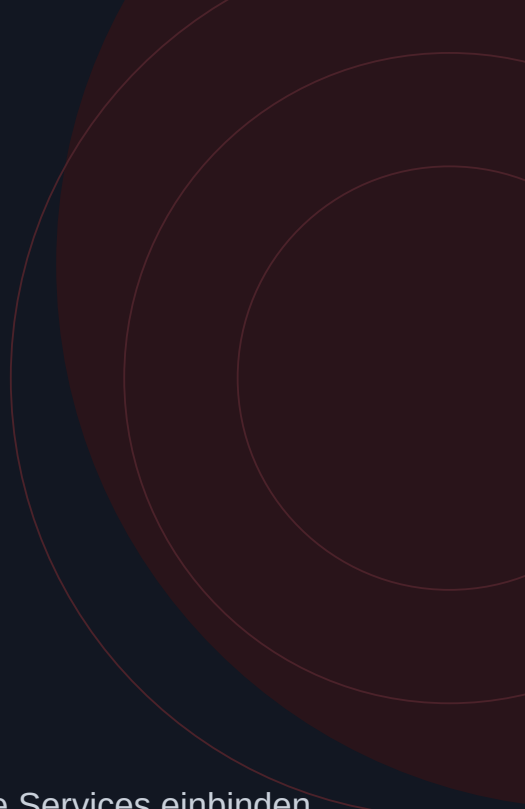




MBCT3

SNITTSTELLENDOKUMENTATION · VERSION 1

MBCT3 Die Lage API

Aktuelle und historische Lageinformationen sicher in andere Services einbinden.

Basis-URL	https://lage.mbct3.org/api/v1
Format	JSON über HTTPS · Read-only
Stand	4. September 2026 · Dienstversion 1.33.0

Entscheidungsunterstützung, kein amtliches Warnsystem.

Was diese API leistet

Die öffentliche API macht die bereits von MBCT3 gespeicherten Lagebilder und Kartenebenen maschinenlesbar. Sie ist für serverseitige Integrationen, interne Dashboards und kontrollierte Datenweitergabe gedacht.

Wichtig

Ein API-Aufruf löst keine Anfrage bei DWD, PEGELONLINE, Autobahn GmbH oder Umweltbundesamt aus. Die Schnittstelle liest ausschließlich aus der MBCT3-Datenbank. Der ausgewiesene Datenstand ist daher maßgeblich.

Inhalt dieser Dokumentation

Kapitel	Thema	Nutzen
2	Architektur und Datenweg	Trennung zwischen Sammler, Datenbank und öffentlicher API
3	Zugang und Schnellstart	API-Schlüssel, Scope und erste Anfrage
4	Endpunkte: Lage und Themen	Aktuelles Lagebild, Themenliste und Verlauf
5	Endpunkte: Historie und Kartenlayer	Pagination und gespeicherte Fachdaten
6	Antwortformat und Datenmodell	Envelope, Ampelwerte, Zeiten und Quellen
7	Cache, Limits und Fehler	ETag, 304, Rate-Limits und stabile Fehlercodes
8	Integrationsbeispiele	Serverseitige Beispiele und sichere Schlüsselhaltung
9	Quellen, Nutzung und Checkliste	Attribution, Grenzen und Produktionsfreigabe

Maschinenlesbare Referenz

Die vollständige OpenAPI-3.1-Beschreibung liegt unter </api/v1/openapi.json>. Sie ist die maßgebliche Referenz für Pfade, Parameter, Feldtypen und Fehlerantworten.

Datenweg und Sicherheitsgrenzen

Die API ist bewusst als schmale Lese-Schicht ausgeführt. Interne Schreib-, Login- und Sammlerendpunkte gehören nicht zum öffentlichen Vertrag.

