JSON-Schema Datei

```
{
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "title": "EmployeeDayIntervals",
  "description": "format for the EmployeeDayIntervals-Webservice",
  "type": "object",
  "$defs": {
    "planSymbolObjectType": {
      "type": "object",
      "description": "Plansymbol",

```

```

    "properties": {
      "PlanSymbolCode": {
        "type": "string",
        "description": "Kurzcode des Planungscodes"
      },
      "PlanSymbolCodeExt": {
        "type": "string",
        "description": "Externer Kurzcode des Planungscodes"
      },
      "PlanSymbolType": {
        "type": "string",
        "description": "Typ des Planungscodes 'Anwesend' oder 'Abwesend'"
      },
      "Label": {
        "type": "string",
        "description": "Bezeichnung des Planungscodes"
      }
    },
    "shiftObjectType": {
      "type": "object",
      "description": "Dienst",
      "properties": {
        "Label": {
          "type": "string",
          "description": "Bezeichnung des Dienstes"
        },
        "ShiftAnnotation": {
          "type": "string",
          "description": "Anmerkungen"
        }
      }
    },
    "extraObjectType": {
      "type": "object",
      "properties": {
        "Key": {
          "type": "string",
          "description": "Name der Sonderabrechnung"
        },
        "Label": {
          "type": "string",
          "description": "Bezeichnung der Sonderabrechnung"
        },
        "Value": {
          "type": "string",
          "description": "Wert der Sonderabrechnung"
        }
      }
    },
    "planObjectType": {
      "type": "object",
      "description": "Plan",
      "properties": {
        "PlanCode": {
          "type": "string",
          "description": "Kurzcode des Plans"
        }
      }
    },
    "properties": {
      "Employees": {
        "type": "array",
        "description": "Array der Personen für die Daten gefunden wurden",
        "items": {
          "type": "object",
          "properties": {
            "EmployeeID": {
              "type": "string",
              "description": "Personalnummer"
            },
            "Surname": {
              "type": "string",
              "description": "Zuname"
            }
          }
        }
      }
    }
  }

```

wurden",

Planungscodes mit Typ 'Abwesend')"

```

},
"Forename": {
  "type": "string",
  "description": "Vorname"
},
"EmployeeTitleBeforeName": {
  "type": "string",
  "description": "Titel vor Name"
},
"EmployeeTitleAfterName": {
  "type": "string",
  "description": "Titel nach Name"
},
"MasterAllocation": {
  "type": "string",
  "description": "Stammplanungseinheit"
},
"MasterAllocationObject": {
  "$ref": "#/$defs/planObjectType"
},
"EmployeeJobGroup": {
  "type": "string",
  "description": "Berufsgruppe"
},
"Days": {
  "type": "array",
  "description": "Array aller Tage für die Daten gefunden
wurden",
  "items": {
    "type": "object",
    "properties": {
      "ForDate": {
        "type": "string",
        "description": "Datum des Tages"
      },
      "ForState": {
        "type": "string",
        "description": "Status des Tages"
      },
      "TextCategory": {
        "type": "string",
        "description": "Textkategorie des Tages"
      },
      "InfoText": {
        "type": "string",
        "description": "Tagtext"
      },
      "DayPlanSymbol": {
        "type": "string",
        "description": "Abwesenheit auf Tag (nur
Planungscodes mit Typ 'Abwesend')"
      },
      "DayPlanSymbolObject": {
        "$ref": "#/$defs/planSymbolObjectType"
      },
      "DayShift1": {
        "type": "string",
        "description": "1. Dienst am Tag"
      },
      "DayShift1Object": {
        "$ref": "#/$defs/shiftObjectType"
      },
      "DayShift2": {
        "type": "string",
        "description": "2. Dienst am Tag"
      },
      "DayShift2Object": {
        "$ref": "#/$defs/shiftObjectType"
      },
      "DayRequestKind": {
        "type": "string",
        "description": "Aufgabe am Tag"
      },
      "DayEvents": {

```

```

        "type": "array",
        "description": "zusätzliche Aufgaben am Tag",
        "items": {
            "type": "object",
            "properties": {
                "Label": {
                    "type": "string",
                    "description": "Bezeichnung der
Aufgabe"
                }
            }
        },
        "ForeignShift1": {
            "type": "string",
            "description": "1. Dienst auf anderer
Planungseinheit: Bezeichnung von Planungseinheit"
        },
        "ForeignShift2": {
            "type": "string",
            "description": "2. Dienst auf anderer
Planungseinheit: Bezeichnung von Planungseinheit"
        },
        "DayTotalMinutesComputed": {
            "type": "string",
            "description": "Total in Minuten aus Status"
        },
        "DayPoolAllocations": {
            "type": "array",
            "description": "Array aller Poolzuteilungen an
diesem Tag",
            "items": {
                "type": "string",
                "description": "Bezeichnung von
Planungseinheit"
            }
        },
        "AssignedExtra": {
            "type": "array",
            "description": "Array aller Sonderabrechnung am
Tag",
            "items": {
                "$ref": "#/$defs/extraObjectType"
            }
        },
        "DayIntervals": {
            "type": "array",
            "description": "Array aller Kommt/Geht-Paare
am Tag",
            "items": {
                "type": "object",
                "properties": {
                    "From": {
                        "type": "string",
                        "description": "Kommt-Zeitpunkt"
                    },
                    "To": {
                        "type": "string",
                        "description": "Geht-Zeitpunkt"
                    },
                    "DayIntervalPlanSymbol": {
                        "type": "string",
                        "description": "Abwesenheit auf
Kommt/Geht-Paar (nur Planungscode mit Typ 'Abwesend')
"
                    },
                    "DayIntervalPlanSymbolObject": {
                        "$ref": "#/$defs
/planSymbolObjectType"
                    },
                    "DayIntervalInfoTextCategory": {
                        "type": "string",
                        "description": "Textkategorie auf
Kommt/Geht-Paar"
                    },
                    "DayIntervalInfoText": {

