JSONP API - mobileRequest

Allgemeines

Die datenbanken24 JSONP Schnittstelle ermöglicht den automatisierten Zugriff auf Ihre datenbanken24.de Cloud-Application. Sie können über diese API:

- Suchabfragen gegen die Datenbank absetzen
- Dokumenten- / Datensatz- Informationen auslesen
- Konfigurationseinstellungen auslesen, z.B. Auswahllisten, Feldhilfetexte, etc.
- Dokumente / Datensätze kategorisieren
- auf eingebettete Dateien (z.B. PDF, Bilder, Grafiken) zugreifen

Sie können über die JSONP API keine Änderungen an der Datenbank vornehmen, insbesondere

- keine neuen Dokumente erstellen
- keine Dokumente bearbeiten oder löschen
- keine bestehende Dokumente bearbeiten bzw. ändern
- keine Konfigurationseinstellungen ändern

Die Schnittstelle ist dahingehend konzipiert, mit einer vom Nutzer selbst programmierten Benutzeroberfläche, z.B. für Smartphones, Android-Geräte, etc., Abfragen (Requests) gegen die Datenbank24 zu stellen und die im JSON-Format zurückgegebenen Ergebnisse wieder in einer selbst programmierten Benutzeroberfläche auszuwerten und darzustellen.
Die mobileRequest URL's haben also im Gegensatz zu anderen datenbanken24-URL's keinerlei Benutzeroberfläche oder UI-Funktionalität.

JSONP - ”JSON mit Padding”

JSON mit Padding ermöglicht die Übertragung von JSON-Daten über Domaingrenzen.

Üblicherweise erfolgen Ajax-Datenabfragen an Server über das XMLHttpRequest-Objekt eines Webbrowsers. Aufgrund der Same-Origin-Policy funktioniert das nicht, wenn die in einem Webbrowser angezeigte Webseite über dieses Objekt auf einen Server zuzugreifen versucht, der in einer anderen Domain als die angezeigte Webseite liegt.
Dieses Problem kann durch JSONP umgangen werden.

Die Anfrage / ”Request” erfolgt über einen GET oder POST (z.B. Ajax POST)
Die Antwort / ”Response” erfolgt im JSON Format.

API Version vom 02.02.2015, benötigt datenbanken24 Version ab 11.1502.02

Lizenz

Der Zugriff auf Ihre datenbanken24 Cloud-Application per SOAP Schnittstelle ist kostenfrei und in jeder Lizenzstufe möglich.

Zugriffssicherheit und Authentifizierung

Die JSONP API Requests laufen ausschließlich im öffentlichen Bereich
Ihrer Datenbank, also in der für Anonymous zugänglichen public.nsf.

Daher ist ein Login nicht erforderlich bzw. nicht nötig.

Das bedeutet, dass alle Daten, auf die Sie per JSONP mobileRequest zugreifen möchten, vorher in Ihrer Datenbank veröffentlicht werden müssen, also in den öffentlichen Bereich freigegeben werden müssen. Diese Schnittstelle läuft immer mit den Rechten eines "Lesers", schreibende und löschende Funktionen sind nicht möglich.

Datensatz = Dokument

datenbanken24.de ist eine nicht-relationale Dokumentendatenbank.

Ein datenbanken24 Dokument ist hierbei vergleichbar mit einem Datensatz / einem Record in einer relationalen Datenbank. Ein Dokument enthält bis zu 200 Felder und beschreibt z.B: ein Kundenprofil, einen Artikel, einen Bewerber, etc.

Ein datenbanken24 Formular entspricht grundsätzlich dem Aufbau einer Table.
Mit Formularen werden die nötigen Felder eines Dokuments individuell konfiguriert.
Für die Schnittstelle sind hierbei jedoch **nicht** Ihre individuell vergebenen Feldlabel relevant - sondern die internen Feldnummern der genutzten Felder.

Die Feldnummer eines Feldes erkennen Sie im Konfigurationsmenü "Formulargenerator" des entsprechenden Moduls. Hovern Sie mit der Maus über den Link "Feldeigenschaften".
Oder öffnen Sie den Feldeigenschaften-Dialog des betreffenden Feldes.
Dort steht im Reiter "Feldeigenschaften" z.B. der Hinweis:
Der interne Name dieses Feldes ist: F40.

