

# DÉDALO PUBLICATION API INTRODUCTION

El modelo de publicación más utilizado en Dédaló es la publicación sobre tablas en una base de datos MySQL.

Una vez generado el modelo de publicación (ontología de publicación) ya sea utilizando un modelo pre-existente como el de Historia Oral o creando uno a medida, puede acceder a los datos a través de consultas GET/POST al API, usando parámetros en formato:

[http://mydomain/dedalo/lib/dedalo/publication/server\\_api/v1/json/tables\\_info?code=XXXX](http://mydomain/dedalo/lib/dedalo/publication/server_api/v1/json/tables_info?code=XXXX)

Puede acceder al entorno de pruebas en la url:

[http://mydomain/dedalo/lib/dedalo/publication/server\\_api/v1/docu/ui/?code=XXXX](http://mydomain/dedalo/lib/dedalo/publication/server_api/v1/docu/ui/?code=XXXX)

También pulsando el link desde el panel de la herramienta de administración:

[http://mydomain/dedalo/lib/dedalo/main/?m=tool\\_administration&t=dd242&menu=1](http://mydomain/dedalo/lib/dedalo/main/?m=tool_administration&t=dd242&menu=1)

The screenshot shows the administration interface for Dédaló. At the top, there's a configuration section with a gear icon, a text input field containing 'tld2 like: ts', a 'modelo' dropdown, and an 'Update' button. Below this, there are three settings: 'API DOC -->' (circled in red), 'DOWNLOAD JSON DATA FROM GEONAMES API', and 'Navigate records'. The main part of the screenshot is the 'Dédaló Web REST API 1.0.9' documentation page. It includes a 'Schemes' dropdown set to 'HTTP', a list of API endpoints categorized by function (Info, Records, Thesaurus, Free, Video view, Global, Utils), and a 'comb1' endpoint for combined calls.

En el panel podrá verificar la correcta configuración del API y generar peticiones de prueba sobre sus datos.

Todas las respuestas serán en formato JSON.

(!) Para solucionar problemas de configuración, revise el apartado de configuración del API de publicación: cómo asignar un código de seguridad, etc.

Explicamos a continuación el modelo para Historia Oral, que viene habilitado por defecto en todas las instalaciones.

### INFO

GET

/tables\_info

Info about tables schema

La petición GET 'tables\_info' retorna información de todas las tablas existentes en nuestra base de datos de publicación.

Para Historia Oral están estandarizadas estas tablas:

'interview': contiene información genérica de la entrevista, así como los vínculos a los recursos utilizados en la misma (imagen, audiovisual, informante)

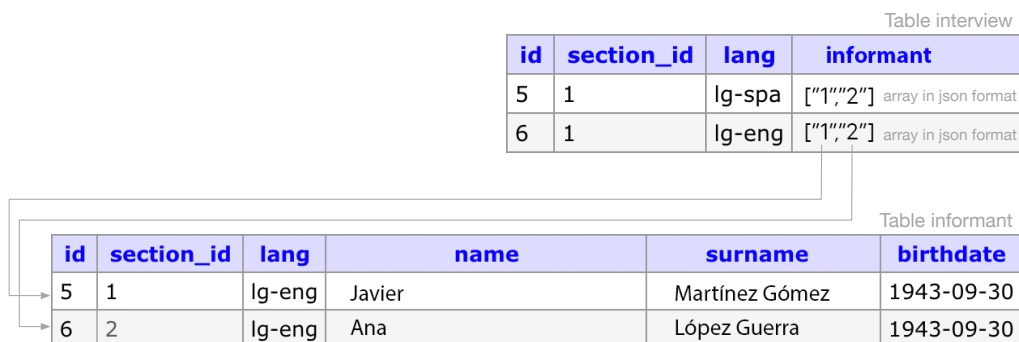
'image': contiene los datos de las imágenes asociadas a la entrevista (url, título, pie de foto, descripción)

'audiovisual': contiene los datos de los audiovisuales (grabaciones) a nivel de 'cinta', con la url del fichero de video, el texto de transcripción, la duración, subtítulos, etc.

'informant': contiene los datos de los informantes asociados a las entrevistas (nombre, apellidos, año y lugar de nacimiento, etc.)

