

Abgeordnetenhaus REST API

Einstieg und Anwendungshandbuch

BETA · 2026-09-01

Überblick

Die Abgeordnetenhaus REST API stellt die Metadaten der Parlamentsdokumentation (PARDOK) des Abgeordnetenhauses von Berlin maschinenlesbar bereit. Antworten werden standardmäßig im JSON-Format ausgegeben; alternativ steht XML zur Verfügung. Sämtliche Endpunkte sind über die nachfolgend angegebene Basis-URL erreichbar.

Basis-URL: `https://pardok.parlament-berlin.de/api/v1`

Dieses Dokument richtet sich an alle, die die Schnittstelle nutzen möchten - **Vorkenntnisse im Umgang mit APIs werden nicht vorausgesetzt**. Es erklärt, wie Sie einen Zugang erhalten, welche Daten zur Verfügung stehen und wie die typischen Abläufe aussehen. Die **vollständige technische Referenz** aller Endpunkte, Parameter und Datenfelder finden Sie in der stets aktuellen Schnittstellenbeschreibung (Swagger UI und ReDoc, siehe [Swagger UI und ReDoc als vollständige Referenz](#)).

Die API befindet sich derzeit im **BETA-Modus**. Änderungen an Endpunkten, Parametern oder Datenstrukturen sind während dieser Phase möglich. Hinweise und Feedback nehmen wir unter opendata@parlament-berlin.de gern entgegen.

Inhaltsverzeichnis

1. [Überblick](#)
2. [Erste Schritte: Was diese Schnittstelle bereitstellt](#) · [Zugang und API-Schlüssel](#) · [Ihre erste Abfrage](#) · [Die Antwort verstehen](#) · [Swagger UI und ReDoc als vollständige Referenz](#)
3. [Welche Daten gibt es?: Die fünf Datenarten](#) · [Wie die Daten zusammenhängen](#) · [IDs und ihre Präfixe](#)
4. [Typische Nutzungsszenarien: Einzelne Entität abrufen](#) · [Listen abfragen und filtern](#) · [Alle Treffer laden](#) · [Verknüpfte Entitäten auflösen](#) · [Volltextsuche](#) · [Aktualisierungen abholen](#)
5. [Regeln und Grenzen: Seitenweiser Abruf \(Pagination\)](#) · [Ratenbegrenzung](#) · [Antwortformate](#) · [Sortierung](#) · [HTTP-Methoden](#) · [Fehler verstehen](#)
6. [Glossar](#)
7. [Anhang: Dokumenttypen und Vorgangspositionen: Als Drucksache geführte Dokumenttypen](#) · [Nicht als Drucksache geführte Typen](#)
8. [BETA-Phase und Änderungshistorie: Änderungen während der BETA-Phase](#) · [Änderungshistorie](#)

Erste Schritte

Was diese Schnittstelle bereitstellt

Das Abgeordnetenhaus von Berlin dokumentiert seine parlamentarischen Materialien in der PARDOK-Datenbank - Drucksachen, Plenarprotokolle, Ausschussprotokolle und die zugehörigen Vorgänge, zurückreichend bis zur **11. Wahlperiode (1989)**. Diese Metadaten werden bislang als regelmäßig aktualisierte XML-Dateien je Wahlperiode zum Download bereitgestellt.

Mit der hier beschriebenen Schnittstelle können die Daten künftig zusätzlich gezielt abgefragt werden: Statt einen kompletten Datenbestand herunterzuladen und selbst zu durchsuchen, stellen Sie gezielte Anfragen und erhalten genau die Treffer, die Sie benötigen - etwa alle Anträge einer Fraktion in einer bestimmten Wahlperiode oder alle Dokumente zu einem Stichwort.

Die Daten stehen als **Open Data** zur Verfügung. Es gelten die [Nutzungsbedingungen des Abgeordnetenhauses von Berlin](#) (PDF), veröffentlicht auf der Seite, über die auch diese Schnittstelle bereitgestellt wird.

Zugang und API-Schlüssel

Jede Anfrage benötigt einen **API-Schlüssel**. Sie erhalten einen personalisierten API-Schlüssel auf Anfrage per E-Mail an opendata@parlament-berlin.de bzw. nutzen den auf der Open-Data-Seite der Parlamentsdokumentation zur Verfügung gestellten API-Schlüssel.

Der Schlüssel wird bei jeder Anfrage im HTTP-Kopfzeilenfeld `X-API-Key` mitgesendet - nicht als Bestandteil der Adresse. Er identifiziert Ihren Zugang und steuert, wie viele Anfragen Sie pro Minute stellen dürfen (siehe [Ratenbegrenzung](#)).

Fehlt der Schlüssel oder ist er ungültig, antwortet die Schnittstelle mit dem Fehlercode 403 (siehe [Fehler verstehen](#)).