Dieses Feld 40 können Sie in der Schnittstelle ansprechen über "F40"

Module umfassen alle Dokumente, die mit dem gleichem Formular erstellt wurden.
Sie sind vergleichbar mit Arbeitsmappen in Excel.
Die Zuordnung eines Dokuments zu einem Modul erfolgt über den Parameter ”Module=”, in der Form "1" bis "10"

Basis-URL der API

Eine Initialisierung der Schnittstelle ist nicht nötig. Die Basis-URL der API ist:

*.../**public.nsf/mobileRequest?openagent&callback=db24&...***

Die URL kann sowohl mit GET als auch mit POST angesprochen werden,
empfehlenswert und häufigster Anwendungsfall ist ein sogenannter Ajax POST
innerhalb Ihres Programmcodes.

Die Rückgabe (Response) ist immer ein JSON Objekt.

Alle Parameter sind nicht case-sensitiv.

&callback=

An jeden Request hängen Sie bitte den Parameter "**&callback=db24**" an, wobei der Wert des Parameters (z.B. "db24") frei wählbar ist. Er stellt den Namen der Callback-Funktion dar, um die domainübergreifende Funktionalität von JSONP zu ermöglichen.
Wenn Sie z.B. für die Requests jQuery verwenden, muß die Funktion nicht selbst programmiert werden, da unsere Response diese automatisch mitliefert.

Einfache &callback=? Auswertung mit jQuery

Bei jQuery Ajax Requests nutzn Sie einfach den Parameter: dataType: "jsonp".
Die Callback-Funktion wird hier immer als Fragenzeichen "?" übergeben

```
$.ajax({
        type: 'GET',
        url: url ...&callback=?,
        dataType: "jsonp",
});
```

Am Ende dieser Dokumentation finden Sie zwei ausführliche Beispiele für Abfragen mit der JSONP API für die Nutzung mit jQuery.

&Module=n
z.B.: &Module=1

Die Modulnummer bestimmt das Datenbank-Modul, in welchem nach Datensätzen gesucht wird ["1" … "10"]
Beim Weglassen dieses Parameters wird in Modul 1 gesucht.

Suchabfragen

Über den task=search Parameter können Suchabfragen auf die Datenbank abgesetzt werden

task=search

- Die Suche arbeitet nicht case-sensitive.
  Groß- und Kleinschreibung werden nicht beachtet.
- Hochkommas für die Suchparameter sind unschädlich, aber nicht nötig
- Wird in Auswahllisten gesucht, werden immer korrekte Treffer verlangt
- Wird in Textfeldern gesucht, wird der Suchwert auch als Teilstring gefunden
- alle Suchabfragen bzw. Suchparameter können miteinander kombiniert werden

Folgende Parameter können in diesem Kontext verwendet werden

Feldsuchparameter

&Fx=fieldvalue
z.B.: &F10=Berlin

Der Parameter übergibt den Wert oder die Werte für eine Feldsuche.
Finde Datensätze, die im Feld Nr. 10 den Wert ”Berlin” enthalten

&maxresults=

Der Parameter „maxResults“ bestimmt die Anzahl der Suchergebnisse,
die maximal vom Request zurück geliefert werden sollen.
Wenn Sie diesen Parameter weglassen greift die Default-Einstellung von maximal 250 Treffern

Default = maximal 250

&maxresults=0
gibt alle Suchergebnisse zurück, die gefunden wurden - ohne Einschränkung.

”UND” - Verknüpfung bei der Feldsuche

&Fn=fieldvalue&Fm=fieldvalue wird als ”UND” - Verknüpfung ausgewertet

&F10=Berlin&F11=Restaurant

Feldsuchparameter für verschiedene Felder
werden immer als ”AND” bzw. ”UND” Verknüpfung ausgewertet.

Das Beispiel sucht demnach nach Datensätzen,
in denen im Feld 10 der Wert ”Berlin” enthalten ist
UND gleichzeitig im Feld 11 der Wert ”Restaurant” enthalten ist.

”ODER” - Verknüpfungen bei der Feldsuche

Mehrere mögliche Feldwerte (”ODER”) können übergeben werden, indem man den gleichen Feldsuchparameter mehrfach anhängt, analog dem Post eines Checkbox-Feldes in HTML.