```


/Geht - Paar"

diesem Kommt/Geht-Paar zugewiesen ist"

auf Kommt/Geht-Paar",

von Aufgabe"

```

    "type": "string",
    "description": "Text auf Kommt
},
"DayIntervalShift": {
    "type": "string",
    "description": "Dienst, welcher
},
"DayIntervalShiftObject": {
    "$ref": "#/$defs/shiftObjectType"
},
"DayIntervalAssignedEvents": {
    "type": "array",
    "description": "Array von Aufgaben
    "items": {
        "type": "string",
        "description": "Bezeichnung
    }
}

```

Beispiele

Beispiel 1: gesamtes Personal

Ausgangssituation

Der Dienstplan A hat in diesem Beispiel 4 Mitarbeiter. Folgende Änderungen wurden beim Personal erfasst.

- Weitblick
 - 01.03.2023 Dienst: "8", Vormittags auswärts.
 - 01.03.2023 Tagedtext: Termin BitFactoy GmbH Supervisorenschulung 09:00 Uhr
- Rentier
 - 01.03.2023 Planungscode Schulung "SU"
- Imglück
 - 01.03.2023 Krankenstand "K"
 - 01.03.2023 Textkategorie: „MG“ Tagedtext: Mitarbeitergespräch
- Bauer
 - 01.03.2023 Dienst "8", Homeoffice "HO"

Dienstplan A

März 2023		Mi
Plan A		01
Ist		
Weitblick	Chef 	8 
Rentier	AT 	8
Imglück	AT 	K
Bauer	AT 	8

Aufruf der Funktion

Ein Alex-Server muss auf die unten angehängte Datenbank gestartet werden.

Danach können die Tagesinformationen des gesamten Personals vom 01.03.2023 abgerufen werden, wenn folgende URL in einem Webbrowser am selben PC eingegeben wird:

<http://localhost:4401/EmployeeDayIntervals?allEmployees=true&begin=01.03.2023&end=01.03.2023>

Ergebnis

JSON Datei

```
{
  "Employees": [
    {
      "EmployeeID": "1",
      "Surname": "Bauer",
      "Forename": "Berta",
      "Nickname": "",
      "EmployeeTitleBeforeName": "",
      "EmployeeTitleAfterName": "",
      "EmployeeJobGroup": "Arbeitstier",
      "Days": []
    },
    {
      "EmployeeID": "2",
      "Surname": "Auer",
      "Forename": "Anna",
      "Nickname": "",
      "EmployeeTitleBeforeName": "",
      "EmployeeTitleAfterName": "",
      "EmployeeJobGroup": "Arbeitstier",
      "Days": []
    },
    {
      "EmployeeID": "3",
      "Surname": "Eder",
      "Forename": "Ernst",
      "Nickname": "",
      "EmployeeTitleBeforeName": "",
      "EmployeeTitleAfterName": "",
      "EmployeeJobGroup": "Arbeitstier",
      "Days": []
    },
    {
      "EmployeeID": "4",
      "Surname": "Bauer",
      "Forename": "Birgit",
      "Nickname": "BBauer",
      "EmployeeTitleBeforeName": "Diplom-Ingenieurin",
      "EmployeeTitleAfterName": "BA",
      "Days": []
    }
  ]
}
```

```

"MasterAllocation": "Plan A",
"EmployeeJobGroup": "Arbeitstier",
"Days": [
  {
    "FromDate": "01.03.2023",
    "ForState": "Ist",
    "TextCategory": "",
    "InfoText": "",
    "DayPlanSymbol": "",
    "DayPlanSymbolObject": {},
    "DayShift1": "8",
    "DayShift2": "",
    "DayRequestKind": "H0",
    "DayEvents": [],
    "DayTotalMinutesComputed": 480.0,
    "DayPoolAllocations": [],
    "AssignedExtra": [],
    "DayIntervals": [
      {
        "From": "01.03.2023 8:00",
        "To": "01.03.2023 12:00",
        "DayIntervalPlanSymbol": "",
        "DayIntervalPlanSymbolObject": {},
        "DayIntervalInfoTextCategory": "",
        "DayIntervalInfoText": "",
        "DayIntervalShift": "8",
        "DayIntervalAssignedEvents": []
      },
      {
        "From": "01.03.2023 13:00",
        "To": "01.03.2023 17:00",
        "DayIntervalPlanSymbol": "",
        "DayIntervalPlanSymbolObject": {},
        "DayIntervalInfoTextCategory": "",
        "DayIntervalInfoText": "",
        "DayIntervalShift": "8",
        "DayIntervalAssignedEvents": []
      }
    ]
  }
],
{
  "EmployeeID": "5412",
  "Surname": "Imglück",
  "Forename": "Hans",
  "Nickname": "Lucky",
  "EmployeeTitleBeforeName": "",
  "EmployeeTitleAfterName": "",
  "MasterAllocation": "Plan A",
  "EmployeeJobGroup": "Arbeitstier",
  "Days": [
    {
      "FromDate": "01.03.2023",
      "ForState": "Ist",
      "TextCategory": "MG",
      "InfoText": "Mitarbeitergespräch",
      "DayPlanSymbol": "K",
      "DayPlanSymbolObject": {
        "PlanSymbolCode": "K",
        "PlanSymbolType": "Abwesend"
      },
      "DayShift1": "8",
      "DayShift2": "",
      "DayRequestKind": "",
      "DayEvents": [],
      "DayTotalMinutesComputed": 480.0,
      "DayPoolAllocations": [],
      "AssignedExtra": [],
      "DayIntervals": [
        {
          "From": "01.03.2023 8:00",
          "To": "01.03.2023 12:00",
          "DayIntervalPlanSymbol": "K",