Ihre erste Abfrage

Das folgende Beispiel ruft die zehn zuletzt dokumentierten Vorgänge ab. Ersetzen Sie `<Ihr-Schlüssel>` durch Ihren API-Schlüssel:

```
curl -H "X-API-Key: <Ihr-Schlüssel>" "https://pardok.parlament-berlin.de/api/v1/vorgang?limit=10"
```

`curl` ist ein auf den meisten Systemen vorinstalliertes Kommandozeilenwerkzeug. Wenn Sie lieber im Browser arbeiten, können Sie dieselbe Abfrage in der [Swagger UI](#) ausführen - dort tragen Sie den Schlüssel einmalig ein und starten Anfragen per Mausklick.

Die Antwort verstehen

Die Antwort auf eine Listenabfrage besteht immer aus denselben drei Bestandteilen:

```
{
  "numFound": 115494,
  "cursor": "anmMwjExNTQ5NCwyMDI1LjEyLjA1LExBSC5MSVNTSF81MDQ2OTVijmozYruxPw",
  "documents": [
    {
      "typ": [ { "main": "Vorgang" } ],
      "id": [ { "main": "V-441658" } ],
      "vorgangstyp": [ { "main": "Antrag", "n": "1" } ],
      "wahlperiode": [ { "main": "19" } ],
      "vorgangsnummer": [ { "main": "19/1//2820/" } ],
      "aktualisiert": [ { "D": "2025 12 09", "T": "13:29:44" } ],
      "datum": "2025.12.08"
    }
  ]
}
```

- **numFound** - die exakte Gesamtzahl der Treffer Ihrer Abfrage, nicht die Anzahl der gelieferten Einträge.
- **documents** - die aktuelle Ergebnisseite. Standardmäßig 10 Einträge, maximal 200 (siehe [Seitenweiser Abruf \(Pagination\)](#)).
- **cursor** - die Lesemarke für die nächste Seite. Fehlt dieses Feld, haben Sie die letzte Seite erreicht.

Zwei Eigenheiten der Feldstruktur, die beim ersten Kontakt überraschen:

- **Feldwerte sind Listen von Objekten**, auch wenn nur ein Wert vorhanden ist. Der eigentliche Wert steht in `main` - `wahlperiode` liefert also nicht "19", sondern [{ "main": "19" }]. Grund ist, dass viele Felder mehrfach belegt sein können (etwa mehrere Urheber). Zusätzliche Schlüssel neben `main` enthalten Ergänzungen zum jeweiligen Wert.
- **Zeitstempel sind auf zwei Felder aufgeteilt**: `D` für das Datum, `T` für die Uhrzeit.

Bei Einzelabrufen entfällt der Rahmen - Sie erhalten direkt das angefragte Dokument.

Swagger UI und ReDoc als vollständige Referenz

Die Schnittstelle beschreibt sich selbst. Diese Selbstbeschreibung ist die **führende, fortlaufend aktualisierte und abschließende Dokumentation**: Funktionen und Abfragemöglichkeiten, die dort nicht aufgeführt sind, existieren auch nicht. Sie steht in vier Ausprägungen bereit:

Ansicht	Adresse	Wofür
Swagger UI	/api/v1/swagger-ui	Anfragen direkt im Browser ausprobieren
ReDoc	/api/v1/redoc	Strukturiert lesen und nachschlagen
OpenAPI (JSON)	/api/v1/openapi.json	Maschinenlesbar, z. B. zur Code-Erzeugung
OpenAPI (YAML)	/api/v1/openapi.yaml	Wie JSON, in kompakterer Schreibweise

Swagger UI und ReDoc zeigen dieselbe Beschreibung in zwei Darstellungen: Swagger UI ist auf das Ausführen von Anfragen ausgelegt, ReDoc auf das Lesen.

So schlagen Sie einen Filter oder ein Feld nach:

1. Öffnen Sie die Swagger UI und klicken Sie oben rechts auf **Authorize**. Tragen Sie dort Ihren API-Schlüssel ein - er gilt anschließend für alle Testabfragen dieser Sitzung.
2. Wählen Sie in der Liste den Endpunkt aus, der Sie interessiert, etwa `GET /drucksache`. Ein Klick klappt ihn auf.
3. Unter **Parameters** stehen alle Filter dieses Endpunkts mit Beschreibung, Datentyp und - wo es eine feste Werteliste gibt - den zulässigen Werten.
4. **Try it out** macht die Felder ausfüllbar, **Execute** führt die Anfrage aus. Darunter erscheinen die fertige Adresse, die Antwort und der zugehörige `curl`-Aufruf, den Sie in Ihre eigene Anwendung übernehmen können.
5. Unter **Schemas** am Seitenende finden Sie den Aufbau der Antworten mit allen Feldern.

