

# API der Deutschen Digitalen Bibliothek

## Einführung

Das Application Programming Interface (API) ist eine Programmierschnittstelle, die den Zugriff auf Daten und Methoden der Deutschen Digitalen Bibliothek (DDB) ermöglicht. Sie erlaubt die Entwicklung vielfältiger Anwendungen, die die in der DDB vorgehaltenen Inhalte nutzen und sie jeweils nach den eigenen Wünschen darstellen und in unterschiedliche Kontexte einbetten. Das API steht allen Personen offen.

Für die Nutzung des Application Programming Interfaces der Deutschen Digitalen Bibliothek ist eine Authentifikation in Form eines Schlüssels (API Key) notwendig. Ein Authentifikationsschlüssel ist eine dem Nutzer eindeutig zugeordnete Zeichenfolge, die bei jeder Anfrage an das API zum Zwecke der Authentifikation mit zu übertragen ist. Authentifikationsschlüssel sind vertraulich und ausschließlich für die Verwendung durch den Nutzer bestimmt. Sie dürfen nicht an Dritte weitergegeben werden, es sei denn mit ausdrücklicher schriftlicher Zustimmung der DDB.

## Registrierung

Jeder registrierte Benutzer der Deutschen Digitalen Bibliothek kann einen Authentifikationsschlüssel für die Verwendung des APIs beantragen. Dies erfolgt über das [Profil](#) im Bereich „Meine DDB“. Den [Nutzungsbedingungen](#) muss jeder, an einem API Key interessierte Nutzer zustimmen.

## Schritte zum API Key

1. Zunächst ist es notwendig sich ein ordentliches [Konto in der Deutschen Digitalen Bibliothek](#) anzulegen. Die Anmeldung über einen OpenID-Service (Google oder Yahoo) ist nicht ausreichend.
2. Wenn Sie sich in der Deutschen Digitalen Bibliothek anmelden, können Sie nun im Bereich „Meine DDB“ und „Konto“ im rechten Menü „API-Zugang“ auswählen.
3. Sobald Sie den [Nutzungsbestimmungen](#) zugestimmt haben, wird Ihnen ihr [persönlicher API Key](#) angezeigt.

## Lizenz

Über das API der Deutschen Digitalen Bibliothek werden ausschließlich **CC0-lizenzierte Metadaten** ausgegeben. Somit unterscheidet sich der über das API ausgegebene Datenbestand von dem im Portal der Deutschen Digitalen Bibliothek, da nicht alle darin enthaltenen Metadaten unter CC0 gestellt wurden.

## Feedback

Ein Feedback zum API bzw. zur Benutzung des API ist sehr willkommen und kann an [feedback@deutsche-digitale-bibliothek.de](mailto:feedback@deutsche-digitale-bibliothek.de) gerichtet werden. Das öffentliche API steht in einer **Beta-Version** zur Verfügung. Zu einem späteren Zeitpunkt ist die Veröffentlichung einer weiterentwickelten Version vorgesehen, in die auch Rückmeldungen aus dem Nutzerkreis einfließen sollen. Dasselbe gilt für die Dokumentation des API.

## Benutzung des API

Das API der Deutschen Digitalen Bibliothek bietet ausschließlich lesenden Zugriff. Über die folgenden **15 Methoden** können die unterschiedlichen Daten und -bestandteile abgerufen werden. Die Dokumentation der API-Methoden wurde, abweichend von der übrigen Dokumentation und wie allgemein in Entwickler-Gemeinschaften üblich, in englischer Sprache verfasst.