```

```

        "DayIntervalPlanSymbolObject": {
            "PlanSymbolCode": "K",
            "PlanSymbolType": "Abwesend"
        },
        "DayIntervalInfoTextCategory": "",
        "DayIntervalInfoText": "",
        "DayIntervalShift": "",
        "DayIntervalAssignedEvents": []
    },
    {
        "From": "01.03.2023 13:00",
        "To": "01.03.2023 17:00",
        "DayIntervalPlanSymbol": "K",
        "DayIntervalPlanSymbolObject": {
            "PlanSymbolCode": "K",
            "PlanSymbolType": "Abwesend"
        },
        "DayIntervalInfoTextCategory": "",
        "DayIntervalInfoText": "",
        "DayIntervalShift": "",
        "DayIntervalAssignedEvents": []
    }
]
}
}
],
{
    "EmployeeID": "8812",
    "Surname": "Rentier",
    "Forename": "Rudolf",
    "Nickname": "Rudi",
    "EmployeeTitleBeforeName": "",
    "EmployeeTitleAfterName": "",
    "MasterAllocation": "Plan A",
    "EmployeeJobGroup": "Arbeitstier",
    "Days": [
        {
            "ForDate": "01.03.2023",
            "ForState": "Ist",
            "TextCategory": "",
            "InfoText": "",
            "DayPlanSymbol": "",
            "DayPlanSymbolObject": {},
            "DayShift1": "8",
            "DayShift2": "",
            "DayRequestKind": "SU",
            "DayEvents": [],
            "DayTotalMinutesComputed": 480.0,
            "DayPoolAllocations": [],
            "AssignedExtra": [],
            "DayIntervals": [
                {
                    "From": "01.03.2023 8:00",
                    "To": "01.03.2023 12:00",
                    "DayIntervalPlanSymbol": "",
                    "DayIntervalPlanSymbolObject": {},
                    "DayIntervalInfoTextCategory": "",
                    "DayIntervalInfoText": "",
                    "DayIntervalShift": "8",
                    "DayIntervalAssignedEvents": []
                },
                {
                    "From": "01.03.2023 13:00",
                    "To": "01.03.2023 17:00",
                    "DayIntervalPlanSymbol": "",
                    "DayIntervalPlanSymbolObject": {},
                    "DayIntervalInfoTextCategory": "",
                    "DayIntervalInfoText": "",
                    "DayIntervalShift": "8",
                    "DayIntervalAssignedEvents": []
                }
            ]
        }
    ]
}
]

```

```

    },
    {
      "EmployeeID": "4711",
      "Surname": "Weitblick",
      "Forename": "Susanne",
      "Nickname": "Susi",
      "EmployeeTitleBeforeName": "",
      "EmployeeTitleAfterName": "",
      "MasterAllocation": "Plan A",
      "EmployeeJobGroup": "Hauptling",
      "Days": [
        {
          "ForDate": "01.03.2023",
          "ForState": "Ist",
          "TextCategory": "",
          "InfoText": "Termin: BitFactoy GmbH Supervisorenschulung 09:00 Uhr",
          "DayPlanSymbol": "",
          "DayPlanSymbolObject": {},
          "DayShift1": "8",
          "DayShift2": "A",
          "DayRequestKind": "",
          "DayEvents": [],
          "DayTotalMinutesComputed": 480.0,
          "DayPoolAllocations": [],
          "AssignedExtra": [],
          "DayIntervals": [
            {
              "From": "01.03.2023 8:00",
              "To": "01.03.2023 12:00",
              "DayIntervalPlanSymbol": "",
              "DayIntervalPlanSymbolObject": {},
              "DayIntervalInfoTextCategory": "",
              "DayIntervalInfoText": "",
              "DayIntervalShift": "8",
              "DayIntervalAssignedEvents": []
            },
            {
              "From": "01.03.2023 13:00",
              "To": "01.03.2023 17:00",
              "DayIntervalPlanSymbol": "",
              "DayIntervalPlanSymbolObject": {},
              "DayIntervalInfoTextCategory": "",
              "DayIntervalInfoText": "",
              "DayIntervalShift": "8",
              "DayIntervalAssignedEvents": []
            }
          ]
        }
      ]
    }
  ]
}

```

Datenbank für Beispiel

[DokuDB Bsp 1](#)

Beispiel 2: Personal mit Personalnummer

Ausgangssituation

Der Dienstplan A hat in diesem Beispiel 4 Mitarbeiter. Folgende Personalnummer sind erfasst:

- Weitblick: 4711
- Rentier: 8812
- Imglück: 5412
- Bauer: 4

Mitarbeiter mit der Personalnummer

März 2023		Mi
Plan A		01
Ist		
Weitblick	4711 	
Rentier	8812 	
Imglück	5412 	
Bauer	4 	

Aufruf der Funktion

Ein Alex-Server muss auf die unten angehängte Datenbank gestartet werden.