Das vorliegende Dokument erklärt die Zusammenhänge und die typischen Abläufe; die Detailangaben zu einzelnen Parametern und Feldern schlagen Sie in der Swagger UI oder in ReDoc nach.

Welche Daten gibt es?

Die fünf Datenarten

Datenart	Endpunkt	Was Sie darin finden
Vorgang	/vorgang	Die parlamentarische Klammer: ein Beratungsverfahren von der Initiative bis zum Abschluss, mit allen zugehörigen Dokumenten. Der übliche Einstieg, wenn Sie ein Thema als Ganzes verfolgen wollen.
Drucksache	/drucksache	Das einzelne Schriftstück - Anträge, Gesetzentwürfe, Anfragen und deren Antworten, Vorlagen, Berichte, Beschlussempfehlungen.
Plenarprotokoll	/plenarprotokoll	Die Wortprotokolle der Plenarsitzungen.
Ausschussprotokoll	/ausschussprotokolle	Die Protokolle der Ausschusssitzungen.
Aktivität	/aktivitaet	Redebeiträge - Plenarprotokolle, aufgeschlüsselt nach den beteiligten Rednerinnen und Rednern. Der Zugang, wenn Sie nach Personen statt nach Dokumenten suchen.

Die Drucksachen unterscheiden sich nach ihrem Typ - **Anfragen** und **Antworten**, **Anträge** (einschließlich Gesetzentwürfen und Änderungsanträgen), **Vorlagen** und **Mitteilungen** des Senats, **Ausschussdokumente** wie Beschlussempfehlungen und Ausschussberichte sowie **Wahlvorschläge**. Über den Filter `f.drucksachtyp` lässt sich gezielt nach einem Typ suchen; die vollständige Werteliste finden Sie im [Anhang](#).

Wie die Daten zusammenhängen

Ein **Vorgang** bündelt die Dokumente eines Beratungsverfahrens. In der Antwort eines Vorgangs stehen die zugehörigen Dokumente im Feld `unterlagen` - ein Antrag samt der zugehörigen Plenarbehandlung erscheint also gemeinsam an einer Stelle. Umgekehrt verweist jedes Dokument über das Feld `vorgangsbezug` auf seinen Vorgang.

An Vorgängen können Sie zusätzlich die zugehörigen **Angaben aus dem Lobbyregister** abrufen: Senden Sie an `/vorgang` oder `/vorgang/{id}` den Parameter `f.beteiligungsbeitraege_anzeigen=true` mit, enthält die Antwort vorliegende Beteiligungsbeiträge als zusätzliches Feld `beteiligungsbeitraege`. Ohne diesen Parameter bleibt die Antwort unverändert.

Die eingebetteten Unterlagen tragen ein **verkürztes Feldset** - unter anderem fehlen Titel und Volltext. Wenn Sie das vollständige Dokument brauchen, holen Sie es über einen zweiten Aufruf (siehe [Verknüpfte Entitäten auflösen](#)).

IDs und ihre Präfixe

Jede Entität hat eine ID, über die sie direkt abrufbar ist. Zwei Punkte dazu ersparen Ihnen Fehlsuchen:

- **Vorgangs-IDs** tragen unterschiedliche Präfixe - V-, B-, C-, E-, A-. Das Präfix ist **keine Typkennung**; alle diese IDs werden über `/vorgang/{id}` abgerufen.

- **Dokument-IDs beginnen immer mit D-** - unabhängig davon, ob es sich um eine Drucksache, ein Plenar- oder ein Ausschussprotokoll handelt. Welcher der drei Endpunkte zuständig ist, steht im Feld `dokumentart` (D, P oder A), nicht im Präfix.

Typische Nutzungsszenarien

Dieses Kapitel zeigt die üblichen Abläufe anhand vollständiger Beispiele. Ersetzen Sie in allen Beispielen `<Ihr-Schlüssel>` durch Ihren API-Schlüssel.

Einzelne Entität abrufen

Jede Entität ist über ihre ID direkt abrufbar; die Antwort enthält unmittelbar das Dokument:

```
GET https://pardok.parlament-berlin.de/api/v1/{datenart}/{id}
```

Beispiel - die Metadaten eines Antrags:

```
curl -H "X-API-Key: <Ihr-Schlüssel>" "https://pardok.parlament-berlin.de/api/v1/vorgang/V-441658"
```

