

Inhalt

1	API-Übersicht	2
1.1.	Motivation	2
1.2.	Überblick	2
1.3.	Templating	2
2	Versionen	3
2.1.	Version 1.1, (April 2018) Umkreissuche für Veranstaltungen 	3
2.2.	Version 1.2, (Nov. 2018) Optimierungen und JSON-Schnittstelle	3
2.2.1.	Getrennte Felder für Veranstalterdaten (fehlende Werte unterdrückbar)	3
2.2.2.	Kategorie-ID zur Veranstaltung	3
2.2.3.	Feld „Leitung“ (neu)	3
2.2.4.	Datumsformat	3
2.2.5.	GetOrtAll() mit optionalem Parameter für Region	3
2.2.6.	JSON-Ausgabe im Flex-ABO möglich (Erprobung)	3
2.2.7.	Sicherheitsverbesserungen, Fehlermeldungen	3
2.3.	Klassifikationen	4
2.4.	Veranstaltungsdaten	4
3	HTML-Beispiel 	7
4	JSON-Beispiel 	11

1 API-Übersicht

1.1. Motivation

Eine Anmeldung als Abonnent (Flexibel), [hier Abo anlegen](#), ermöglicht die gezielte, maschinelle Abfrage von Daten aus dem aktuellen Terminbestand des WebKalenders. Diese Daten können (z.B. als spezielle Terminübersichten) dann Eingang in die Web-Angebote einzelner Gemeinden oder Interessengruppen finden.

Das Abonnement (Flexibel) erlaubt es gegenüber dem Abonnement (Standard), das Datenformat und das Layout sowie die Abfrageparameter und die Datenaufbereitung mit Hilfe des ZOPE-Editors zur Bearbeitung von DTML/HTML-Templates frei zu konfigurieren. Damit lassen sich bei Bedarf auch interaktive oder mehrseitige Web Auftritte erstellen.

Um dieses Angebot effizient nutzen zu können, sollte man über eine gewisse Praxis im Umgang mit HTML und ZOPE bzw. von vergleichbaren Template-Engines verfügen.

1.2. Überblick

Die Abfrage von Daten mit Hilfe des Abo-Typs „Flexibel“ unterscheidet zwischen steuernden Klassifikations- und den eigentlichen Veranstaltungsdaten. Im Bereich Klassifikation stehen diese Methoden:

- `getKategorieAll()`
- `getRegionAll()`,
- `getOrtAll()`, `getOrtAll([...])`

zur Verfügung. Sie liefern jeweils alle relevanten Attribute, die für die Entitäten „Kategorie“, „Region“ und „Ort“ im Veranstaltungskalender vorgehalten werden. Die eigentlichen Veranstaltungsdaten können mit den Methoden:

- `select_tmpl(...)` *(nicht mehr empfohlen)*
- `select_tmpl2(...)` *(empfohlen)*
- `select_tmpl_json(...)` *(ab V1.2 in Erprobung)*

abgefragt werden. Diese Methoden nehmen eine Reihe von Parametern, u. A. der Klassifikation, entgegen, mit der Veranstaltungen z.B. nur über einen bestimmten Zeitraum, in einer oder mehreren Regionen ausgewählt werden können.

1.3. Templating

Innerhalb der ZOPE-Umgebung, hier unter Ihrem „./Abonnent/[login-name]/manage“-Ordner, werden die o.g. Methoden zum Auslesen typischerweise in eine `<dtml-in>`-Schleife eingebettet und dann über die Template-Variablen `<dtml-var ...>` ausgeliefert:

```
<dtml-in "getKategorieAll()" size="5">
  <tr>
    <td><dtml-var kat_id null="&nbsp;"></td>
    ...
  </tr>
</dtml-in>
```

Damit lassen sich HTML- aber auch XML-Dokumente erzeugen, ein vollständiges Beispiel folgt unten.

Generell ist es empfehlenswert, das Zielencoding des zu erzeugenden Dokuments als erste Zeile des Templates wie folgt festzulegen, z.B.:

```
<dtml-call "RESPONSE.setHeader('Content-Type','text/html; charset=UTF-8')">
```

2 Versionen

2.1. Version 1.1, (April 2018) Umkreissuche für Veranstaltungen

In Version 1.1 wurde die API um eine Umkreissuche für Veranstaltungsorte erweitert. `select_tmp1 (...)` und `select_tmp2 (...)` nehmen die neuen Parameter

`p_lat` („Latitude“ für die geogr. Breite),
`p_lon` (für die geogr. Länge), und
`p_rds` (für den Radius des Umkreises) entgegen.