Danach Tagesinformationen vom Personal Bauer mit der Personalnummer 4 vom 01.03.2023 abgerufen werden, wenn folgende URL in einem Webbrowser am selben PC eingegeben wird:

<http://localhost:4401/EmployeeDayIntervals?id=4&begin=01.03.2023&end=01.03.2023>

Ergebnis

JSON Datei

```
{
  "Employees": [
    {
      "EmployeeID": "4",
      "Surname": "Bauer",
      "Forename": "Birgit",
      "Nickname": "BBauer",
      "EmployeeTitleBeforeName": "Diplom-Ingenieurin",
      "EmployeeTitleAfterName": "BA",
      "MasterAllocation": "Plan A",
      "EmployeeJobGroup": "Arbeitstier",
      "Days": [
        {
          "ForDate": "01.03.2023",
          "ForState": "Ist",
          "TextCategory": "",
          "InfoText": "",
          "DayPlanSymbol": "",
          "DayPlanSymbolObject": {},
          "DayShift1": "8",
          "DayShift2": "",
          "DayRequestKind": "H0",
          "DayEvents": [],
          "DayTotalMinutesComputed": 480.0,
          "DayPoolAllocations": [],
          "AssignedExtra": [],
          "DayIntervals": [
            {
              "From": "01.03.2023 8:00",
              "To": "01.03.2023 12:00",
              "DayIntervalPlanSymbol": "",
              "DayIntervalPlanSymbolObject": {},
              "DayIntervalInfoTextCategory": "",
              "DayIntervalInfoText": "",
              "DayIntervalShift": "8",
              "DayIntervalAssignedEvents": []
            }
          ]
        }
      ]
    }
  ]
}
```

```
{
  "From": "01.03.2023 13:00",
  "To": "01.03.2023 17:00",
  "DayIntervalPlanSymbol": "",
  "DayIntervalPlanSymbolObject": {},
  "DayIntervalInfoTextCategory": "",
  "DayIntervalInfoText": "",
  "DayIntervalShift": "8",
  "DayIntervalAssignedEvents": []
}
```

Datenbank für Beispiel

[DokuDB Bsp 2](#)

Beispiel 3: Personal auf Plan

Ausgangssituation

Der Dienstplan A hat in diesem Beispiel 4 Mitarbeiter.

Dienstplan A Kurzzeichen: PA

- Weitblick
- Rentier
- Imglück
- Bauer

Mitarbeiter			
März 2023		Mi	
Plan A		01	
Ist			
Weitblick	4711 	8	A
Rentier	8812 	8	
Imglück	5412 	K	
Bauer	4 	8	

Aufruf der Funktion

Ein Alex-Server muss auf die unten angehängte Datenbank gestartet werden.

Danach kann man die Tagesinformationen für die Planungseinheit mit dem Kurzzeichen PA vom 01.03.2023 abgerufen werden, wenn folgende URL in einem Webbrowser am selben PC eingegeben wird:

<http://localhost:4401/EmployeeDayIntervals?planCode=PA&begin=01.03.2023&end=01.03.2023>

Ergebnis

JSON Datei

```

{
  "Employees": [
    {
      "EmployeeID": "4711",
      "Surname": "Weitblick",
      "Forename": "Susanne",
      "Nickname": "Susi",
      "EmployeeTitleBeforeName": "",
      "EmployeeTitleAfterName": "",
      "MasterAllocation": "Plan A",
      "EmployeeJobGroup": "Hauptling",
      "Days": [
        {
          "ForDate": "01.03.2023",
          "ForState": "Ist",
          "TextCategory": "",
          "InfoText": "Termin: BitFactoy GmbH Supervisorenschulung 09:00 Uhr",
          "DayPlanSymbol": "",
          "DayPlanSymbolObject": {},
          "DayShift1": "8",
          "DayShift2": "A",
          "DayRequestKind": "",
          "DayEvents": [],
          "DayTotalMinutesComputed": 480.0,
          "DayPoolAllocations": [],
          "AssignedExtra": [],
          "DayIntervals": [
            {
              "From": "01.03.2023 8:00",
              "To": "01.03.2023 12:00",
              "DayIntervalPlanSymbol": "",
              "DayIntervalPlanSymbolObject": {},
              "DayIntervalInfoTextCategory": "",
              "DayIntervalInfoText": "",
              "DayIntervalShift": "8",
              "DayIntervalAssignedEvents": []
            },
            {
              "From": "01.03.2023 13:00",
              "To": "01.03.2023 17:00",
              "DayIntervalPlanSymbol": "",
              "DayIntervalPlanSymbolObject": {},
              "DayIntervalInfoTextCategory": "",
              "DayIntervalInfoText": "",
              "DayIntervalShift": "8",
              "DayIntervalAssignedEvents": []
            }
          ]
        }
      ]
    },
    {
      "EmployeeID": "8812",
      "Surname": "Rentier",
      "Forename": "Rudolf",
      "Nickname": "Rudi",
      "EmployeeTitleBeforeName": "",
      "EmployeeTitleAfterName": "",
      "MasterAllocation": "Plan A",
      "EmployeeJobGroup": "Arbeitstier",
      "Days": [
        {
          "ForDate": "01.03.2023",
          "ForState": "Ist",
          "TextCategory": "",
          "InfoText": "",
          "DayPlanSymbol": "",
          "DayPlanSymbolObject": {},