Antwort (gekürzt):

```
{
  "typ": [ { "main": "Vorgang" } ],
  "id": [ { "main": "V-441658" } ],
  "vorgangstyp": [ { "main": "Antrag", "n": "1" } ],
  "wahlperiode": [ { "main": "19" } ],
  "vorgangsnummer": [ { "main": "19/1//2820/" } ],
  "aktualisiert": [ { "D": "2025 12 09", "T": "13:29:44" } ],
  "datum": "2025.12.08",
  "angelegt": [ { "D": "2025 12 09", "T": "13:21:59" } ],
  "vorgangsdeskriptor": [
    { "titel": "Unterstützung von Wahlhelfern", "deskriptor": "Wahl" }
  ],
  "nebeneintrag": [
    { "deskriptor": "Ehrenamtliche Tätigkeit" },
    { "deskriptor": "Rechtsschutz" },
    { "deskriptor": "Wahlrecht" }
  ],
  "unterlagen": [
    {
      "id": [ { "main": "D-441660" } ],
      "typ": [ { "main": "DOKDBE" } ],
      "dokumentart": [ { "main": "P" } ],
      "drucksachentyp": [ { "main": "Behandlung im Plenum" } ],
      "wahlperiode": [ { "main": "19" } ],
      "fundstelle": [ { "main": " Behandlung im Plenum" } ]
    },
    {
      "id": [ { "main": "D-441659" } ],
      "typ": [ { "main": "DOKDBE" } ],
      "dokumentart": [ { "main": "D" } ],
      "drucksachentyp": [ { "main": "Antrag" } ],
      "dokumentnummer": [ { "main": "2820" } ],
      "wahlperiode": [ { "main": "19" } ],
      "urheber_koerperschaft": [
        { "main": "CDU", "Y": "7082" },
        { "main": "SPD", "Y": "7081" }
      ],
      "fundstelle": [ { "main": " Antrag CDU, SPD Drucksache 19/2820 vom 08.12.2025" } ],
      "datum": [ { "main": "08.12.2025" } ],
      "pdf_url": [ { "main": "https://pardok.parlament-berlin.de/starweb/adis/cit" } ]
    }
  ]
}
```

Gut zu erkennen: Der Vorgang „Unterstützung von Wahlhelfern“ bündelt zwei Dokumente - den Antrag selbst (dokumentart D) und dessen Behandlung im Plenum (dokumentart P).

Existiert keine Entität mit der angegebenen ID, antwortet die Schnittstelle mit dem Fehlercode 404.

Listen abfragen und filtern

Listen-Endpunkte liefern Treffer zu beliebig kombinierbaren Filtern. Filter beginnen mit `f.` und werden an die Adresse angehängt; **mehrere unterschiedliche Filter wirken als UND-Verknüpfung**.

Beispiel - alle Anträge der 19. Wahlperiode ab dem 01.11.2025:

```
curl -H "X-API-Key: <Ihr-Schlüssel>" "https://pardok.parlament-berlin.de/api/v1/drucksache?f.wahlperiode=19&f.drucksachtyp=Antrag&f.datum.start=2025-11-01"
```

So sieht ein vollständiger Eintrag der Drucksachenliste aus:

```
{
  "numFound": 181940,
  "cursor": "anmMwjE4MTk0MCwyMDI1LjEyLjA4LExBSC5MSVNTSF81MDQ3MzK5yZTtR47jXw",
  "documents": [
    {
      "typ": [ { "main": "DOKDBE" } ],
      "dokumentart": [ { "main": "D" } ],
      "dokumentnummer": [ { "main": "2820" } ],
      "drucksachtyp": [ { "main": "Antrag" } ],
      "id": [ { "main": "D-441659" } ],
      "titel": [ { "main": "", "t": "Unterstützung von Wahlhelfern", "d": "Wahl"
    } ],
      "urheber_koerperschaft": [
        { "main": "CDU", "Y": "7082" },
        { "main": "SPD", "Y": "7081" }
      ],
      "wahlperiode": [ { "main": "19" } ],
      "fundstelle": [ { "main": "Antrag CDU, SPD Drucksache 19/2820 vom 08.12.2025" } ],
      "datum": "2025.12.08",
      "aktualisiert": [ { "D": "2025 12 09", "T": "13:29:44" } ],
      "angelegt": [ { "D": "2025 12 09", "T": "13:22:19" } ],
      "vorgangsbezug": [ { "main": "V-441658", "p": "0000" } ],
      "pdf_url": [ { "main": "https://pardok.parlament-berlin.de/starweb/adis/cit" } ]
    }
  ]
}
```

Drei Hinweise, die häufige Missverständnisse vermeiden:

- **Ein unbekannter Filterwert erzeugt keinen Fehler.** Bis auf wenige Ausnahmen prüft die Schnittstelle die Werte nicht - ein Tippfehler liefert `numFound: 0` statt einer Fehlermeldung. Bei unerwartet leeren Ergebnissen prüfen Sie deshalb zuerst die Schreibweise Ihrer Filterwerte.
- **Werte müssen exakt übereinstimmen** (Groß- und Kleinschreibung ausgenommen). Wo es historische Schreibvarianten gibt, müssen beide abgefragt werden - etwa `Antwort` (Schlussbericht) **neben** `Antwort` (Schlußbericht).
- **Die Reihenfolge der Treffer ist fest** und nicht über Parameter änderbar (siehe [Sortierung](#)).