Das Ergebnis steht in `ort_distance` (Abstand [in km]) zur Verfügung.

Des Weiteren wurde die Anzahl der Informationen für Veranstaltungsorte in `select_tmp1 (...)` und `select_tmp2 (...)` sowie `getOrtAll ()` in

<code>ort_xcode,</code>	(PLZ)
<code>ort_xtown,</code>	(Ort/Stadt)
<code>ort_xtown2,</code>	(Ortsteil, falls vorhanden)
<code>ort_xstreet,</code>	(Straße, Hausnummer)
<code>ort_latitude,</code>	(geographische (nördliche) Breite) und
<code>ort_longitude</code>	(geographische (östliche) Länge) separiert.

2.2. Version 1.2, (Nov. 2018) Optimierungen und JSON-Schnittstelle

2.2.1. Getrennte Felder für Veranstalterdaten (fehlende Werte unterdrückbar)

Ab Version 1.2 sind die Informationen für Veranstaltungsanbieter für `select_tmp1 (...)` sowie `select_tmp2 (...)` in

<code>tmn_xsite_mgr,</code>	(Leitung)	
<code>tmn_xsite_name,</code>	(Name)	
<code>tmn_xsite_phone,</code>	(Telefon)	
<code>tmn_xsite_fax,</code>	(Fax)	und
<code>tmn_xsite_email</code>	(E-Mail) separiert.	

2.2.2. Kategorie-ID zur Veranstaltung

Dergleichen werden in `tmn_kat_ids` für jeden Termin jetzt auch die IDs der zugeordneten Kategorien geliefert.

2.2.3. Feld „Leitung“ (neu)

`tmn_xsite_mgr` (Leitung), siehe Kapitel 2.2.1

2.2.4. Datumsformat

Schließlich werden „von“- und gegebenenfalls „bis“-Zeitpunkt von Terminen mit `tmn_von_datum_iso` und `tmn_bis_datum_iso` nun auch im ISO8601-Format zur Verfügung gestellt.

2.2.5. GetOrtAll() mit optionalem Parameter für Region

Die Methode `getOrtAll ()` akzeptiert jetzt eine Liste von Region-IDs, wie bekannt aus der Verfahrensweise in `select_tmp2 (...)`. Aus Gründen der Kompatibilität kann dieser Parameter aber auch nach wie vor weggelassen werden.

2.2.6. JSON-Ausgabe im Flex-ABO möglich (Erprobung)

Die neue Methode `select_tmp1_json (...)` liefert die Ergebnisse in JSON-kompatibler Formatierung aus.

2.2.7. Sicherheitsverbesserungen, Fehlermeldungen

Die Fehlerausgabe wurde u.a. vereinheitlicht, Beispiel:

Es ist ein Fehler aufgetreten

Bitte wenden Sie sich mit Nennung der letzten URL, Datum / Uhrzeit sowie dem folgenden Text an cn@evlks.de.

[Hier steht die jeweilige (englische) Fehlermeldung.]

2.3. Klassifikationen

- `getKategorieAll()`

Ausgabe, Lieferung als Rowset

Attribut	Bemerkung	Version
kat_id	Datenbank-ID der Kategorie	1.0
kat_name	Name	1.0
kat_bemerkung	Bemerkung	1.0

- `getRegionAll()`

Ausgabe, Lieferung als Rowset

Attribut	Bemerkung	Version
reg_id	Datenbank-ID der Region	1.0
reg_name	Name	1.0
reg_bemerkung	Bemerkung	1.0

- `getOrtAll()`, `getOrtAll([region_id])`

Eingabe, alle Parameter AND-verknüpft

Attribut	Bemerkung	Version
p_reg_id	optional, Region-ID, Array, Wertebereich aus <code>getRegionAll()</code> , Leersuche mit <code>p_reg_id=[]</code> oder (nach wie vor) Weglassen des Parameters	1.2

Ausgabe, Lieferung als Rowset

Attribut	Bemerkung	Version
ort_id	Datenbank-ID des Ortes	1.0
ort_name	Name	1.0
ort_bemerkung	Bemerkung	1.0
ort_xcode	Postleitzahl	1.1
ort_xtown	Stadt/Gemeinde	1.1
ort_xtown2	Ortsteil	1.1
ort_xstreet	Straße Nummer	1.1
ort_latitude	geogr. Breite	1.1 
ort_longitude	geogr. Länge	1.1 

Beispiel-Treffermengen für Klassifikationen von Regionen und Orten vom 08.01.2019 siehe:

[https://webkalender.evks.de/Abonent/ABO_203/Doku/getRegionAll\(\)](https://webkalender.evks.de/Abonent/ABO_203/Doku/getRegionAll())

2.4. Veranstaltungsdaten

Als erster Parameter einer Abfrage an den Web Kalender dient ein Zeiger auf das Abonnement, der mit Hilfe der Methode `getAbo()` ermittelt werden kann.