1 · Externe Quellen	DWD, PEGELONLINE/WSV, Autobahn GmbH und UBA stellen Fachdaten bereit.
2 · Geschützter Hermes-Sammler	Der serverseitige Sammler ruft Quellen in festgelegten Intervallen ab, prüft Formate und speichert jeden Lauf.
3 · MBCT3-Datenbank	Aktueller Stand und append-only Historie werden getrennt gehalten. Radarbinärdaten liegen im eigenen Speicher.
4 · Öffentliche API v1	Authentifizierte GET-Anfragen lesen nur gespeicherte Datensätze und liefern stabile JSON-Antworten.
5 · Ihr Service	Ihre serverseitige Anwendung cached, verarbeitet und zeigt Daten mit Datenstand und Quellenhinweis an.

Sicherheitsprinzipien

Prinzip	Umsetzung
Read-only	Nur GET und OPTIONS; keine Veröffentlichung oder Änderung über /api/v1.
Minimale Schlüsselablage	Auf dem Server wird nur der SHA-256-Prüfwert des Schlüssels gespeichert.
Berechtigung	Jeder Datenendpunkt verlangt den Scope lage:read.
Kontrollierte Browsernutzung	CORS wird nur für ausdrücklich freigegebene Origins gesetzt.
Missbrauchsschutz	Zähler pro Schlüssel und Minute; keine IP-Adresse in der Rate-Limit-Tabelle.

Authentifizierung und Schnellstart

Ein API-Schlüssel wird separat vergeben. Behandeln Sie ihn wie ein Passwort und speichern Sie ihn nur in einer serverseitigen Secret-Verwaltung oder Umgebungsvariable.

Bevorzugter Header

```
Authorization: Bearer <API_KEY>
```

Alternative

```
X-API-Key: <API_KEY>
```

Aktuelles Lagebild abrufen

```
curl --fail --silent \  
-H "Authorization: Bearer <API_KEY>" \  
https://lage.mbct3.org/api/v1/lage/latest
```

Schlüssel nicht in Frontend-Code einbauen

Ein Schlüssel in JavaScript, einer mobilen App oder einem öffentlich einsehbaren Repository kann kopiert und missbraucht werden. Rufen Sie die MBCT3 API von Ihrem Backend ab und geben Sie an das Frontend nur die tatsächlich benötigten Daten weiter.

Zugangsfehler

HTTP	Code	Bedeutung
401	authentication_required	Kein Schlüssel übermittelt.
401	invalid_api_key	Schlüssel ungültig, rotiert oder widerrufen.
403	insufficient_scope	Schlüssel besitzt nicht den Scope lage:read.

Lage und Themen

Die Pfade werden relativ zur Basis-URL <https://lage.mbct3.org/api/v1> angegeben.

Methode und Pfad	Zweck	Wichtige Felder / Parameter
GET /status	Kompakter Betriebs- und Datenstatus.	service, service_version, latest_lage, layers
GET /lage/latest	Jüngstes veröffentlichtes Lagebild.	overall_severity, executive_summary, source_count, published_at, topics
GET /topics	Themen mit jeweils jüngster Beobachtung.	status=active all, severity=green yellow red, limit=1..100
GET /topics/{key}	Stammdaten und Verlauf eines Themas.	key im Pfad, limit=1..120

Beispiel: aktive erhöhte Themen

```
GET /topics?status=active&severity=yellow&limit=20
```

Themen-Schlüssel

Der Wert key ist die stabile technische Kennung eines Themas. Verwenden Sie ihn in Verweisen und für /topics/{key}. Überschrift und Bezeichnung können sich im Verlauf präzisieren.

Quellen je Beobachtung

Jede ausführliche Beobachtung enthält eine Liste sources mit Titel, URL, Herausgeber und - soweit vorhanden - Veröffentlichungszeitpunkt. Diese Links dienen der Nachvollziehbarkeit und sollten bei einer weitergehenden Darstellung erhalten bleiben.

Ampelwerte

green	Beobachten	Interne MBCT3-Kategorie
yellow	Erhöht	Interne MBCT3-Kategorie
red	Kritisch	Interne MBCT3-Kategorie

Historie und Kartenebenen

Historienseiten sind cursorbasiert. Kartenebenen liefern immer den letzten verwendbaren Stand aus der MBCT3-Datenbank.