- **binary** — The method binary returns the content of a binary file of an item for a given item-ID. This method provides response data as application/octet-stream. A binary file at DDB can be a picture, a thumbnail of a picture, a video clip, an audio file etc. It is a read-only service and must be accessed with a HTTP-GET-request.
- **entities** — The method entities is providing access to the the Lucene search index of entities used at DDB. This method provides response data only as application/json. It is a read-only service and must be accessed with a HTTP-GET-request.
- **institutions** — The method institutions returns a list of institutions which are registered at the DDB. This method provides response data only as application/json. It is a read-only service and must be accessed with a HTTP-GET-request.
  - **sectors** — The method sectors returns the list of available institution sectors at the DDB. Each sector contains a name in the property value and the number of institutions that belong to this sector (count). As institutions can belong to multiple sectors, the overall sum of the different counts can be higher than the total number of institutions. This method provides response data only as application/json. It is a read-only service and must be accessed with a HTTP-GET-request.
- **items**
  - **aip** — The method aip returns the Archive Information Package (AIP) of an item for a given item-ID. An AIP contains all available information of an item including a persistent identifier. This method provides response data

- as application/json and application/xml. It is a read-only service and must be accessed with a HTTP-GET-request.
- **binaries** — The method binaries returns a list of binary data files related to an item for a given item-ID. Binary data can be accessed with the binary method. The binaries method provides response data as application/json and application/xml. It is a read-only service and must be accessed with a HTTP-GET-request.
  - **children** — The method children returns metadata of these items which are related to the inquired item and which are one level deeper (child item) in the hierarchy than the inquired item. The child items will be sorted according to the position field of the hierarchy nodes. If the position is the same the label will be used for sorting. The provided metadata can be empty if the item does not have any child items. This method provides response data as application/json and application/xml. It is a read-only s
  - **edm** — The method edm returns the Europeana Data Model (DDB profile) of an item for a given item-ID. This method provides response data as application/json and application/xml. It is a read-only service and must be accessed with a HTTP-GET-request.
  - **indexing-profile** — The method indexing-profile returns the profile of an item for a given item-ID, which was used for the indexing process. This method provides response data as application/json and application/xml. It is a read-only service and must be accessed with a HTTP-GET-request.
  - **parents** — The method parents returns item-IDs of these items which are related to the inquired item and which are one or more levels higher (parent, grandparent ... item) in the hierarchy than the inquired item. This means that all parent items up to the root item will be provided. The provided metadata can be empty if the item does not have any parent items. This method provides response data as application/json and application/xml. It is a read-only service and must be accessed with a HTTP-GET-request.
  - **source** — The method source returns the initial XML metadata of an item for a given item-ID. The format can be one of the accepted input data formats (e.g. MARCXML, METS/MODS, LIDO, Dublin Core). This method provides response data only as application/xml because the DDB only accept XML-based metadata from its contributing institutions. It is a read-only service and must be accessed with a HTTP-GET-request.
  - **view** — The method view returns the view of an item for a given item-ID. A view is the data set a frontend page at DDB is based on. This method provides response data as application/json and application/xml. It is a read-only service and must be accessed with a HTTP-GET-request.
  - **search** — The method search is providing an interface to the search engine of the DDB. This method provides response data only as application/json. It is a read-only service and must be accessed with a HTTP-GET-request. dd
    - **facets** — The method facets of search returns all available facets or all available facets of a specific type. This method provides response data only as application/json. It is a read-only service and must be accessed with a HTTP-GET-request.
    - **organization** — The method search/organization is providing access to the the Lucene search index of organizations used at DDB. This method provides response data only as application/json. It is a read-only service and must be accessed with a HTTP-GET-request.
    - **sortcriteria** — The method sortcriteria of search returns the available sort criteria and the default criterion of the search result sets. This method provides response data only as application/json. It is a read-only service and must be accessed with a HTTP-GET-request.
    - **suggest** — The method suggest of search returns suggestions for a given query. This method provides response data only as application/json. It is a read-only service and must be accessed with a HTTP-GET-request.

## Datenmodell

Das [Datenmodell](#) der Deutschen Digitalen Bibliothek ist im entsprechenden Abschnitt des Wikis dokumentiert.

## Vokabular

Das verwendete [Vokabular](#) wird im entsprechenden Abschnitt des Datenmodells dokumentiert. Zur Zeit verwendet die Deutsche Digitale Bibliothek das folgende spezifische Vokabular in ihrem Datenmodellen.

- [Hierarchietyp](#)
- [Label](#)
- [Medientyp](#)
- [Sparte](#)
- [Zeitvokabular](#)