Zur Einschränkung einer Abfrage dienen jeweils

- Kategorien, Regionen, Orte, Termine,
- ein Zeitfenster,
- Freitext sowie
- Breite, Länge und Umkreis. 

Kategorien, Regionen, Orte und Termine werden als ID-Liste angegeben, z.B. [1, 2, 3]. Keine Angabe benötigt ebenso ein Objekt des Typs Liste, hier [].

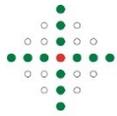
Der Rahmen des gewünschten Zeitfensters (nur **von**, nur **bis** oder beides kombiniert **von-bis** sind möglich) erwartet einen String des Formats Jahr-Monat-Tag, z.B. '2004-10-30'. Keine Auswahl ist dabei ein leerer String, hier ''.

Wenn die Freitextsuche nicht verwendet werden soll ist für `p_text` ebenfalls ein leerer String zu verwenden.

Für die Umkreissuche werden für Breite und Länge jeweils Dezimalzahlen und für den Umkreis eine Ganzzahl erwartet, die anderen Parameter sind damit nicht relevant. 

Eine Übergabe von jeweils 0 führt **keine** Umkreissuche aus, **nur damit** sind die anderen Parameter (wie bisher) relevant.

(siehe Folgeseite)



Veranstaltungskalender - Datenbank-API

API für Abo-Typ „Flexibel“

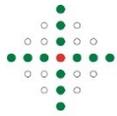
- `select_tmpl(...)` *(nicht mehr empfohlen)*

Eingabe, alle Parameter AND-verknüpft

Attribut	Bemerkung	Version
<code>p_abo=getAbo()</code>	Abo-Zeiger, nicht anfassen	1.0
<code>p_reg_id</code>	Region-ID, Array, Wertebereich aus <code>getRegionAll()</code> , Leersuche mit <code>p_reg_id=[]</code>	1.0
<code>p_kat_id</code>	Kategorie-ID, Array, Wertebereich aus <code>getKategorieAll()</code> , Leersuche mit <code>p_kat_id=[]</code>	1.0
<code>p_ort_id</code>	Ort-ID, Array, Wertebereich aus <code>getOrtAll()</code> , Leersuche mit <code>p_ort_id=[]</code>	1.0
<code>p_tm_n_id</code>	Termin-ID, Array, Leersuche mit <code>p_tm_n_id=[]</code>	1.0
<code>p_dat_von</code>	Datum, String, ISO, Leersuche mit <code>p_dat_von=''</code> , Default von <code>_.DateTime().strftime('%Y-%m-%d')</code> wegen der erwarteten großen Treffermenge sehr empfehlenswert	1.0
<code>p_dat_bis</code>	Datum, String, ISO, Leersuche mit <code>p_dat_bis=''</code>	1.0
<code>p_ftxt</code>	Freitext, String, Substring-Suche, Leersuche mit <code>p_ftxt=''</code>	1.0
<code>p_lat</code>	geogr. Breite, Dezimalzahl, Leersuche mit <code>p_lat=0</code>	1.1
<code>p_lon</code>	geogr. Länge, Dezimalzahl, Leersuche mit <code>p_lon=0</code>	1.1
<code>p_rds</code>	Umkreis [km], Ganzzahl, Leersuche mit <code>p_rds=0</code>	1.1