```



```

        "DayShift1": "8",
        "DayShift2": "",
        "DayRequestKind": "SU",
        "DayEvents": [],
        "DayTotalMinutesComputed": 480.0,
        "DayPoolAllocations": [],
        "AssignedExtra": [],
        "DayIntervals": [
            {
                "From": "01.03.2023 8:00",
                "To": "01.03.2023 12:00",
                "DayIntervalPlanSymbol": "",
                "DayIntervalPlanSymbolObject": {},
                "DayIntervalInfoTextCategory": "",
                "DayIntervalInfoText": "",
                "DayIntervalShift": "8",
                "DayIntervalAssignedEvents": []
            },
            {
                "From": "01.03.2023 13:00",
                "To": "01.03.2023 17:00",
                "DayIntervalPlanSymbol": "",
                "DayIntervalPlanSymbolObject": {},
                "DayIntervalInfoTextCategory": "",
                "DayIntervalInfoText": "",
                "DayIntervalShift": "8",
                "DayIntervalAssignedEvents": []
            }
        ]
    },
    {
        "EmployeeID": "5412",
        "Surname": "Imglück",
        "Forename": "Hans",
        "Nickname": "Lucky",
        "EmployeeTitleBeforeName": "",
        "EmployeeTitleAfterName": "",
        "MasterAllocation": "Plan A",
        "EmployeeJobGroup": "Arbeitstier",
        "Days": [
            {
                "ForDate": "01.03.2023",
                "ForState": "Ist",
                "TextCategory": "MG",
                "InfoText": "Mitarbeitergespräch",
                "DayPlanSymbol": "K",
                "DayPlanSymbolObject": {
                    "PlanSymbolCode": "K",
                    "PlanSymbolType": "Abwesend"
                },
                "DayShift1": "8",
                "DayShift2": "",
                "DayRequestKind": "",
                "DayEvents": [],
                "DayTotalMinutesComputed": 480.0,
                "DayPoolAllocations": [],
                "AssignedExtra": [],
                "DayIntervals": [
                    {
                        "From": "01.03.2023 8:00",
                        "To": "01.03.2023 12:00",
                        "DayIntervalPlanSymbol": "K",
                        "DayIntervalPlanSymbolObject": {
                            "PlanSymbolCode": "K",
                            "PlanSymbolType": "Abwesend"
                        },
                        "DayIntervalInfoTextCategory": "",
                        "DayIntervalInfoText": "",
                        "DayIntervalShift": "",
                        "DayIntervalAssignedEvents": []
                    },
                    {

```

```

        "From": "01.03.2023 13:00",
        "To": "01.03.2023 17:00",
        "DayIntervalPlanSymbol": "K",
        "DayIntervalPlanSymbolObject": {
            "PlanSymbolCode": "K",
            "PlanSymbolType": "Abwesend"
        },
        "DayIntervalInfoTextCategory": "",
        "DayIntervalInfoText": "",
        "DayIntervalShift": "",
        "DayIntervalAssignedEvents": []
    }
}

}

{
    "EmployeeID": "4",
    "Surname": "Bauer",
    "Forename": "Birgit",
    "Nickname": "BBauer",
    "EmployeeTitleBeforeName": "Diplom-Ingenieurin",
    "EmployeeTitleAfterName": "BA",
    "MasterAllocation": "Plan A",
    "EmployeeJobGroup": "Arbeitstier",
    "Days": [
        {
            "ForDate": "01.03.2023",
            "ForState": "Ist",
            "TextCategory": "",
            "InfoText": "",
            "DayPlanSymbol": "",
            "DayPlanSymbolObject": {},
            "DayShift1": "8",
            "DayShift2": "",
            "DayRequestKind": "H0",
            "DayEvents": [],
            "DayTotalMinutesComputed": 480.0,
            "DayPoolAllocations": [],
            "AssignedExtra": [],
            "DayIntervals": [
                {
                    "From": "01.03.2023 8:00",
                    "To": "01.03.2023 12:00",
                    "DayIntervalPlanSymbol": "",
                    "DayIntervalPlanSymbolObject": {},
                    "DayIntervalInfoTextCategory": "",
                    "DayIntervalInfoText": "",
                    "DayIntervalShift": "8",
                    "DayIntervalAssignedEvents": []
                },
                {
                    "From": "01.03.2023 13:00",
                    "To": "01.03.2023 17:00",
                    "DayIntervalPlanSymbol": "",
                    "DayIntervalPlanSymbolObject": {},
                    "DayIntervalInfoTextCategory": "",
                    "DayIntervalInfoText": "",
                    "DayIntervalShift": "8",
                    "DayIntervalAssignedEvents": []
                }
            ]
        }
    ]
}

```

}

Datenbank für Beispiel

DokuDB Bsp 3

Beispiel 4: Beispiel Personal auf Plan und Personal mit Personalnummer

Ausgangssituation

In diesem Beispiel sind jeweils zwei Dienstpläne mit jeweils zwei Mitarbeiter erfasst:

Dienstplan A

Kurzzeichen: PA

- Weitblick: 4711
- Rentier: 8812

Dienstplan A	
März 2023	
Plan A	
Ist	
Weitblick	4711 !
Rentier	8812 !

Dienstplan C

Kurzzeichen: PC

- Imglück: 5412
- Bauer: 4

Dienstplan C	
März 2023	
Plan C	
Ist	
Bauer	4
Imglück	5412 !

Aufruf der Funktion

Ein Alex-Server muss auf die unten angehängte Datenbank gestartet werden.

Danach kann das Personal mit der Personalnummer 4 und das gesamte Personal von Dienstplan A vom 01.03.2023 abgerufen werden, wenn folgende URL in einem Webbrowser am selben PC eingegeben wird:

```
http://localhost:4401/EmployeeDayIntervals?begin=01.03.2023&end=01.03.2023&resourcesBson={"employeeIDs":["4"],"planCodes":["PA"]}
```