Alle Treffer laden

Übersteigt `numFound` die Seitengröße, laden Sie die weiteren Treffer über Folgeanfragen. Das Verfahren heißt **Cursor-Pagination** und funktioniert wie eine Lesemarke: Sie wiederholen **alle Parameter der ursprünglichen Anfrage** und ergänzen den `cursor`-Wert aus der letzten Antwort. Das wiederholen Sie so lange, bis das Feld `cursor` in der Antwort **fehlt** - das ist das Ende.

Das folgende Beispiel lädt alle Anträge der 19. Wahlperiode: 922 Treffer, also fünf Seiten zu je 200 (die letzte mit 122).

1. Anfrage - erste Seite:

```
curl -H "X-API-Key: <Ihr-Schlüssel>" "https://pardok.parlament-berlin.de/api/v1/drucksache?f.wahlperiode=19&f.drucksachtyp=Antrag&limit=200"
```

```
{
  "numFound": 922,
  "cursor": "anmOUDkyMiwymDI1LjAyLjA0LExBSC5MSVNTSF80OTE0NDSfhQ28Fe31JQ",
  "documents": [ "... 200 Einträge ..." ]
}
```

1. Anfrage - zweite Seite, mit Cursor und **denselben Filtern**:

```
curl -H "X-API-Key: <Ihr-Schlüssel>" "https://pardok.parlament-berlin.de/api/v1/drucksache?f.wahlperiode=19&f.drucksachtyp=Antrag&limit=200&cursor=anmOUDkyMiwymDI1LjAyLjA0LExBSC5MSVNTSF80OTE0NDSfhQ28Fe31JQ"
```

Die Seiten 3 und 4 folgen demselben Muster. In der Antwort auf die fünfte Anfrage fehlt das Feld `cursor` - Sie sind am Ende:

```
{
  "numFound": 922,
  "documents": [ "... 122 Einträge ..." ]
}
```

Damit das zuverlässig funktioniert:

- **Alle Filter bei jeder Folgeanfrage erneut mitsenden.** Der Cursor merkt sich nur die Position, nicht die Abfrage - fehlt ein Filter, stammt die nächste Seite unbemerkt aus einer anderen Treffermenge.
- **Cursor sind 15 Minuten gültig.** Führen Sie Folgeanfragen zügig aus; bei einem abgelaufenen Cursor antwortet die Schnittstelle mit Fehlercode 400, und Sie beginnen ohne Cursor von vorn.
- **Behandeln Sie den Cursor als undurchsichtige Zeichenkette** - nicht auswerten, nicht selbst zusammenbauen, nicht dauerhaft speichern.
- **Große Seiten sind günstiger als viele kleine.** Eine Anfrage mit `limit=200` kostet kaum mehr Zeit als eine mit `limit=10`; auch tiefes Blättern bleibt gleichmäßig schnell.

Verknüpfte Entitäten auflösen

Die in einem Vorgang eingebetteten Unterlagen tragen ein verkürztes Feldset. Um ein Dokument vollständig zu laden, brauchen Sie zwei Angaben: seine **ID** und den **passenden Endpunkt**. Welcher

Endpunkt der richtige ist, lässt sich der ID nicht ansehen - Drucksachen, Plenar- und Ausschussprotokolle tragen alle dasselbe Präfix `D-`. Maßgeblich ist stattdessen das Feld `dokumentart` des eingebetteten Eintrags:

<code>dokumentart</code>	Endpunkt
<code>D</code>	<code>GET /drucksache/{id}</code>
<code>P</code>	<code>GET /plenarprotokoll/{id}</code>
<code>A</code>	<code>GET /ausschussprotokolle/{id}</code>

Beispiel - aus dem Vorgang oben das verknüpfte Dokument `D-441659` (dokumentart `D`) vollständig laden:

```
curl -H "X-API-Key: <Ihr-Schlüssel>" "https://pardok.parlament-berlin.de/api/v1/drucksache/D-441659"
```

In der Gegenrichtung führt das Feld `vorgangsbezug` eines Dokuments zurück zum Vorgang.

Besonderheit bei Aktivitäten: Das Feld `redner` enthält normalerweise alle Rednerinnen und Redner. Filtern Sie jedoch nach einer Person, enthält es bewusst nur die passenden Einträge - dieselbe Aktivität liefert also je nach Abfrage unterschiedliche `redner`-Inhalte.

Volltextsuche

Die Endpunkte `/drucksache`, `/plenarprotokoll` und `/ausschussprotokolle` durchsuchen mit dem Filter `f.volltext` den Text der Dokumente. Angegeben wird **ein Suchbegriff oder eine Phrase**; gesucht wird nach Zeichenfolgen an beliebiger Stelle im Text, unabhängig von Groß- und Kleinschreibung.

```
curl -H "X-API-Key: <Ihr-Schlüssel>" "https://pardok.parlament-berlin.de/api/v1/plenarprotokoll?f.volltext=Wahlhelfer&limit=200"
```

Mit dem zusätzlichen Filter `f.volltext_anzeigen=true` wird der gefundene Volltext als Feld in die Antwort aufgenommen.