Ausgabe, Lieferung als Rowset

Attribut	Bemerkung	Version
<code>tmn_id</code>	Datenbank-Id	1.0
<code>ort_name</code>	wie <code>getOrtAll()</code>	1.0
<code>ort_xcode</code>	wie <code>getOrtAll()</code>	1.1
<code>ort_xtown</code>	wie <code>getOrtAll()</code>	1.1
<code>ort_xtown2</code>	wie <code>getOrtAll()</code>	1.1
<code>ort_xstreet</code>	wie <code>getOrtAll()</code>	1.1
<code>ort_latitude</code>	wie <code>getOrtAll()</code>	1.1
<code>ort_longitude</code>	wie <code>getOrtAll()</code>	1.1
<code>ort_distance</code>	Umkreis [km]	1.1
<code>ort_bemerkung</code>	wie <code>getOrtAll()</code>	1.0
<code>tmn_von datum fmt</code>	Datum/Zeit, ISO	1.0
<code>tmn_bis datum fmt</code>	Datum/Zeit, ISO	1.0
<code>tmn_titel</code>	Titel	1.0
<code>tmn_titel_2</code>	Untertitel	1.0
<code>tmn_beschreibung</code>	Bemerkung	1.0
<code>tmn_kosten</code>	Kosten	1.0
<code>tmn_va_daten</code>	Leitung: <...> <i>(neues Feld, manuelle Bearbeitung nötig)</i> Name: <...> Telefon: <...> Telefax: <...> E-Mail: <...> <i>(jede Zeile nur, wenn entsprechende Daten vorhanden sind)</i>	1.2
<code>tmn_xsite_mngr</code>	Veranstaltung: Leitung <i>(neues Feld, manuelle Bearbeitung nötig)</i>	1.2
<code>tmn_xsite_name</code>	Veranstaltung: Name	1.2
<code>tmn_xsite_phone</code>	Veranstaltung: Telefon	1.2
<code>tmn_xsite_fax</code>	Veranstaltung: Telefax	1.2
<code>tmn_xsite_email</code>	Veranstaltung: E-Mail	1.2
<code>tmn_link</code>	Kontakt, URL	1.0
<code>tmn_kat_ids</code>	Kategorien (IDs)	1.2



Veranstungskalender - Datenbank-API

API für Abo-Typ „Flexibel“

- `select_tmpl2(...)` **(empfohlen)**

Eingabe, alle Parameter AND-verknüpft

Attribut	Bemerkung	Version
<code>p_abo=getAbo()</code>	Abo-Zeiger, nicht anfassen	1.0
<code>p_reg_id</code>	Region-ID, Array, Wertebereich aus <code>getRegionAll()</code> , Leersuche mit <code>p_reg_id=[]</code>	1.0
<code>p_kat_id</code>	Kategorie-ID, Array, Wertebereich aus <code>getKategorieAll()</code> , Leersuche mit <code>p_kat_id=[]</code>	1.0
<code>p_ort_id</code>	Ort-ID, Array, Wertebereich aus <code>getOrtAll()</code> , Leersuche mit <code>p_ort_id=[]</code>	1.0
<code>p_tm_n_id</code>	Termin-ID, Array, Leersuche mit <code>p_tm_n_id=[]</code>	1.0
<code>p_dat_von</code>	Datum, String, ISO, Leersuche mit <code>p_dat_von=''</code> , Default von <code>_.DateTime().strftime('%Y-%m-%d')</code> wegen der erwarteten großen Treffermenge sehr empfehlenswert	1.0
<code>p_dat_bis</code>	Datum, String, ISO, Leersuche mit <code>p_dat_bis=''</code>	1.0
<code>p_ftxt</code>	Freitext, String, Substring-Suche, Leersuche mit <code>p_ftxt=''</code>	1.0
<code>p_lat</code>	geogr. Breite, Dezimalzahl, Leersuche mit <code>p_lat=0</code>	1.1
<code>p_lon</code>	geogr. Länge, Dezimalzahl, Leersuche mit <code>p_lon=0</code>	1.1
<code>p_rds</code>	Umkreis [km], Ganzzahl, Leersuche mit <code>p_rds=0</code>	1.1
<code>p_ein_tag</code>	nur eintägig, Bool, String, mit <code>p_ein_tag='True'</code> Leersuche (alle Termine) mit <code>p_ein_tag='False'</code>	1.0