Ergebnis

JSON Datei

```

{
  "Employees": [
    {
      "EmployeeID": "4",
      "Surname": "Bauer",
      "Forename": "Birgit",
      "Nickname": "BBauer",
      "EmployeeTitleBeforeName": "Diplom-Ingenieurin",
      "EmployeeTitleAfterName": "BA",
      "MasterAllocation": "Plan C",
      "EmployeeJobGroup": "Arbeitstier",
      "Days": [
        {
          "ForDate": "01.03.2023",
          "ForState": "Ist",
          "TextCategory": "",
          "InfoText": "",
          "DayPlanSymbol": "",
          "DayPlanSymbolObject": {},
          "DayShift1": "8",
          "DayShift2": "",
          "DayRequestKind": "",
          "DayEvents": [],
          "DayTotalMinutesComputed": 480.0,
          "DayPoolAllocations": [],
          "AssignedExtra": [],
          "DayIntervals": [
            {
              "From": "01.03.2023 8:00",
              "To": "01.03.2023 12:00",
              "DayIntervalPlanSymbol": "",
              "DayIntervalPlanSymbolObject": {},
              "DayIntervalInfoTextCategory": "",
              "DayIntervalInfoText": "",
              "DayIntervalShift": "8",
              "DayIntervalAssignedEvents": []
            },
            {
              "From": "01.03.2023 13:00",
              "To": "01.03.2023 17:00",
              "DayIntervalPlanSymbol": "",
              "DayIntervalPlanSymbolObject": {},
              "DayIntervalInfoTextCategory": "",
              "DayIntervalInfoText": "",
              "DayIntervalShift": "8",
              "DayIntervalAssignedEvents": []
            }
          ]
        }
      ]
    },
    {
      "EmployeeID": "4711",
      "Surname": "Weitblick",
      "Forename": "Susanne",
      "Nickname": "Susi",
      "EmployeeTitleBeforeName": "",
      "EmployeeTitleAfterName": "",
      "MasterAllocation": "Plan A",
      "EmployeeJobGroup": "Hauptling",
      "Days": [
        {
          "ForDate": "01.03.2023",
          "ForState": "Ist",
          "TextCategory": "",
          "InfoText": "Termin: BitFactory GmbH Supervisorenschulung 09:00 Uhr",
          "DayPlanSymbol": "",
          "DayPlanSymbolObject": {},
          "DayShift1": "8",
          "DayShift2": "A",
          "DayRequestKind": "",
          "DayEvents": [],

```

```

        "DayTotalMinutesComputed": 480.0,
        "DayPoolAllocations": [],
        "AssignedExtra": [],
        "DayIntervals": [
            {
                "From": "01.03.2023 8:00",
                "To": "01.03.2023 12:00",
                "DayIntervalPlanSymbol": "",
                "DayIntervalPlanSymbolObject": {},
                "DayIntervalInfoTextCategory": "",
                "DayIntervalInfoText": "",
                "DayIntervalShift": "8",
                "DayIntervalAssignedEvents": []
            },
            {
                "From": "01.03.2023 13:00",
                "To": "01.03.2023 17:00",
                "DayIntervalPlanSymbol": "",
                "DayIntervalPlanSymbolObject": {},
                "DayIntervalInfoTextCategory": "",
                "DayIntervalInfoText": "",
                "DayIntervalShift": "8",
                "DayIntervalAssignedEvents": []
            }
        ]
    },
    {
        "EmployeeID": "8812",
        "Surname": "Rentier",
        "Forename": "Rudolf",
        "Nickname": "Rudi",
        "EmployeeTitleBeforeName": "",
        "EmployeeTitleAfterName": "",
        "MasterAllocation": "Plan A",
        "EmployeeJobGroup": "Arbeitstier",
        "Days": [
            {
                "ForDate": "01.03.2023",
                "ForState": "Ist",
                "TextCategory": "",
                "InfoText": "",
                "DayPlanSymbol": "",
                "DayPlanSymbolObject": {},
                "DayShift1": "8",
                "DayShift2": "",
                "DayRequestKind": "SU",
                "DayEvents": [],
                "DayTotalMinutesComputed": 480.0,
                "DayPoolAllocations": [],
                "AssignedExtra": [],
                "DayIntervals": [
                    {
                        "From": "01.03.2023 8:00",
                        "To": "01.03.2023 12:00",
                        "DayIntervalPlanSymbol": "",
                        "DayIntervalPlanSymbolObject": {},
                        "DayIntervalInfoTextCategory": "",
                        "DayIntervalInfoText": "",
                        "DayIntervalShift": "8",
                        "DayIntervalAssignedEvents": []
                    },
                    {
                        "From": "01.03.2023 13:00",
                        "To": "01.03.2023 17:00",
                        "DayIntervalPlanSymbol": "",
                        "DayIntervalPlanSymbolObject": {},
                        "DayIntervalInfoTextCategory": "",
                        "DayIntervalInfoText": "",
                        "DayIntervalShift": "8",
                        "DayIntervalAssignedEvents": []
                    }
                ]
            }
        ]
    }
]

```

```
}
  }
}
}
```

Datenbank für Beispiel

[DokuDB Bsp 4](#)

Beispiel 5: Alle Status

Ausgangssituation

Der Dienstplan A hat in diesem Beispiel 2 Mitarbeiter. Folgende Personalnummer sind erfasst:

- Weitblick
 - Personalnummer: 4711
- Rentier
 - Personalnummer: 8812

Dienstplan A	
März 2023	
Plan A	
Ist	
Weitblick	4711 !
Rentier	8812 !

Aufruf der Funktion

Ein Alex-Server muss auf die unten angehängte Datenbank gestartet werden.