La tabla *interview* contiene columnas de relación a las tablas *image*, *audiovisual*, *informant*. El dato que albergan es un array stringnificado en formato JSON como '["1","2"]'. Este dato se corresponde con la columna *section\_id* de cada tabla de destino.

En este ejemplo vemos la correspondencia entre la columna *informant* (tabla *interview*) y la columna *section\_id* (tabla *informant*).



Esta correspondencia puede ser resuelta (si la necesitamos) individualmente, a través de peticiones simples a cada tabla, o en una petición conjunta utilizando la opción *resolve\_portals* o *resolve\_portals\_custom* en la misma.

Ejemplo de respuesta JSON para la petición *tables\_info*:

```

{
  "image": [
    "section_id",
    "lang",
    "publication",
    "image",
    "title",
    "footprint",
    "description"
  ],
  "informant": [
    "section_id",
    "lang",
    "publication",
    "name",
    "surname",
    "nickname",
    "birthdate",
    "birthplace",
    "birthplace_id",
    "gender",
    "location",
    "location_id",
    "profession",
    "dead_date",
    "dead_place",
    "dead_id",
    "biography",
    "observations"
  ],
  "interview": [
    "section_id",
    "lang",
    "publication",
    "code",
    "title",
    "abstract",
    "priority",
    "interview_place",
    "primary_lang",
    "date",
    "image",
    "audiovisual",
    "informant",
    "images",
    "project"
  ],
  "ts_web": [
    "section_id",
    "lang",
    "term_id",
    "term",
    "web_path",
    "template_name",
    "titulo",
    "entrada",
    "cuerpo"
  ],
  "publication_schema": [
    "data"
  ]
}

```

**GET** /publication\_schema Info about automatic portal resolution map

La petición GET 'publication\_schema' retorna información de la configuración de la resolución automática de portales (*resolve\_portals*), que recoge los datos de la tabla *publication\_schema*. Se define en las propiedades del elemento 'diffusion\_element' en la ontología de difusión y se actualiza al salvar el mismo.

Como se comentó, será usada como mapa para la resolución automática de referencias (portales). El nombre de cada propiedad corresponde a la columna que alberga el puntero y el valor a la tabla de destino donde se aloja la información.

1 [lg-eng] Dédaló documentation [dd1099] (no traducible) P M diffusion\_element ▼

```
{
  "diffusion": {
    "class_name": "diffusion_mysql"
  },
  "publication_schema": {
    "image": "image",
    "audiovisual": "audiovisual",
    "informant": "informant",
    "images": "image"
  }
}
```

Ejemplo de respuesta JSON para la petición *publication\_schema*:

```
{
  "image": "image",
  "audiovisual": "audiovisual",
  "informant": "informant",
  "images": "image"
}
```

<- Ejemplo de ontología configurada con datos de 'publication\_schema' en el item *diffusion\_element*

## RECORDS

**POST** /records Get records from DDBB

La petición POST 'records' es una búsqueda genérica SQL que retorna el listado de registros encontrados como un array de objetos.

Se puede usar para recuperar registros en cualquier tabla, posibilitando seleccionar las columnas a retornar, el número de registros, el idioma de los mismos, la agrupación, los filtros sql, etc.

Ejemplo:

[http://mydomain/lib/dedalo/publication/server\\_api/v1/json/records?code=XXX&table=image&lang=lg-eng&limit=10](http://mydomain/lib/dedalo/publication/server_api/v1/json/records?code=XXX&table=image&lang=lg-eng&limit=10)

El parámetro booleano *resolve\_portals* habilita la resolución de referencias de una tabla a otra como se define en *publication\_schema*

resolve\_portal Activates automatic resolution of portals option. Default is false  
boolean true

El parámetro *resolve\_portals\_custom* posibilita definir sobre la marcha el mapa de resolución de portales siguiendo el mismo formato que *publication\_schema*

resolve\_portals\_custom Resolve requested portals only. Use json map object and value is target table. To resolve deeper, use forr  
string resolve\_portals\_custom - Resolve requested portals only. Use j