Zwei Punkte zur Antwortzeit, die für die Planung Ihrer Anwendung wichtig sind:

- **Volltextsuchen laufen im Mehrsekundenbereich.** Sehen Sie für diese Abfragen einen **Zeitüberschreitungswert von mindestens 60 Sekunden** in Ihrem Programm vor und führen Sie sie nicht mehrfach parallel aus.
- **Die Dauer hängt nicht von der Trefferzahl ab, sondern vom Suchbegriff.** Maßgeblich ist, wie häufig der Wortanfang im gesamten Textbestand vorkommt: Ein Begriff mit sehr wenigen Treffern kann deutlich länger dauern als ein Begriff mit vielen. Ein Suchwort ganz ohne Treffer kostet ebenso viel Zeit. Wählen Sie deshalb möglichst kennzeichnende, seltene Begriffe - und schließen Sie aus einer langen Antwortzeit nicht auf eine große Treffermenge.

Aktualisierungen abholen

Wenn Sie einen eigenen Datenbestand aufbauen und aktuell halten möchten, rufen Sie die Endpunkte in festen Abständen mit einer zeitlichen Eingrenzung ab. Jede Entität trägt dazu die Felder `angelegt` und `aktualisiert`.

Beispiel - Drucksachen ab dem 09.12.2025, 09:00 Uhr:

```
curl -H "X-API-Key: <Ihr-Schlüssel>" "https://pardok.parlament-berlin.de/api/v1/drucksache?f.aktualisiert.start=2025-12-09T09:00:00"
```

Zwei Hinweise für einen verlässlichen Abgleich:

- **Dokumente einzeln verfolgen.** Der Zeitstempel eines Vorgangs ändert sich nur bei Änderungen am Vorgang selbst, nicht wenn sich eines seiner Dokumente ändert. Fragen Sie `/drucksache`, `/plenarprotokoll` und `/ausschussprotokolle` daher getrennt ab; die in Vorgängen eingebetteten Unterlagen geben stets den aktuellen Stand wieder.
- **Verzögerung einplanen.** Änderungen in PARDOK erscheinen mit einigen Minuten Verzögerung in der Schnittstelle.

Fragen Sie mit überlappenden Zeitfenstern ab, erhalten Sie bereits bekannte Einträge erneut - entfernen Sie Dubletten anhand der `id`.

Regeln und Grenzen

Seitenweiser Abruf (Pagination)

Eine Anfrage an einen Listen-Endpunkt liefert nie die gesamte Treffermenge auf einmal, sondern immer nur eine einzelne Ergebnisseite mit **maximal 200 Entitäten** (`limit`: Minimum 1, Standard 10, Maximum 200; Werte außerhalb dieses Bereichs werden mit Fehlercode 422 abgelehnt). Umfasst Ihre Abfrage mehr Treffer, rufen Sie die weiteren Seiten nacheinander ab - wie das geht, steht unter [Alle Treffer laden](#).

Ratenbegrenzung

Die Ratenbegrenzung wird **pro API-Schlüssel** berechnet. Jeder Schlüssel kann ein Limit an Anfragen pro Minute besitzen; Schlüssel ohne konfiguriertes Limit sind unbeschränkt. Bei Überschreitung antwortet die Schnittstelle mit Fehlercode 429 und nennt in der Meldung das Limit Ihres Schlüssels.

Antworten zu Schlüsseln mit Limit tragen die Kopfzeilenfelder `X-RateLimit-Limit` und `X-RateLimit-Remaining`, eine 429-Antwort zusätzlich `Retry-After` (Sekunden bis zum nächsten Zeitfenster). Richten Sie die Anfragefrequenz Ihres Programms an `X-RateLimit-Remaining` aus und warten Sie bei einer 429-Antwort die in `Retry-After` genannte Zeit ab.

Wichtig: Das Limit gilt für ein gleitendes Fenster von 60 Sekunden. Ein Programm, das eine 429-Antwort ignoriert und einfach weiter anfragt, hält sein Kontingent dauerhaft ausgeschöpft und bleibt gesperrt.

Empfehlung: höchstens **fünf gleichzeitige Anfragen** pro Anwendung. Dieser Wert wird nicht erzwungen.

Antwortformate

Antworten erfolgen standardmäßig als JSON. Für XML senden Sie das Kopfzeilenfeld `Accept: application/xml` oder hängen `format=xml` an die Adresse an. Die XML-Antwort trägt dieselbe Struktur unter einem `<response>`-Wurzelelement, Listenwerte als `<item>`-Elemente.