Danach können die Tagesinformationen der Dienstplan-Status Planung und Ist der Personalnummer 4711 vom 01.03.2021 abgerufen werden, wenn folgende URL in einem Webbrowser am selben PC eingegeben wird:

```
http://localhost:4401/EmployeeDayIntervals?id=4711&begin=01.03.2023&end=01.03.2023
&allstates=true
```

Ergebnis

JSON Datei

```
{
  "Employees": [
    {
      "EmployeeID": "4",
      "Surname": "Bauer",
      "Forename": "Birgit",
      "Nickname": "BBauer",
      "EmployeeTitleBeforeName": "Diplom-Ingenieurin",
      "EmployeeTitleAfterName": "BA",
      "MasterAllocation": "Plan C",
      "EmployeeJobGroup": "Arbeitstier",
      "Days": [
        {
          "ForDate": "01.03.2023",
          "ForState": "Planung",

```

```

    "TextCategory": "",
    "InfoText": "",
    "DayPlanSymbol": "",
    "DayPlanSymbolObject": {},
    "DayShift1": "8",
    "DayShift2": "",
    "DayRequestKind": "",
    "DayEvents": [],
    "DayTotalMinutesComputed": 480.0,
    "DayPoolAllocations": [],
    "AssignedExtra": [],
    "DayIntervals": [
      {
        "From": "01.03.2023 8:00",
        "To": "01.03.2023 12:00",
        "DayIntervalPlanSymbol": "",
        "DayIntervalPlanSymbolObject": {},
        "DayIntervalInfoTextCategory": "",
        "DayIntervalInfoText": "",
        "DayIntervalShift": "8",
        "DayIntervalAssignedEvents": []
      },
      {
        "From": "01.03.2023 13:00",
        "To": "01.03.2023 17:00",
        "DayIntervalPlanSymbol": "",
        "DayIntervalPlanSymbolObject": {},
        "DayIntervalInfoTextCategory": "",
        "DayIntervalInfoText": "",
        "DayIntervalShift": "8",
        "DayIntervalAssignedEvents": []
      }
    ]
  },
  {
    "ForDate": "01.03.2023",
    "ForState": "Ist",
    "TextCategory": "",
    "InfoText": "",
    "DayPlanSymbol": "",
    "DayPlanSymbolObject": {},
    "DayShift1": "8",
    "DayShift2": "",
    "DayRequestKind": "",
    "DayEvents": [],
    "DayTotalMinutesComputed": 480.0,
    "DayPoolAllocations": [],
    "AssignedExtra": [],
    "DayIntervals": [
      {
        "From": "01.03.2023 8:00",
        "To": "01.03.2023 12:00",
        "DayIntervalPlanSymbol": "",
        "DayIntervalPlanSymbolObject": {},
        "DayIntervalInfoTextCategory": "",
        "DayIntervalInfoText": "",
        "DayIntervalShift": "8",
        "DayIntervalAssignedEvents": []
      },
      {
        "From": "01.03.2023 13:00",
        "To": "01.03.2023 17:00",
        "DayIntervalPlanSymbol": "",
        "DayIntervalPlanSymbolObject": {},
        "DayIntervalInfoTextCategory": "",
        "DayIntervalInfoText": "",
        "DayIntervalShift": "8",
        "DayIntervalAssignedEvents": []
      }
    ]
  }
]
}
]
}
]
}
]

```

}

Datenbank für Beispiel

DokuDB Bsp 5

Beispielanfragen

Tage von	URL
gesamtem Personal von 05.12.2017 bis einschließlich 07.12.2017	<code>/EmployeeDayIntervals?allEmployees=true&begin=05.12.2017&end=07.12.2017</code>
Personal mit Personalnummer 20 von 05.12.2017 bis einschließlich 07.12.2017 [1]	<code>/EmployeeDayIntervals?id=20&begin=05.12.2017&end=07.12.2017</code>
Personal auf Plan X von 01.12.2017 bis einschließlich 31.12.2017	<code>/EmployeeDayIntervals?planCode=X&begin=01.12.2017&end=31.12.2017</code>
Personal E1, E2, E3 und Personal auf Plänen P1, P2, P3 von 01.12.2017 bis einschließlich 31.12.2017	<code>/EmployeeDayIntervals?begin=01.12.2017&end=31.12.2017&resourcesBson={"employeeIDs":["E1","E2","E3"],"planCodes":["P1","P2","P3"]}</code>
Personal mit Personalnummer 20 von 05.12.2017 bis einschließlich 07.12.2017 , alle Status	<code>/EmployeeDayIntervals?id=20&begin=05.12.2017&end=07.12.2017&allstates=true</code>