Methode und Pfad	Zweck	Parameter / Werte
GET /history	Lagebilder, absteigend nach Zeit.	limit=1..100; before=ISO-8601
GET /layers	Zusammenfassung aller fünf Layer.	Keine Parameter
GET /layers/{layer}	Vollständiger gespeicherter Layer.	weather, water, traffic, wildfire, air

Historie fortsetzen

Enthält die Antwort `next_before`, wird dieser Wert unverändert als `before` in der nächsten Anfrage verwendet. So bleibt die Abfrage stabil, wenn zwischenzeitlich ein neues Lagebild erscheint.

```
GET /history?limit=24&before=2026-09-04T12:00:00+02:00
```

Layerstatus richtig behandeln

Status	Bedeutung	Empfohlene Darstellung
live	Aktueller gespeicherter Stand innerhalb des vorgesehenen Zeitfensters.	Datenstand normal anzeigen.
stale	Letzter verwendbarer Datenbankstand; neue Sammlung verspätet oder fehlgeschlagen.	Hinweis auf älteren Datenstand sichtbar machen.
unavailable	Noch kein verwendbarer gespeicherter Stand vorhanden.	Keine Nullwerte erfinden; Ebene als nicht verfügbar kennzeichnen.

Keine Vollständigkeitsannahme

Die Layer sind fachlich unterschiedlich. Auswahlregeln, Näherungsflächen, vorläufige Messwerte und Provider-Ausfälle sind auf der Quellenseite erläutert. Der Status beschreibt den technischen Datenstand, nicht die vollständige Gefahrenlage.

Einheitlicher Antwortumschlag

Geschützte Endpunkte verwenden denselben Envelope. Dadurch können Logging, Zeitprüfung und Fehlerbehandlung zentral implementiert werden.

```
{
  "api_version": "v1",
  "data_updated_at": "2026-09-04T12:00:00+02:00",
  "generated_at": "2026-09-04T12:01:03+02:00",
  "disclaimer": "Entscheidungsunterstützung, kein amtliches Warnsystem.",
  "links": { "self": "...", "documentation": "..." },
  "data": { }
}
```

Feld	Typ	Bedeutung
api_version	string	Vertragsversion der Schnittstelle; aktuell v1.
data_updated_at	date-time null	Fachlicher Stand der gelieferten Daten.
generated_at	date-time	Zeitpunkt, zu dem die Antwort erzeugt wurde.
disclaimer	string	Verbindlicher Nutzungshinweis.
links	object	Self-, Dokumentations-, OpenAPI- und Quellenlinks.
data	object / array	Endpunktspezifischer Inhalt.

Zeitangaben

Alle von der API normalisierten Zeitangaben verwenden ISO 8601 einschließlich Zeitzonenoffset. Prüfen Sie für Aktualitätsentscheidungen `data_updated_at` oder den layerbezogenen Wert `collected_at`, nicht `generated_at`.

Geografische Einordnung

Eine Beobachtung enthält `geography.scope`, `affected_regions` und `national_relevance`. Der Scope kann `local`, `regional`, `multi_state`, `nationwide` oder `unknown` sein.

Cache, Rate-Limits und Fehler

Effiziente Abfragen entlasten beide Systeme und halten Integrationen auch bei kurzfristigen Störungen vorhersehbar.

Bedingte Anfragen

Erfolgreiche Antworten enthalten einen ETag. Senden Sie ihn beim nächsten Abruf in If-None-Match. Wenn sich die Repräsentation nicht geändert hat, antwortet die API mit HTTP 304 ohne Body.

```
If-None-Match: "<ETAG_AUS_DER_VORHERIGEN_ANTWORT>"
```

Rate-Limit-Header

Header	Bedeutung
X-RateLimit-Limit	Zulässige Anfragen im aktuellen Minutenfenster.
X-RateLimit-Remaining	Noch verfügbare Anfragen im Fenster.
X-RateLimit-Reset	Unix-Zeitpunkt des nächsten Fensters.
Retry-After	Wartezeit in Sekunden bei HTTP 429.