Ausgabe, Lieferung als Rowset

Attribut	Bemerkung	Version
<code>tmn_id</code>	Datenbank-ID	1.0
<code>ort_name</code>	wie <code>getOrtAll()</code>	1.0
<code>ort_xcode</code>	wie <code>getOrtAll()</code>	1.1
<code>ort_xtown</code>	wie <code>getOrtAll()</code>	1.1
<code>ort_xtown2</code>	wie <code>getOrtAll()</code>	1.1
<code>ort_xstreet</code>	wie <code>getOrtAll()</code>	1.1
<code>ort_latitude</code>	wie <code>getOrtAll()</code>	1.1
<code>ort_longitude</code>	wie <code>getOrtAll()</code>	1.1
<code>ort_distance</code>	Entfernung [km], aber nur, wenn über Umkreissuche abgefragt	1.1
<code>ort_bemerkung</code>	wie <code>getOrtAll()</code>	1.0
<code>tmn_von_datum_fmt</code>	Datum/Zeit, ISO	1.0
<code>tmn_bis_datum_fmt</code>	Datum/Zeit, ISO	1.0
<code>tmn_titel</code>	Titel	1.0
<code>tmn_titel_2</code>	Untertitel	1.0
<code>tmn_beschreibung</code>	Bemerkung	1.0
<code>tmn_kosten</code>	Kosten	1.0
<code>tmn_va_datn</code>	Leitung: <...> (neues Feld, manuelle Bearbeitung nötig) Name: <...> Telefon: <...> Telefax: <...> E-Mail: <...> (jede Zeile nur, wenn entsprechende Daten vorhanden sind) nur empfohlen, wenn genau obiger HTML-Code verwendet werden soll, sonst:	1.2
<code>tmn_xsite_mngr</code>	Veranstaltung: Leitung (neues Feld, manuelle Bearbeitung nötig)	1.2
<code>tmn_xsite_name</code>	Veranstaltung: Name	1.2
<code>tmn_xsite_phone</code>	Veranstaltung: Telefon-Nr.	1.2
<code>tmn_xsite_fax</code>	Veranstaltung: Telefax-Nr.	1.2
<code>tmn_xsite_email</code>	Veranstaltung: E-Mail	1.2
<code>tmn_link</code>	Kontakt, URL	1.0
<code>tmn_kat_ids</code>	Kategorien (IDs)	1.2

3 HTML-Beispiel

Nachfolgend ein Code-Beispiel für einen Großteil der oben getroffenen Aussagen für die XML/HTML-Ausgabe. Mit Hilfe des o.g. Templating wird eine elementare HTML-Seite aufgebaut. Als Treiber dient das Rowset, das `select_tmpl/select_tmpl2` entsprechend der Parametrisierung zurückgibt. Außerdem wird auch gezeigt, wie die Termine abgerufen werden können. Im praktischen Einsatz werden die hier nur beispielhaft fest eingesetzten Parameter meist über Formularfelder von der Benutzereingabe abhängig gestaltet.