Sortierung

Ergebnisse von Listen-Endpunkten sind **nach Dokumentdatum absteigend** sortiert, bei gleichem Datum nach Datensatz-ID absteigend; Dokumente ohne Datum stehen am Ende. Diese feste Reihenfolge ist Voraussetzung der Cursor-Pagination und hält auch das Blättern über große Bestände schnell.

HTTP-Methoden

Die Schnittstelle ist rein lesend und unterstützt ausschließlich **GET**. Alle anderen Methoden - auch `HEAD` und `OPTIONS` - werden mit Fehlercode 405 abgelehnt.

Fehler verstehen

Fehler werden über den HTTP-Statuscode gemeldet; die Antwort enthält im Feld `detail` eine erläuternde Meldung. **Die Fehlertexte sind englischsprachig.**

Code	Bedeutung	Was zu tun ist
400	Ungültige Anfrage - abgelaufener oder fehlerhafter Cursor, falsches Datumsformat, ein für diesen Endpunkt nicht vorgesehener Filter	Parameter prüfen; bei abgelaufenem Cursor die Abfrage ohne Cursor neu starten
403	API-Schlüssel fehlt oder ist ungültig	Schreibweise <code>X-API-Key</code> und den Schlüsselwert prüfen
404	Es gibt keine Entität mit dieser ID	ID prüfen - insbesondere, ob sie zum aufgerufenen Endpunkt passt (siehe IDs und ihre Präfixe)
405	Eine andere Methode als GET wurde verwendet	Anfrage als GET stellen
422	Ein Parameterwert liegt außerhalb des zulässigen Bereichs, etwa <code>limit=999</code> oder ein unbekanntes <code>format</code>	Wert korrigieren
429	Ratenlimit überschritten	Die in <code>Retry-After</code> genannte Zeit abwarten (siehe Ratenbegrenzung)

Beispiel einer Fehlerantwort:

```
{ "detail": "limit: Input should be less than or equal to 200" }
```

Zwei Fälle, die in der Praxis für Verwirrung sorgen:

- **400 und 422 grenzen sich nicht nach Schwere ab, sondern nach Prüfebene.** Ungültiger Cursor, unbekannter Filter und falsches Datumsformat ergeben 400; unzulässige Werte für `limit` und `format` ergeben 422. Behandeln Sie in Ihrem Programm beide Codes.
- **Ein unerwarteter 429 kann am Schlüssel liegen.** Wiederholte Anfragen mit einem *ungültigen* Schlüssel werden gedrosselt: Ab der zweiten Anfrage innerhalb einer Minute antwortet die Schnittstelle mit 429 statt mit 403. Erhalten Sie 429, obwohl Sie das Limit nicht ausgeschöpft haben, prüfen Sie zuerst den Schlüssel.

Glossar

Begriff	Bedeutung
API / Schnittstelle	Ein Zugang, über den ein Programm Daten eines anderen Systems abrufen kann - im Unterschied zu einer Webseite, die für das Lesen durch Menschen gestaltet ist.
Endpunkt	Eine einzelne Adresse der Schnittstelle, die eine bestimmte Art von Daten liefert, z. B. /drucksache.
Parameter / Filter	Eine Angabe, die an die Adresse angehängt wird und die Ergebnismenge einschränkt, z. B. f.wahlperiode=19.
GET	Die HTTP-Methode zum Abrufen von Daten. Diese Schnittstelle unterstützt ausschließlich GET, verändert also nie Daten.
Statuscode	Eine dreistellige Zahl, mit der jede Antwort ihren Ausgang meldet - 200 für Erfolg, 4xx für einen Fehler auf Seiten der Anfrage.
JSON	Ein textbasiertes Datenformat, das von praktisch allen Programmiersprachen gelesen werden kann. Standardformat dieser Schnittstelle.
Kopfzeilenfeld (Header)	Eine Zusatzangabe, die mit einer Anfrage oder Antwort übermittelt wird, ohne Teil der Adresse zu sein - hier z. B. der API-Schlüssel in X-API-Key.
API-Schlüssel	Eine Zeichenfolge, die Ihren Zugang ausweist und Ihr Anfragekontingent bestimmt - als personalisierter Schlüssel per E-Mail oder als öffentlicher Schlüssel auf der Open-Data-Seite (siehe Zugang und API-Schlüssel).
Ratenbegrenzung (Rate Limit)	Die höchstzulässige Anzahl von Anfragen je Zeitraum.
Cursor	Eine Lesemarke, mit der die nächste Ergebnisseite angefordert wird.
Pagination	Die seitenweise Auslieferung großer Treffermengen.
Entität	Ein einzelner Datensatz - ein Vorgang, eine Drucksache, ein Protokoll, eine Aktivität.
Vorgang	Ein parlamentarisches Beratungsverfahren mit allen zugehörigen Dokumenten.
Wahlperiode	Die Legislaturperiode, der ein Dokument zugeordnet ist. Die Schnittstelle enthält Daten ab der 11. Wahlperiode (1989).
Deskriptor	Ein Schlagwort aus dem Thesaurus der Parlamentsdokumentation, mit dem Dokumente inhaltlich erschlossen werden.
Open Data	Frei zugängliche Daten, die unter veröffentlichten Bedingungen weiterverwendet werden dürfen.