So kann z.B: ein Checkbox-Feld aus Ihrer eigenen Benutzeroberfläche über ein „Serialize“ direkt an die Datenbank gesendet werden.

&Fn=fieldvalue1&Fn=fieldvalue2&Fn=fieldvalue3
wird als ”ODER” - Verknüpfung ausgewertet

&F10=Berlin&F11=Restaurant&F11=Kneipe&F11=Kiosk

Das Beispiel sucht also nach Datensätzen,
in denen im Feld 10 der Wert ”Berlin” enthalten ist
UND gleichzeitig im Feld 11 der Wert ”Restaurant” **oder** ”Kneipe” **oder** ”Kiosk” enthalten ist.

Optionaler Parameter: &FxLogicOperator=

Möchten Sie hingegen auch innerhalb eines Feldes mehrere Werte
durch eine „UND“-Verknüpfung auswerten lassen,
dh. es müssen alle gesendeten Parameter im gesuchten Datensatz in dem betreffenden Feld gleichzeitig enthalten sein, dann können Sie die Default-“ODER“-Suche innerhalb eines Feldes in eine „UND“-Suche umschalten:

z.B. die Einsatzgebiete eines Models „Beauty“, Fashion“ und „Bademode“
müssen alle drei im Feld 20 enthalten sein:

&F20=Beauty&F20=Fashion&F20=Bademode&F20LogicOperator=and

Das System findet in diesem Beispiel nur diejenigen Datensätze (Models),
in denen alle diese drei Bereiche enthalten (angeklickt, ausgewählt worden) sind.

”größer <> kleiner” - Suche

Eine Suche in der Form ”F2>100&F2<500” ist NICHT MÖGLICH.

Stattdessen teilt die JSONP API die numerische Suche bzw. die numerischen Felder in ein

_FROM
und ein
_TO
auf.

&F2_From=100

sucht alle Datensätze, in denen der numerische Wert des Feldes 2 größergleich 100 ist.

&F2_To=500

sucht alle Datensätze, in denen der numerische Wert des Feldes 2 kleinergleich 500 ist.

&F2_From=100**&F2_To=**500

sucht alle Datensätze, in denen der numerische Wert des Feldes 2 zwischen 100 und 500 liegt.

Hinweise für die numerische Suche:

Jedes Feld, gleich welchen Typs steht im internen datenbanken24-System sowohl als Textfeld als auch als numerisches Feld zur Verfügung. Daher gibt die numerische Suche auf ein Textfeld prinzipiell keinen Fehler zurück und auf jedes numerische Feld kann auch per Textsuche gesucht werden.

Der numerische Wert von nicht-konvertierbaren Textfeldern, wie z.B. „rot“, ist 0.

Die Suche z.B. nach einer Postleitzahl kann somit sowohl numerisch als auch textlich erfolgen. Ist z.B in einem Feld 30 eine PLZ abgespeichert, völlig egal mit welchen Feldtyp das Feld 30 konfiguriert ist, so kann über:

&F30_From=38100&F30_To=38500 numerisch über dieses Feld gesucht werden

und mit

&F30=381* gleichermaßen textlich über dieses Feld gesucht werden.

Datum-Suche

Die Datum-Suche kann einzeln abgesetzt werden
oder mit den Feldsuchparametern als ”UND” Verknüpfung kombiniert werden.

Jedes Feld, gleich welchen Typs steht im internen datenbanken24-System sowohl als Textfeld als auch als numerisches Feld zur Verfügung. Daher gibt die numerische Suche auf ein Textfeld prinzipiell keinen Fehler zurück und auf jedes numerisches Feld kann auch per Textsuche gesucht werden.

Hierbei werden speziell Felder vom Typ „Datum“ intern neben der Speicherung als Text ebenfalls auf eine einfache numerische Form gespeichert,
als Zahl in einer YEAR MONTH DAY Umrechnung

Das Datum „28.02.2015“ wird auch gespeichert als 20150228 bzw. 20.150.228
Das Datum „10.03.2015“ wird auch gespeichert als 20150310 bzw. 20.150.310

Eine Suche nach Datensätzen, die zwischen zwei Datumswerten liegen,
ist somit möglich über:
&F40_From=20150228&F40_To=20150310

Eine Suche mit diesen Parametern findet z.B. alle Events, die einem bestimmten Zeitraum stattfinden. Eine Kombination mit anderen Feldwerten oder einer Umkreissuche ist möglich.