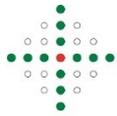
```
<dtml-call "RESPONSE.setHeader('Content-Type', 'text/html; charset=UTF-8')">
<!DOCTYPE html>
<html lang="de">
<head>
  <meta charset="utf-8"/>
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>API V1.2.0 (grob)</title>
</style>

* {
  font-family: sans-serif;
  font-size: 10px;
}
table, th, td {
  border: 1px solid grey;
  border-collapse: collapse;
}

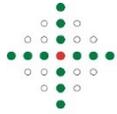
h1 {
  font-size: 16px;
}
th, h2, h3, li, a {
  font-size: 12px;
}

tbody tr:nth-child(even) {
  background-color: #f0f0f0;
  color: #000;
}

</style>
</head>
<body><dtml-comment>(Diese Kommentare unterdrücken jeweils nur den Zeilenwechsel im
ausgelieferten HTML-Quellcode!)
</dtml-comment><dtml-call "REQUEST.set('pq_reg_id', [])"> <dtml-comment>
</dtml-comment><dtml-call "REQUEST.set('pq_kat_id', [])"> <dtml-comment>
</dtml-comment><dtml-call "REQUEST.set('pq_ort_id', [])"> <dtml-comment>
</dtml-comment><dtml-call "REQUEST.set('pq_tmn_id', [])"> <dtml-comment>
</dtml-comment><dtml-call "REQUEST.set('pq_dat_von', _.DateTime().strftime('%Y-%m-%d'))">
<dtml-comment>
</dtml-comment><dtml-call "REQUEST.set('pq_dat_bis', '')"> <dtml-comment>
</dtml-comment><dtml-call "REQUEST.set('pq_ftxt', '')"> <dtml-comment>
</dtml-comment><dtml-call "REQUEST.set('pq_lat', 51.165047)"> <dtml-comment>
</dtml-comment><dtml-call "REQUEST.set('pq_lon', 13.469377)"> <dtml-comment>
</dtml-comment><dtml-call "REQUEST.set('pq_rds', 15)">
<h1>API Dokumentation (einfach, Stand November 2018)</h1>
<h2>neu in Version 1.1:</h2>
<ul><li>Geo-Koordinaten</li>
<li>Umkreis-Suche</li>
</ul>
<h2>neu in Version 1.2:</h2>
<ul><li>Veranstalterdaten: neues Feld 'Leitung' (Die Inhalte sind erst noch redaktionell
zu pflegen.)</li>
<li>Einzelfelder-Rückgabe (fehlende Werte unterdrückbar)</li>
<li>Kategorie-IDs zur Veranstaltung</li>
<li>GetOrtAll([...]) mit optionalem Parameter für Region</li>
</ul>
<h2>Tipps und Links:</h2>
<ul><li>Bitte beachten Sie auch das alternative Beispiel mit <a href="/API_V1.2">farbig
markierten Änderungen</a></li>
<li>Bitte beachten Sie auch das alternative Beispiel mit <a href="/index_json">JSON-
Ausgabe</a></li>
<li>Die veraltete Schnittstelle: select_tmpl(...) sollte zukünftig durch 'select_tmpl2(...)'
ersetzt werden.</li>
<li>Bitte wenden Sie sich an uns, wenn Sie zusätzliche Anforderungen oder Anregungen zur
Verbesserung haben.</li>
</ul>
```



```
<h1>select_tmpl2() und select_tmpl1()</h1>
<table>
  <tr><th colspan="12">select_tmpl2() Aktuelle Termine (Auszug!)</th></tr>
  <tr>
    <th>Termin (ID)</th>
    <th>Ort (PLZ, Ort | Ortsteil <br>Lat/<br>Long<br>Entfernung)</th>
    <th>Ort (Bemerkung)</th>
    <th>Datum von</th>
    <th>Datum bis</th>
    <th>Titel</th>
    <th>Untertitel</th>
    <th>Beschreibung</th>
    <th>Kosten</th>
    <th>Veranstaltungsdaten<br />Das Feld 'Leitung' muss erst noch redaktionell befüllt
werden!</th>
    <th>WWW-Link</th>
    <th>Kat-IDs</th>
  </tr>
<dtml-in "select_tmpl2(p_abo=getAbo(),p_reg_id=pq_reg_id, p_kat_id=pq_kat_id,
p_ort_id=pq_ort_id, p_tm_n_id=pq_tm_n_id, p_dat_von=pq_dat_von, p_dat_bis=pq_dat_bis,
p_ftxt=pq_ftxt, p_ein_tag='True', p_lat=pq_lat, p_lon=pq_lon, p_rds=pq_rds)" size="5">
  <tr>
    <td><dtml-var tm_n_id></td>
    <td><dtml-var ort_name><br /><br />
<dtml-if ort_xtown2><dtml-var ort_xtown2><br /></dtml-if>
<dtml-var ort_xcode>&nbsp;<dtml-var ort_xtown><br />
<dtml-var ort_xstreet><br /><br />
<dtml-var ort_latitude>&nbsp;<br />
<dtml-var ort_longitude>
<dtml-if "ort_distance is not None"><br /><dtml-var ort_distance> km</dtml-if></td>
    <td><dtml-var ort_bemerkung newline_to_br></td>
    <td><dtml-var tm_n_von datum_fmt></td>
    <td><dtml-var tm_n_bis datum_fmt></td>
    <td><dtml-var tm_n_titel newline_to_br></td>
    <td><dtml-var tm_n_titel_2 newline_to_br></td>
    <td><dtml-var tm_n_beschreibung newline_to_br></td>
    <td><dtml-var tm_n_kosten></td>
    <td>
<!-- obsolete, zusammengefasste Variante pre 1.2 (hier nicht dargestellt) -->
<!-- aktuell bevorzugte Variante -->
    <table>
<dtml-if "tm_n_xsite_mgr is not None"> <tr><td><dtml-var
tm_n_xsite_mgr></td></tr></dtml-if>
<dtml-if "tm_n_xsite_name is not None"> <tr><td><dtml-var
tm_n_xsite_name></td></tr></dtml-if>
<dtml-if "tm_n_xsite_phone is not None"> <tr><td><dtml-var
tm_n_xsite_phone></td></tr></dtml-if>
<dtml-if "tm_n_xsite_fax is not None"> <tr><td><dtml-var
tm_n_xsite_fax></td></tr></dtml-if>
<dtml-if "tm_n_xsite_email is not None"> <tr><td><dtml-var
tm_n_xsite_email></td></tr></dtml-if>
    </table>
    </td>
    <td><dtml-var tm_n_link></td>
    <td><dtml-var tm_n_kat_ids></td>
  </tr>
<dtml-comment>Die folgende Zeile wird verwendet, wenn keine Treffermenge vorhanden
ist.</dtml-comment>
<dtml-else><tr><th colspan="12">Aktuell sind KEINE Termine zur Position/Umkreis
gepflegt</th></tr>
</dtml-in>
```