Beispielausgabe

JSON-Datei einer Ausgabe

```
{
  "Employees": [
    {
      "EmployeeID": "20",
      "Surname": "Muster",
      "Forename": "Max",
      "EmployeeTitleBeforeName": "DI",
      "EmployeeTitleAfterName": "M.A",
      "MasterAllocation": "Plan S",
      "EmployeeJobGroup": "SW-Entwicklung",
      "Days": [
        {
          "ForDate": "05.12.2017",
          "ForState": "Ist",
          "TextCategory": "Korrektur am Tag",
          "InfoText": "Pause verlängert, Auto kaputt",
          "DayPlanSymbol": "",
          "DayPlanSymbolObject": {},
          "DayShift1": "8",
          "DayShift2": "",
          "DayRequestKind": "",
          "DayEvents": [],
          "DayTotalMinutesComputed": 480.0,
          "ForeignShift1": "Projektmanagment",
          "DayPoolAllocations": [],
          "AssignedExtra": [
            {
              "Key": "9010",
              "Label": "Diensttausch",
              "Value": 0.0
            }
          ],
          "DayIntervals": [
            {
```



```

        "From": "05.12.2017 8:00",
        "To": "05.12.2017 12:00",
        "DayIntervalPlanSymbol": "",
        "DayIntervalPlanSymbolObject": {},
        "DayIntervalInfoTextCategory": "",
        "DayIntervalInfoText": "",
        "DayIntervalShift": "8",
        "DayIntervalAssignedEvents": []
    },
    {
        "From": "05.12.2017 13:00",
        "To": "05.12.2017 17:00",
        "DayIntervalPlanSymbol": "",
        "DayIntervalPlanSymbolObject": {},
        "DayIntervalInfoTextCategory": "",
        "DayIntervalInfoText": "",
        "DayIntervalShift": "8",
        "DayIntervalAssignedEvents": []
    }
]
},
{
    "ForDate": "06.12.2017",
    "ForState": "Ist",
    "TextCategory": "",
    "InfoText": "",
    "DayPlanSymbol": "",
    "DayPlanSymbolObject": { "PlanSymbolCode": "S", "PlanSymbolCodeExt": "ext",
"PlanSymbolType": "Anwesend" },
    "DayShift1": "8",
    "DayShift2": "",
    "DayRequestKind": "",
    "DayEvents": [],
    "DayTotalMinutesComputed": 360.0,
    "DayPoolAllocations": [ "Plan A", "Plan B" ],
    "AssignedExtra": [],
    "DayIntervals": [
        {
            "From": "06.12.2017 8:00",
            "To": "06.12.2017 12:00",
            "DayIntervalPlanSymbol": "",
            "DayIntervalPlanSymbolObject": {},
            "DayIntervalInfoTextCategory": "",
            "DayIntervalInfoText": "",
            "DayIntervalShift": "8",
            "DayIntervalAssignedEvents": []
        },
        {
            "From": "06.12.2017 13:00",
            "To": "06.12.2017 15:00",
            "DayIntervalPlanSymbol": "",
            "DayIntervalPlanSymbolObject": {},
            "DayIntervalInfoTextCategory": "",
            "DayIntervalInfoText": "",
            "DayIntervalShift": "8",
            "DayIntervalAssignedEvents": []
        },
        {
            "From": "06.12.2017 15:00",
            "To": "06.12.2017 17:00",
            "DayIntervalPlanSymbol": "Z",
            "DayIntervalPlanSymbolObject": { "PlanSymbolCode": "Z", "PlanSymbolCodeExt":
"Zext", "PlanSymbolType": "Abwesend" },
            "DayIntervalInfoTextCategory": "",
            "DayIntervalInfoText": "",
            "DayIntervalShift": "8",
            "DayIntervalAssignedEvents": []
        }
    ]
},
{
    "ForDate": "07.12.2017",
    "ForState": "Ist",
    "TextCategory": "",

```

```

    "InfoText": "",
    "DayPlanSymbol": "",
    "DayPlanSymbolObject": {},
    "DayShift1": "VM",
    "DayShift2": "NM",
    "DayRequestKind": "",
    "DayEvents": [],
    "DayTotalMinutesComputed": 480.0,
    "DayPoolAllocations": [ "Plan B" ],
    "AssignedExtra": [],
    "DayIntervals": [
      {
        "From": "07.12.2017 8:00",
        "To": "07.12.2017 12:00",
        "DayIntervalPlanSymbol": "",
        "DayIntervalPlanSymbolObject": { "PlanSymbolCode": "B", "PlanSymbolCodeExt":
"Bext", "PlanSymbolType": "Anwesend" },
        "DayIntervalInfoTextCategory": "",
        "DayIntervalInfoText": "",
        "DayIntervalShift": "VM",
        "DayIntervalAssignedEvents": []
      },
      {
        "From": "07.12.2017 13:00",
        "To": "07.12.2017 17:00",
        "DayIntervalPlanSymbol": "",
        "DayIntervalPlanSymbolObject": {},
        "DayIntervalInfoTextCategory": "",
        "DayIntervalInfoText": "",
        "DayIntervalShift": "NM",
        "DayIntervalAssignedEvents": []
      }
    ]
  }
}

```

Fußnoten

1. [↑ #8081](#)

Wt-Webservice

Ein **Wt-Webservice** ist eine [Programmfunktion](#) zur Bereitstellung von Schnittstellen über das HTTP-Protokoll.

Voraussetzungen

Um die Programmfunktion nutzen zu können

- muss das [WebUI](#) konfiguriert sein.
- muss ein gültiger [WebUI-Port](#) konfiguriert sein.
- kann ein Zertifikat zur [Verschlüsselung](#) verwendet werden.
- kann ein [Sicherer Link-Aufruf](#) konfiguriert werden.

Das Webservice stellt dann Zugriffspfade an der konfigurierten [Webservice-Adresse](#) zur Verfügung.

Der Bediener des Webservice muss mit dem **HTTP-Protokoll** und seinen **Möglichkeiten zur Argumentübertragung** ^{[1][2]} vertraut sein.

Der Bediener muss das Format der Anfrage und Ausgabe kennen.

Funktionstest

[Live-Demo](#)

[Sourcecode](#)

Das Beispielprogramm **http-client-example** in HTML und JavaScript dient zur Veranschaulichung der Bedienung eines ALEX[®]-Webservice.

Mit diesem Beispielprogramm kann

- ein HTTP-GET Request an ein Webservice gesendet werden (Body-Eingabefeld bleibt leer)
- ein HTTP-POST Request an ein Webservice gesendet werden (Body-Eingabefeld ist gefüllt)
- ein API-Key gesendet werden

Im Response-Ausgabefeld wird das Ergebnis des Aufrufs angezeigt.

Mit diesem Beispielprogramm können alle ALEX[®]-Webservices bedient werden.

http-client-example

API-Key: 92834gZfr-942e-231x-f943-bb8c214779x2cb

URL: http://localhost:4401/Set

Body:

```
{
  "objectType": "Employee",
  "indexQuery": "EmployeeIDX",
  "importType": "EmployeeEmailAddress",
  "lines": [
    {
      "matchString": "4",
      "valueString": "birgit.bauer@bitfactory.at"
    }
  ]
}
```

Send

Response

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

Fußnoten

1. ↑ https://www.w3schools.com/tags/ref_httpmethods.asp
2. ↑ https://de.wikipedia.org/wiki/Hypertext_Transfer_Protocol#Argument%C3%BCbertragung