Die Volltextsuche

Die Volltextsuche sucht nicht in bestimmten Feldern nach bestimmten Werten
sondern sucht in allen Feldern eines Datensatzes nach irgendeinem Vorkommen dieser Suchbegriffe

Die Volltextsuche kann einzeln abgesetzt werden
oder zusammen mit den Feldsuchparametern als ”UND” Verknüpfung kombiniert werden.

&FTKeywords=phrase1+phrase2+...

&FTKeywords=Pizza

findet alle Datensätze, welche das Wort ”Pizza” **in irgendeinem Feld** enthalten

&FTLogicOperator=and

and: Mehrere Suchphrasen werden mit einer „UND“-Verknüpfung ausgewertet
or: Mehrere Suchphrasen werden mit einer „ODER“-Verknüpfung ausgewertet
x: Mehrere Suchphrasen müssen als EXAKT DIESE Wortgruppe gesucht

&F10=Berlin&F11=Restaurant&F11=Kneipe&F11=Kiosk&FTKeywords=pizza+pasta
&FTLogicOperator=and

sucht nach allen Datensätzen, in denen im Feld 10 der Wert ”Berlin” enthalten ist
UND gleichzeitig im Feld 11 der Wert ”Restaurant” **oder** ”Kneipe” **oder** ”Kiosk” enthalten ist
UND gleichzeitig in irgendeinem Feld im gesamten Datensatz das Wort ”pizza” vorkommt.

&FTLogicOperator:
and: „pizza“ und „pasta“ müssen in irgendwelchen Feldern stehen
or: „pizza“ oder „pasta“ müssen in irgendwelchen Feldern stehen
x: „pizza hawai“ muss in irgendeinem Feld stehen

Score

Die Response Property „Score“ bildet eine Rangliste ab, wie häufig der gesuchte Begriff oder die gesuchten Begriffe in diesem Datensatz vorkommen. Je höher der Score desto häufiger wurden die Suchbegriffe in diesem Datensatz gefunden.

Volltext-Suche in eingebetteten Dateien

Sofern Dateien in den Datensätzen angehängt sind (hochgeladen wurden),
die textlich auswertbar sind., z.B.
PDFs, Word-, Excel- oder OpenOffice-Dokumente, Powerpoint-Präsentationen, etc.
können diese Dateien in die Volltextsuche einbezogen werden.

Das datenbanken24-System kann also während der Volltextsuche auf Datensätze,
gleichzeitig die in den Datensätzen hochgeladenen Dateien
nach Suchwörtern oder Stichwörtern durchsuchen.
Diese Option kann durch unsere Hotline freigeschaltet werden.

Umkreissuche mit geo-codierten Adressen

Eine Umkreissuche kann durchgeführt werden,

- wenn als Suchparameter ein Startpunkt mit einer geocodierbaren Adresse
- oder ein Startpunkt definiert über Geo-Koordinaten
- und ein Umkreis-Radius

mitgegeben wird - und die zu durchsuchenden Datensätze bzw.

- **Adressen in der Datenbank bereits geocodiert sind.**

Die Umkreissuche kann einzeln abgesetzt werden
oder mit anderen Feldsuchparametern kombiniert werden.

&F_Radius_S=50000

Radius im Metern.

&F_GM_S=Lange+Str+38100+Braunschweig+Germany
[Adresse als geocodierbarer Text / String]

oder

&F_GM_S=@52.2630024,10.521431399999983
[Adresse als LatLng Geo-Koordinaten]

oder die Kombination aus beiden

&F_GM_S=Lange+Straße+61+38100+Braunschweig+Deutschland+
/@52.26689,10.51837999999998

Beispiel:
&F_Radius_S=10000&F_GM_S=Berlin+Fernsehturm&F11=Restaurant&F11=Kneipe&F11=Kiosk
&FTKeywords=pizza

sucht nach allen Datensätzen, in denen
im Feld 11 der Wert ”Restaurant” **oder** ”Kneipe” **oder** ”Kiosk” enthalten ist
UND gleichzeitig in irgendeinem Feld im gesamten Datensatz das Wort ”pizza” vorkommt
UND die Adresse im Umkreis von 10 km vom Fernsehturm Berlin liegt.