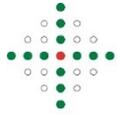


Veranstaltungskalender - Datenbank-API

API für Abo-Typ „Flexibel“

```
<tr><td colspan="12">select_tmpl() Bitte verwenden Sie stattdessen ab sofort  
möglichst obiges 'select_tmpl2()' mit zusätzlichem Parameter!</td></tr>  
<dtml-in "select_tmpl(p_abo=getAbo(), p_reg_id=pq_reg_id, p_kat_id=pq_kat_id,  
p_ort_id=pq_ort_id, p_tm_n_id=pq_tm_n_id, p_dat_von=pq_dat_von,  
p_dat_bis=pq_dat_bis,p_ftxt=pq_ftxt, p_lat=pq_lat, p_lon=pq_lon, p_rds=pq_rds)"  
size="10">  
  <tr>  
    <td><dtml-var tmn_id></td>  
    <td><dtml-var ort_name><br /><br />  
      <dtml-var ort_xcode>&nbsp;<dtml-var ort_xtown>  
<dtml-if ort_xtown2><br />OT&nbsp;<dtml-var ort_xtown2></dtml-if><br />  
<dtml-var ort_xstreet><br /><br />  
<dtml-var ort_latitude>&nbsp;</br><dtml-var ort_longitude>  
<dtml-if "ort_distance is not None"><br /><dtml-var ort_distance> km</dtml-if></td>  
    <td><dtml-var ort_bemerkung newline_to_br></td>  
    <td><dtml-var tmn_von_datum_fmt></td>  
    <td><dtml-var tmn_bis_datum_fmt></td>  
    <td><dtml-var tmn_titel newline_to_br></td>  
    <td><dtml-var tmn_titel_2 newline_to_br></td>  
    <td><dtml-var tmn_beschreibung newline_to_br></td>  
    <td><dtml-var tmn_kosten></td>  
    <td>  
  <!-- obsolete, zusammengefasste Variante pre 1.2 (aber verwendbar) -->  
<dtml-var tmn va daten newline to br>  
<!-- aktuell bevorzugte Variante -->  
    <table>  
      <tr style="border: 1px solid grey;"><td><dtml-var tmn_xsite_image></td></tr>  
      <tr style="border: 1px solid grey;"><td><dtml-var tmn_xsite_name></td></tr>  
      <tr style="border: 1px solid grey;"><td><dtml-var tmn_xsite_phone></td></tr>  
      <tr style="border: 1px solid grey;"><td><dtml-var tmn_xsite fax></td></tr>  
      <tr style="border: 1px solid grey;"><td><dtml-var tmn_xsite_email></td></tr>  
    </table>  
  </td>  
    <td><dtml-var tmn link></td>  
    <td><dtml-var tmn_kat_ids></td>  
  </tr>  
</dtml-in>  
</table>
```

```
<h1>getKategorieAll()</h1>  
<table>  
  <tr><th colspan="3">Kategorien</th></tr>  
  <tr><th>ID</th><th>Name</th><th>Bemerkung</th></tr>  
<dtml-in "getKategorieAll()" size="5">  
  <tr>  
    <td><dtml-var kat_id></td>  
    <td><dtml-var kat_name></td>  
    <td><dtml-var kat_bemerkung></td>  
  </tr>  
</dtml-in>  
</table>
```



```
<h1>getRegionAll()</h1>
<h2>getOrtAll([...]), hier je Region aufgerufen</h2>
<dtml-in "getRegionAll()" size="25">
  <table>
    <tr><th>ID<br /><dtml-var reg_id</th><th>Name<br /><dtml-var reg_name</th><th
  colspan="7">Bemerkung<br /><dtml-var reg_bemerkung</th></tr>
    <tr>
      <td>Orts-<br />ID</td>
      <td>Name</td>
      <td>Bemerkung</td>
      <td>PLZ</td>
      <td>Ort</td>
      <td>Ortsteil</td>
      <td>Straße</td>
      <td>Latitude</td>
      <td>Longitude</td>
    </tr>
  <dtml-in "getOrtAll(p_reg_id=reg_id)">
    <tr>
      <td><dtml-var ort_id</td>
      <td><dtml-var ort_name</td>
      <td><dtml-var ort_bemerkung</td>
      <td><dtml-var ort_xcode</td>
      <td><dtml-var ort_xtown</td>
      <td><dtml-var ort_xtown2</td>
      <td><dtml-var ort_xstreet</td>
      <td><dtml-var ort_latitude</td>
      <td><dtml-var ort_longitude</td>
    </tr>
  <dtml-else><tr><th colspan="9">Keine Orte für diese Region gepflegt.</th></tr>
</dtml-in>
</table>
<br />
</dtml-in>
```

```
<dtml-call "REQUEST.set('pq_reg_id', [])">
<h1>getOrtAll()</h1>
<table>
  <tr><th colspan="9">Orte <dtml-var pq_reg_id> (hier begrenzt auf 50)</th></tr>

  <tr>
    <th>Orts-<br />ID</th>
    <th>Name</th>
    <th>Bemerkung</th>
    <th>PLZ</th>
    <th>Ort</th>
    <th>Ortsteil</th>
    <th>Straße</th>
    <th>Latitude</th>
    <th>Longitude</th>
  </tr>
  <dtml-in "getOrtAll(pq_reg_id)" size="50">
    <tr>
      <td><dtml-var ort_id</td>
      <td><dtml-var ort_name</td>
      <td><dtml-var ort_bemerkung</td>
      <td><dtml-var ort_xcode</td>
      <td><dtml-var ort_xtown</td>
      <td><dtml-var ort_xtown2</td>
      <td><dtml-var ort_xstreet</td>
      <td><dtml-var ort_latitude</td>
      <td><dtml-var ort_longitude</td>
    </tr>
  </dtml-in>
</table>
<br />
Code-Stand 17.12.2018
</body>
</html>
```

