

Lexparency Bot-API

Die Bot-API erlaubt eine einfache und maschinenlesbare Abfrage von kürzlich aktualisierten Inhalten. Die API ist ReST-basiert und besteht aus zwei Teilen: die „Recents“- und die Dokument- Abfrage.

Die Recents-Abfrage

Hierbei handelt es sich um ein GET-Request mit folgendem Format:

`https://lexparency.de/_botapi/recents.json?date_from=YYYY-MM-DD&page=P`

z.B.:

https://lexparency.de/_botapi/recents.json?date_from=2020-10-01

wobei

- „YYYY-MM-DD“ einen Platzhalter für ein Datum darstellt ab welchem man die Updates abfragen möchte,
- „P“ einen ganzzahligen Wert für die Seite der Abfrage darstellt. Dies ist notwendig, falls man, z.B., die Updates eines ganzen Monats abfragt. Da das gesamte Ergebnis nicht vollständig mit einer einzigen Antwort zurückgegeben wird, sondern in Seiten aufgeteilt.

Die Antwort ist ein JSON-Dokument. Das Schema kann unter folgender URL abgefragt werden:

https://lexparency.de/static/botapi_recents.schema.json

Eine Schema-Validierung findet Serverseitig statt.

Folgende Graphik soll die einzelnen Bestandteile der Antwort erklären:



Die Dokument Abfrage

Hierüber erhält man die eigentlichen aktualisierten Inhalte – worüber man über die Recents-Abfrage informiert wurde.

Die Abfrage ist ein GET-Request mit folgendem Format:

`https://lexparency.de/_botapi/<CELEX>.json`

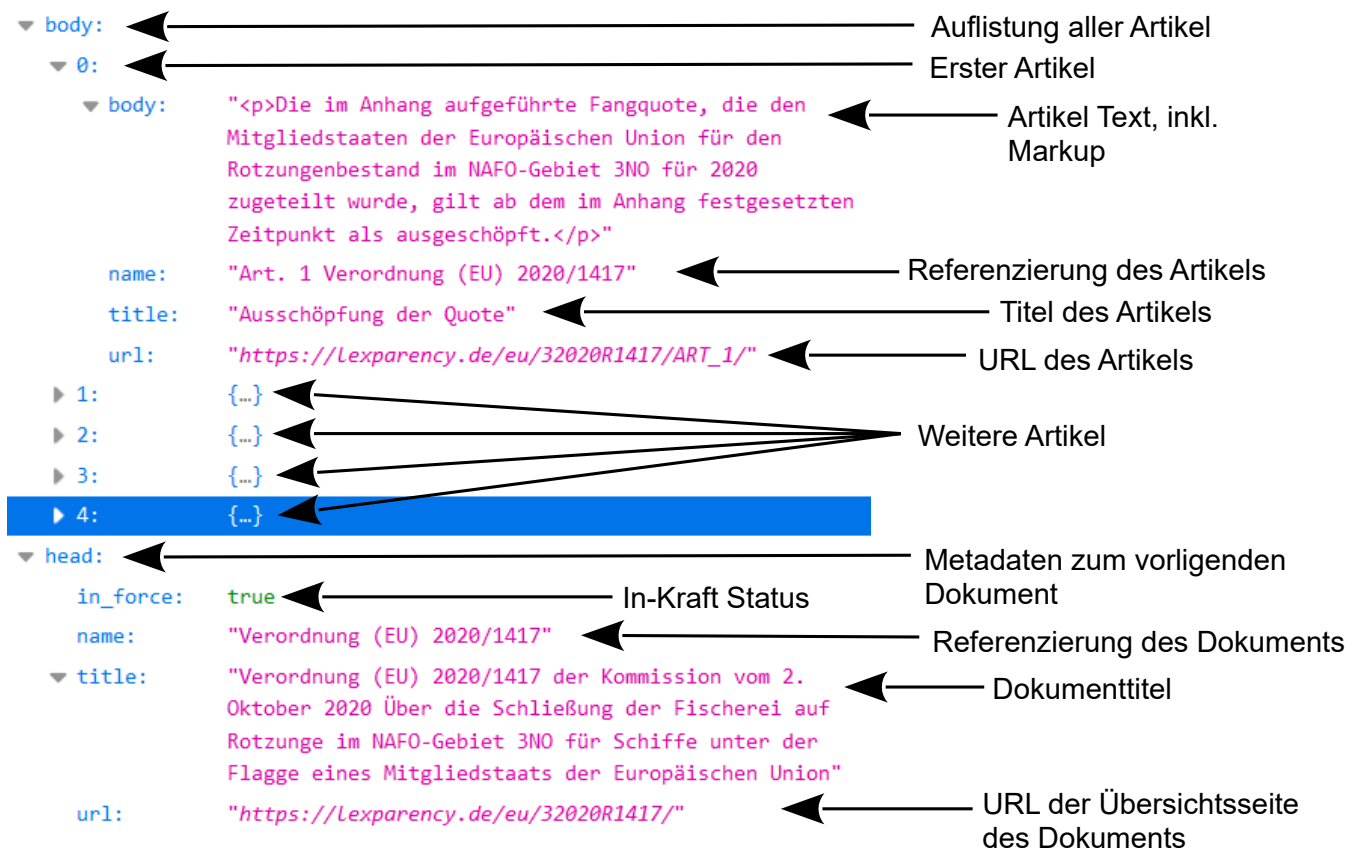
z.B.:

https://lexparency.de/_botapi/32016R1675.json

Die Antwort ist ein JSON-Dokument. Das zugehörige Schema kann unter folgender URL abgefragt werden:

https://lexparency.de/static/botapi_doc.schema.json

Folgende Graphik soll die Zusammensetzung der Antwort erklären:



Entsprechung zu den Suchtreffern auf deJure.org

The screenshot shows a legal document viewer on the left and a JSON API response on the right. Blue arrows indicate the mapping between the document content and the JSON fields:

- The document title "IRB-Ansatzes" maps to the JSON field `name`.
- The document subtitle "§ 6b KWG" maps to the JSON field `title`.
- The document content "1. die Ergebnisse der internen Stress-Tests eines Instituts, das einen IRB-Ansatz verwendet oder das zur Berechnung der in den Artikeln 362 bis 377 der Verordnung (EU) Nr. 575/2013 in der jeweils geltenden Fassung festgelegt ..." maps to the JSON field `body`.
- The document footer "OVG Hamburg, 11.01.2018 - 1 Bf 81/17" maps to the JSON field `url`.

Single-Step Option

Statt die vollen aktualisierten Dokumente über die zwei Schritte abzufragen, kann man auch direkt die Änderungen in der recents-Anfrage einbinden lassen – via des Queryparameters „single_step“.

`https://lexparency.de/_botapi/recents.json?date_from=YYYY-MM-DD&page=P&single_step`

In dem Fall werden nicht alle Artikel je dokument ausgegeben, sondern nur diejenigen mit einer Änderung. Die Art der Änderung wird in dem Attribut „transaction_type“ angegeben.

```

current_page: 1
hits:
  0:
    body:
      0: {}
      1:
        body: "<div class=\"normal\"> <_mmen wird.</div> </div>"
        name: "Art. 3 Verordnung (EG) Nr. 999/2001"
        title: "Begriffsbestimmungen"
        transaction_type: "update"
        url: "https://Lexparency.de/eu/32001R0999/ART_3/"
      2: {}
    bot_url: "https://Lexparency.de/_botapi/32001R0999.json"
    date_document: "2020-11-19"
    id_local: "32001R0999"
    in_force: true
    name: "Verordnung (EG) Nr. 999/2001"
    title: "Verordnung (EG) Nr. 999/_former Enzephalopathien"
    url: "https://Lexparency.de/eu/32001R0999/"
total_pages: 12

```

In dem gezeigten Beispiel ist ein Artikel mit `transaction_type = update` dargestellt. Folgende drei Ausprägungen für `transaction_type` sind vorgesehen:

- `insert`: Artikel ist neu. Z.B. durch eine Änderungsvorschrift oder bei einem neuen Dokument.

- update: Artikel existiert bereits länger. Der Inhalt wurde jedoch durch eine Änderungsvorschrift geändert.
- delete: Artikel wird durch Änderung aufgehoben. In diesem Fall sind die Attribute body, name und title enthalten. Also ausschließlich url.

Statusübersicht aller Dokumente

Die Bot-API bietet leider keine Auskunft, wenn ein Dokument vollständig außer Kraft gesetzt wird, da das „außer-kraft-seit“ Prädikat von den Eur-Lex APIs leider nur unvollständig abrufbar ist, und somit auch im Lexparency Datenmodell nicht zuverlässig zum Filtern verwendet werden kann. Stattdessen kann man durch eine separate Abfrage den status aller dokumente abfragen:

<https://lexparency.de/dbmap.json&page=P>

Die Antwort ist ein JSON Dokument von der folgenden Form:

```

current_page: 1
date: "2020-12-16"
hits:
  0:
    id_local: "31976L0372"
    in_force: false
    versions:
      0:
        available: true
        date_document: "1976-03-01"
        version: "initial"
      1:
        available: false
        date_document: "1981-08-19"
        version: "19810819"
      2:
        available: false
        date_document: "1992-11-10"
        version: "19921110"
      3:
        available: false
        date_document: "1994-05-03"
        version: "19940503"
    1: {}
    2: {}
    95: {}
total: 96
total_pages: 1
  
```

Annotations:

- Seite der Abfrage (points to `current_page: 1`)
- Datum der Abfrage (points to `date: "2020-12-16"`)
- Celex ID (points to `id_local: "31976L0372"`)
- In-Kraft Status (points to `in_force: false`)
- Details zu den Dokumentversionen. Jeder Listeneintrag enthält die Attribute:
 - available: Bool'scher Wert zeigt an, ob die Version erfolgreich hochgeladen wurde.
 - date_document: Datum der Version
 - version: Versions-ID
- Gesamtzahl der Dokumente (points to `total: 96`)
- Gesamtzahl der Seiten (points to `total_pages: 1`)

Analog zur „recents“-Abfrage muss man mehrere Requests stellen

Mögliche Ausbaustufen

- Recents Abfrage:

- Filter auf Dokumenttyp in der Recents-Abfrage: Verordnung, Beschluss, Richtlinie
- Dokument Abfrage
 - Einbeziehung der Erwägungsgründe – aktuell noch nicht in der Dokument-Abfrage enthalten.
 - Einbeziehung des Inhaltsverzeichnisses
 - Weitere Metadaten