Distance
Im JSON-Response finden Sie für jeden gefundenen Datensatz in der Property ”Distance”
seine Entfernung im km zum Startpunkt.
Somit können die Ergebnisse - über eigene Routinen - nach Entfernung - sortiert werden.

Response-Properties eines Datensatz-Objektes

Die Response einer Suchabfrage ist ein JSON Array, in welchem jedes Element / Objekt einen gefundenen Datensatz widerspiegelt.

UNID:
bezeichnet die Universal-ID des Datensatzes

ID:
bezeichnet die Dokumenten-ID des Datensatzes innerhalb der Datenbank

Fx:
enthalten die Feldwerte aller konfigurierten Felder F1 bis F200 dieses Datensatzes. Es werden nur Konfigurierte, also genutzte Felder ausgegeben, in der Reihenfolge Ihrer Platzierung im Formular

CDate:
”Creation date” - ist das Erstelldatum dieses Datensatzes bzw. Dokumentes

LCMDate:
”Last customer modification date” - ist das Datum, an welchem dieser Datensatz das letzte mal geändert wurde

Score:
keine Bedeutung im mobileRequest

HasCommunity:
wenn ”1” dann befindet sich das Dokument in einem aktiven Workflow

HasAttachment:
wenn ”1” dann beinhaltet der Datensatz eingebettete Dateianhänge

RA:
keine Bedeutung für mobileRequest

Attachments:
alle eingebetteten Dateianhänge in diesem Dokument bzw. Datensatz, getrennt durch ein Pipe-Zeichen ”|”

Distance:
Die Entfernung zum Startpunkt bei einer Umkreissuche

Geo_LatLng:
Die Geo-Koordinaten bei einem geocodierten Datensatz

PathDia:
Der URL-Pfad zu dem Thumbnail in diesem Datensatz

CustViewField1:, CustViewField2:
Vorbereitete Felder für Customizing bzw. individuelle Kundenanpassungen

Beispiel-Response einer Suchabfrage

db24(
[
{"UNID" : "19D87BED420BB760C1257DDD003A0680",
"ID" : "121018-31003-AD-927917928",
"F52" : "Jenny",
"F33" : "Meyer",
"F1" : "06.06.1969",
"F42" : "04.10.1969",
"F13" : "Portrait,Fashion,Beauty,Bademode,Dessous,Teilakt”,
"F14" : "International",
"F16" : "Fashionweek 2011",
"F44" : "Ringstraße 10",
"F45" : "38518",
"F46" : "Gifhorn",
"F47" : "Deutschland",
"CDate" : "2012.10.18.08.36.42",
"LCMDate" : "2015.01.18.01.41.42",
"Score" : "0",
"HasCommunity" : "0",
"HasAttachment" : "1",
"RA" : "0",
"Attachments" : "Fotolia_16439573_XS.jpg|Sedcard.pdf|autothumb.jpg",
"Distance" : "",
"Geo_LatLng" : "52.4709091,10.5392504",
"PathDia" : "(IDs)/19D87BED420BB760C1257DDD003A0680/$FILE/autothumb.jpg",
"CustViewField1" : "",
"CustViewField2" : ""
},
…
...

Das Objekt ”ServiceDoc”

Das letzte Objekt einer search Response ist immer das sogenannte ”ServiceDoc”.
Es stellt zusammenfassende Informationen zu dieser Response zur Verfügung.
Eine Suchabfrage, die 10 Datensätze findet und diese als Ergebnis zurückliefert,
hat in ihrem JSON-Response Array 10 Datensatz-Objekte und als 11. Objekt das Service-Doc.

UNID:
Konstante: ”ServiceDoc"

FieldList:
gibt die Feldnummern in der Reihenfolge zurück, wie Sie im Formulargenerator konfiguriert wurden, getrennt durch Semikolon

Fx_Label:
Das konfigurierte Label des Feldes,
um es in eigenen Ansichten oder Darstellungsformen kennzeichnen zu können.