Anhang: Dokumenttypen und Vorgangspositionen

Die folgenden Werte werden in der Parlamentsdokumentation für den Typ eines Dokuments bzw. für eine Vorgangsposition - eine Station im parlamentarischen Ablauf eines Vorgangs - verwendet. Sie sind die zulässige Eingabe für den Filter `f.drucksachtyp` (siehe [Die fünf Datenarten](#)).

Die Liste ist danach unterteilt, ob ein Typ in der Parlamentsdokumentation **als Drucksache geführt** wird. Davon hängt ab, über welchen Endpunkt Sie ihn finden: Als Drucksache geführte Typen liefert `/drucksache`. Alle übrigen Typen erreichen Sie über `/vorgang`, weil dort über sämtliche Dokumente eines Vorgangs gesucht wird.

Drei Hinweise zur Verwendung:

- **Die Schreibweise muss exakt übereinstimmen** (Groß- und Kleinschreibung ausgenommen). Ein abweichend geschriebener Wert wird nicht abgelehnt, sondern liefert `numFound: 0` - bei unerwartet leeren Ergebnissen lohnt daher der Abgleich mit dieser Liste.
- **Ähnlich benannte Typen können unterschiedlich einsortiert sein.** Schriftliche Anfrage wird als Drucksache geführt, Mündliche Anfrage nicht; Erklärungen § 66 GO und Erklärungen § 72 GO sind Drucksachen, Persönliche Bemerkungen § 65 GO nicht. Das ist kein Fehler in der Liste, sondern gibt die Praxis der Parlamentsdokumentation wieder.
- **Die Liste bildet den Bestand ab, nicht eine feste Auswahl.** Werte können im Zuge der parlamentarischen Praxis hinzukommen; Änderungen weisen wir in der [Änderungshistorie](#) aus.

Als Drucksache geführte Dokumenttypen

- Änderungsantrag
- Antrag
- Antrag (Gesetzentwurf)
- Antrag (Wahlvorschlag)
- Antwort
- Antwort (2. Zwischenbericht)
- Antwort (Schlussbericht)
- Antwort (Zwischenbericht)
- Ausschussbericht
- Ausschussbericht (Zwischenbericht)
- Bericht des Senats
- Beschlüsse zum parlamentarischen Verfahren
- Beschlussempfehlung
- Beschlussempfehlung und Ausschussbericht
- Erklärungen § 66 GO
- Erklärungen § 72 GO
- Mitteilung zur Kenntnisnahme
- Mitteilung zur Kenntnisnahme (Schlussbericht)
- Mitteilung zur Kenntnisnahme (Zwischenbericht)
- Neufassung
- Schriftliche Anfrage

- Unterrichtung
- Verordnung
- Vorlage zur Beschlussfassung
- Vorlage zur Beschlussfassung (Gesetzentwurf)
- Vorlage zur Beschlussfassung (Gesetzentwurf/Ergänzung)
- Vorlage zur Beschlussfassung gem. § 38 GO
- Vorlage zur Kenntnisnahme
- Wahlvorschlag

Nicht als Drucksache geführte Typen

- Aktuelle Stunde
- Ausschussberatung
- Ausschussbesprechung
- Ausschussbesprechung § 21 Abs. 3 GO
- Behandlung im Plenum
- Bekanntmachung (Gesetz- und Verordnungsblatt)
- Gesetz- und Verordnungsblatt
- I. Lesung
- I. und II. Lesung
- II. Lesung
- III. Lesung
- Mündliche Anfrage
- Nachrufe u. Gedenkreden
- Ordnungsruf
- Persönliche Bemerkungen § 65 GO
- Regierungserklärung

BETA-Phase und Änderungshistorie

Änderungen während der BETA-Phase

Anpassungen und Erweiterungen der Schnittstelle werden in dieser Dokumentation nachvollziehbar ausgewiesen. Änderungen mit Auswirkungen auf bestehende Schnittstellen werden einschließlich der betroffenen Endpunkte und Datenstrukturen dokumentiert. Nach Abschluss der BETA-Phase wird die Schnittstelle in einen regulären Versionsstand überführt.

Änderungshistorie

Diese Fassung ist der Ausgangsstand der Dokumentation. Künftige Änderungen an der Schnittstelle werden an dieser Stelle mit Datum und Dokumentstand ausgewiesen.