Einheitliche Fehlerantwort

```
{
  "api_version": "v1",
  "error": {
    "status": 400,
    "code": "invalid_parameter",
    "message": "...",
    "request_id": "..."
  }
}
```

Protokollieren Sie bei Supportanfragen die request_id, aber niemals den API-Schlüssel. Bei 429 bis zum Reset warten; bei 503 mit begrenztem exponentiellem Backoff erneut versuchen.

Serverseitige Beispiele

Die Beispiele verwenden Platzhalter. Der echte Schlüssel gehört in eine geschützte Konfiguration und darf nicht in Logs oder Fehlermeldungen erscheinen.

JavaScript / Node.js (serverseitig)

```
const response = await fetch(
  'https://lage.mbct3.org/api/v1/lage/latest',
  { headers: { Authorization: `Bearer ${process.env.MBCT3_API_KEY}` } }
);
if (!response.ok) throw new Error(`MBCT3 API: ${response.status}`);
const envelope = await response.json();
console.log(envelope.data.overall_severity);
```

PHP (serverseitig)

```
$options = ['http' => ['header' =>
  'Authorization: Bearer '.getenv('MBCT3_API_KEY')."\\r\\n"
]];
$json = file_get_contents(
  'https://lage.mbct3.org/api/v1/lage/latest', false,
  stream_context_create($options)
);
$data = json_decode($json, true, 512, JSON_THROW_ON_ERROR);
```

Empfohlene Integrationsgrenze

Backend -> MBCT3 API: API-Schlüssel, Cache und Fehlerbehandlung. Frontend -> eigenes Backend: nur benötigte, bereits geprüfte Felder. So bleibt der Schlüssel geheim und Sie kontrollieren Aktualisierung, Darstellung und Ausfallverhalten.

CORS

Für ausdrücklich freigegebene Origins setzt die API Access-Control-Allow-Origin. CORS ersetzt keine Geheimhaltung des Schlüssels. Für öffentliche Webanwendungen bleibt eine serverseitige Vermittlung erforderlich.

Quellen, Nutzung und Produktionscheck

MBCT3 bündelt und verarbeitet Inhalte verschiedener Stellen. Originalquellen, deren Bedingungen und die Grenzen automatisierter Verarbeitung bleiben relevant.

Pflichtangaben bei Weiterverwendung

Angabe	Anforderung
Quelle	Erkennbar: Quelle: MBCT3 Die Lage.
Datenstand	data_updated_at beziehungsweise collected_at sichtbar oder unmittelbar erreichbar.
Nutzungshinweis	Entscheidungsunterstützung, kein amtliches Warnsystem.
Originalquellen	Verknüpfte Quellen und deren Hinweise nicht irreführend entfernen.

Rechte und fachliche Grenzen

Die API räumt keine weitergehenden Rechte an Daten, Texten, Karten, Marken oder Datenbanken Dritter ein. Es gelten die Bedingungen der jeweiligen Originalangebote. MBCT3 sichert Verfügbarkeit, Aktualität, Vollständigkeit und Eignung für einen bestimmten Zweck nicht zu. Bei Gefahr sind amtliche Warnkanäle und behördliche Anweisungen maßgeblich.

Checkliste vor Produktivbetrieb

- API-Schlüssel liegt ausschließlich in einer Secret-Verwaltung oder Serverumgebung.
- ETag beziehungsweise ein sinnvolles eigenes Cache-Intervall ist umgesetzt.
- HTTP 401, 404, 429 und 503 werden kontrolliert behandelt.
- stale und unavailable werden sichtbar und ohne erfundene Ersatzwerte dargestellt.
- Datenstand, Quellenangabe und Warnsystem-Hinweis sind in der Oberfläche vorhanden.
- Die Integration protokolliert request_id, aber niemals Schlüssel oder vollständige Autorisierungsheader.
- OpenAPI-Änderungen und neue Dienstversionen werden regelmäßig geprüft.

Kontakt und Referenzen

API-Dokumentation: <https://lage.mbct3.org/api-dokumentation.php>

OpenAPI: <https://lage.mbct3.org/api/v1/openapi.json>

Quellen: <https://lage.mbct3.org/quellen.php>

Rechtliches: <https://lage.mbct3.org/rechtliches.php>

Kontakt: martin@mbct3.org