Task:
Der Task, der diesem Response zugrunde lag.

maxResults:
Der dem zugrundeliegenden Request mitgegebene Parameter ”&maxresults=”.
Wurde kein Parameter mitgegeben, dann ist maxResults per Default 250.

Results:
Anzahl der als Suchergebnisse in dieser Response zurückgegebenen Datensätze.
Results ist abhängig vom Parameter maxresults.

allResults:
Anzahl der Suchergebnisse für diese Suche.
Results ist unabhängig vom Parameter maxresults.

Für eine Suchanfrage werden z.B. 30 Datensätze in Ihrer Datenbank gefunden:
In diesem Fall liefern Results und allResults beide den Wert 30 zurück.

Für eine Suchanfrage werden z.B. 4.000 Datensätze in Ihrer Datenbank gefunden.
Es wurde kein maxResults im Request mitgegeben:
In diesem Fall liefert Results den Wert 250
und allResults den Wert 4.000 zurück.

Für eine Suchanfrage werden z.B. 4.000 Datensätze in Ihrer Datenbank gefunden.
Es wurde ein maxResults=0 als Parameter im Request mitgegeben:
In diesem Fall liefern Results und allResults beide den Wert 4.000 zurück.

Error:
Beinhaltet den Fehlermeldungs-Text,
falls es bei der Ausführung der Suche zu einem internen Fehler kommt.

isRadialSearch:
wenn ”1” dann hat das System diese Suche als Umkreissuche erkannt

Konfigurationsdaten von Feldern auslesen

/mobileRequest?openagent&task=configuration&Module=n

Um in Ihrer Benutzeroberfläche, z.B in einer Suchabfrage analoge Auswahllisten anzubieten, wie diese auch in der internen Application-Oberfläche abgebildet sind, benötigen Sie die konfigurierte Auswahlliste eines Auswahlfeldes aus der Konfiguration Ihrer Datenbank.

task=configuration

&Module=n
z.B.: &Module=1
Die Modulnummer bestimmt das Datenbank-Modul, in welchem nach Datensätzen gesucht wird ["1" … "10"] Beim Weglassen dieses Parameters wird in Modul 1 gesucht.

Response:

[Array of objects] Gibt die Felder als Objekte mit deren Feldeigenschaften zurück, in der Reihenfolge, wie Sie im Formulargenerator konfiguriert wurden im Formular erscheinen.

FieldNo:
[Text] Gibt die interne Feldnummer des Feldes zurück

FieldType:
[Text] Gibt den Feldtyp des Feldes zurück, z.B. "TEXT", "NUMERIC", "KW", "COMPUTED", etc.

FieldLabel:
[Text] Gibt den Feldlabel des Feldes zurück.

FieldHelp:
[Text] Gibt die konfigurierte Feldhilfe (Hilfetext) z.B: für Tooltips, Eingabehilfen, etc. zurück.

FieldKW:
[Array] Gibt die konfigurierte Keywordliste zurück, wenn es sich bei dem Feld um eine Auswahlliste handelt. Die Einzelwerte der Auswahlliste werden als Array zurückgegeben. Der Aufbau eines Array-Elements sind durch Pipe-Zeichen getrennte Option|Value Pärchen.

FieldUnit:
[Text] Gibt die konfigurierte Maßeinheit des Feldes zurück .

FieldDefault:
[Text] Gibt den konfigurierten Vorgabewert des Feldes zurück.

Beispiel einer Response für das Konfigurationsdaten-Objekt

...public.nsf/mobileRequest?openagent&task=configuration&Module=1&callback=db24


```
db24(
[
{
"FieldNo" : "52",
"FieldType" : "TEXT",
"FieldLabel" : "Model-ID",
"FieldHelp" : "Model-ID<br>Zum bestimmen des Models.",
"FieldAlias" : "0",
"FieldKW" : [""],
"FieldUnit" : "",
"FieldDefault" : "UNIQUE"
},
{
"FieldNo" : "4",
"FieldType" : "NUMERIC",
"FieldLabel" : "Gewicht",
"FieldHelp" : "",
"FieldAlias" : "0",
"FieldKW" : [""],
"FieldUnit" : "kg",
"FieldDefault" : ""
},
{
"FieldNo" : "5",
"FieldType" : "KW",
"FieldLabel" : "Konfektionsgröße",
"FieldHelp" : "",
"FieldAlias" : "0",
"FieldKW" : ["30 (EU)|30","32 (EU)|32","34 (EU)|34","36 (EU)|36","38 (EU)|38”],
"FieldUnit" : "",
"FieldDefault" : ""
}
]
);
```

Kategorisierung von Datensätzen

&task=category

Die Kategorisierung von Datensätzen ist eine Abfrage über alle Datensätze eines Moduls einer Datenbank, welche Feldwerte in einem bestimmten Feld vorkommen - und wie oft.