Beispiel-Treffermenge vom 08.01.2019 siehe https://webkalender.evks.de/Abonnent/ABO_203/Doku/API

4 JSON-Beispiel

Die JSON-Ausgabe kann verwendet werden, um nachgelagerte CMS, Webseiten, Webdienste, ... mit Daten zu versorgen.

Der folgende Quellcode ist ein Beispiel für die JSON-Ausgabe für mehrere Termine (in Erprobung)

```
<dtml-call "RESPONSE.setHeader('Access-Control-Allow-Origin','*')">
<dtml-call "RESPONSE.setHeader('Content-Type','application/json; charset=UTF-8')">
<dtml-call "REQUEST.set('pq_reg_id', [])">
<dtml-call "REQUEST.set('pq_kat_id', [])">
<dtml-call "REQUEST.set('pq_ort_id', [])">
<dtml-call "REQUEST.set('pq_tm_n_id', [])">
<dtml-call "REQUEST.set('pq_dat_von', _.DateTime().strftime('%Y-%m-%d'))">
<dtml-call "REQUEST.set('pq_dat_bis', _.DateTime().strftime('%Y-%m-%d'))">
<dtml-call "REQUEST.set('pq_ftxt', '')">
<dtml-call "REQUEST.set('pq_lat', 51.165047)">
<dtml-call "REQUEST.set('pq_lon', 13.469377)">
<dtml-call "REQUEST.set('pq_rds', 5)">
<dtml-var expr="select_tmpl_json(p_abo=getAbo(),
p_reg_id=pq_reg_id,
p_kat_id=pq_kat_id,
p_ort_id=pq_ort_id,
p_tm_n_id=pq_tm_n_id,
p_dat_von=pq_dat_von,
p_dat_bis=pq_dat_bis,
p_ftxt=pq_ftxt,
p_lat=pq_lat,
p_lon=pq_lon,
p_rds=pq_rds)" >
```

Je nach Browser-/einstellung werden diese zum ‚Download‘ oder zum ‚Öffnen‘ im Browser angeboten.

HinweisG:

Der Parameter „p_dat_bis“ wird (aktuell/vorerst) nur bei Terminen ohne Ende-Datum und ohne Ende-Uhrzeit berücksichtigt!

Mehrtägige Termine oder eintägige Termine mit ausgefülltem Ende-Datum/-Uhrzeit werden (aktuell/vorerst) auch nach dem „p_dat_bis“ Datum angezeigt.

Die Abfrage „pq_dat_von“ erfolgt taggenau, ohne Berücksichtigung der Uhrzeit.

Beispiel-Treffermenge vom 08.01.2019 siehe https://webkalender.evks.de/Abonnent/ABO_203/Doku/index_json