Um eine Abfrage nach Städten durchzuführen, können Sie in Ihrer eigenen Abfragemaske entweder die Auswahlliste des Feldes aus dem Konfigurationsobjekt nutzen.
Hier bekommen Sie alle Auswahlmöglichkeiten zurück, die für dieses Feld konfiguriert wurden, gleichgültig, ob diese Auswahl bereits in einem Dokument genutzt (ausgewählt) wurde, oder nicht, z.B.: die Städte
”Berlin”, ”Hamburg”, ”München”, ”Braunschweig”, ”Düsseldorf”, ”Köln”

Möchten Sie in Ihrer Abfragemaske jedoch nur die Auswahlmöglichkeiten (z.B. Städte) zur Auswahl anbieten, zu denen auch wirklich bereits Dokumente / Datensätze existieren, dann können Sie diesen Task vor Ihre Auswahl vorschalten.

&cat=Fn

&task=category&FN=F1&cat=F10

Sie erhalten eine Response, welche Werte bereits im Feld 10 über alle Datensätze vorhanden sind, d.h. jede Einzel-Suche, die über Ihre Oberfläche abgesendet wird, findet auch ein Ergebnis.

z.B.: die Städte ”Berlin”, ”Hamburg”, ”München”

Response:

Zurückgegeben wird ein JSON Array aus Objekten mit folgenden Properties:

UNID: der Feldwert, der in vorhanden Datensätzen gefunden wurde
Count: in wievielen Datensätzen ist der Wert gefunden worden

Beispiel einer category-Response

```
db24(
[
{"UNID" : "Hamburg",
"Count" : "4"
},
{"UNID" : "Berlin",
"Count" : "3"
},
{"UNID" : "München",
"Count" : "7"
}
]
);
```

Abfrage von Einzel-Dokumenten bzw. Einzel-Datensätzen

&task=document

Über den Task „document" können Sie einzelne Dokumente / Datensätze auslesen.
Der Parameter hierbei ist die ID des jeweiligen Dokuments.

&id=Doc-ID

&task=document&id=121018-31003-AD-927917928

Der Aufbau des Response Datensatz-Objektes und des zugehörigen ServiceDoc-Objektes entspricht analog dem des Datensatz-Objektes und ServiceDoc-Objektes einer Suchabfrage.

Erweiterungen für dieses Objekt sind geplant.

Beispiel für ein Ajax GET Request mit jQuery

var url = "…/***public.nsf/mobileRequest?openagent&callback=?***"

url = url + "&Module=1&task=search&FTKeywords=Berlin"

```
$.ajax({
        type: 'GET',
        url: url,
        dataType: "jsonp",
        Error : function(errObject, status, errorThrown) {
                        alert(errObject.responseText);
                        alert(status);
                        alert(errorThrown);
        },
        success: function(json, textStatus) {
                        alert("Success with GET request");
                        var searchResult = json.slice(0, -1);
                        var ServiceDoc = json.slice(-1);
                        alert( "Results: " + ServiceDoc[0].Results);
        }
});
```

Beispiel für ein Ajax POST Request mit jQuery

var url = "…/***public.nsf/mobileRequest?openagent&callback=?***"

var data = "&Module=1&task=search&FTKeywords=Berlin"

```
$.ajax({
        type: 'POST',
        url: url,
        data: data,
        dataType: "jsonp",
        Error : function(errObject, status, errorThrown) {
                        alert(errObject.responseText);
                        alert(status);
                        alert(errorThrown);
        },
        success: function(json, textStatus) {
                        alert("Success with POST request");
                        var searchResult = json.slice(0, -1);
                        var ServiceDoc = json.slice(-1);
                        alert( "Results: " + ServiceDoc[0].Results);
        }
});
```