Ejemplo de respuesta JSON para la petición *records*:

```
{
  "result": [
    {
      "table": "image",
      "section_id": "5824",
      "lang": "lg-eng",
      "publication": "yes",
      "image": "/dedalo/media/image/1.5MB/5000/rsc29_rsc170_5824.jpg",
      "title": "",
      "footprint": "",
      "description": ""
    },
    {
      "table": "image",
      "section_id": "5831",
      "lang": "lg-eng",
      "publication": "yes",
      "image": "/dedalo/media/image/1.5MB/5000/rsc29_rsc170_5831.jpg",
      "title": "",
      "footprint": "",
      "description": ""
    }
  ]
}
```

Puede ver el listado completo de opciones en la ficha de información técnica.

## THESAURUS

Para trabajar con el tesaurus, existe una serie de llamadas específicas que facilitan operaciones y consultas que serían muy complejas de hacer usando solamente la petición *records*.

El *thesaurus* puede ser una sola tabla o un grupo de tablas definido en la configuración del API server, o en cada petición, on the fly.

**POST** /reel\_terms Get reel indexed terms

La petición POST 'reel\_terms' retorna la resolución de todos los términos utilizados en la indexación de una transcripción. Esto es útil, por ejemplo para saber qué términos son referidos a lo largo de una entrevista.

Se entiende como 'reel' cada uno de los registros de la tabla 'audiovisual'. Una entrevista (tabla 'interview') puede referenciar a varios 'reels'.

La información de la transcripción siempre está contenida en la columna de nombre 'rsc36' llamada así porque es el tipo del componente real que la aloja en Dédalo.

Ejemplo de respuesta JSON para la petición *reel\_terms*:

```
{
  "result": [
    {
      "term_id": "aa1_30",
      "term": "Child labor",
      "locators": [
        {
          "type": "dd96",
          "tag_id": "2",
          "section_id": "1",
          "section_tipo": "rsc167",
          "component_tipo": "rsc36",
          "section_top_id": "1",
          "section_top_tipo": "oh1",
          "from_component_tipo": "hierarc40"
        }
      ]
    }
  ]
}
```

**POST** /fragment\_from\_index\_locator Build fragment of text and video from a index locator

La petición POST 'fragment\_from\_index\_locator' retorna el fragmento indexado con el localizador de indexación dado.

Ejemplo de localizador (locator):

```
{"section_tipo":"rsc167","section_id":"2","tag_id":"1"}
```

En determinados contextos, podemos conocer el locator de la indexación (por ejemplo con el resultado de la petición 'reel\_terms')

Si conocemos el locator de la indexación (como en el supuesto anterior) podemos extraer el fragmento de la entrevista que se indexó con el.

El 'fragmento' retornado contendrá la selección del texto de la transcripción indexada, además de la url del fragmento de video y de los subtítulos así como la información de los tiempos de inicio y fin del fragmento (tc in / tc out )

Ejemplo de respuesta JSON para la petición *fragment\_from\_index\_locator*:

```
{
  "result": [
    {
      "fragm": " Or I am married from the 64,
that married me with my boyfriend of
all ..",
      "video_url": "/dedalo/media/av/404/
rsc35_rsc167_2.mp4?
vbegin=13&vend=50",
      "subtitles_url": "/dedalo/lib/dedalo/
publication/server_api/v1/subtitles/?
section_id=2&lang=lg-
eng&tc_in=13,526&tc_out=49,763",
      "tcin_secs": 13.526,
      "tcout_secs": 49.763
    }
  ]
}
```

**POST** /thesaurus\_root\_list Get first level of terms from thesaurus

La petición POST 'thesaurus\_root\_list' retorna el listado completo de términos raíz de la jerarquía del tesoro, con información propia y de sus hijos en la misma.

Puede ser útil para generar el punto de partida de un tesoro o búsqueda temática.

Optional comma separated names of thesaurus tables. If undefined, server config thesaurus tables are used

table - Optional comma separated names of thesaurus tables.

Se puede configurar la petición para una o varias tablas, separándolas por ',' en el parámetro 'table' según hayamos construido el tesoro general con una o varias jerarquías temáticas.

Optional comma separated parents used as roots like 'hierarchy1\_245,hierarchy1\_253'. If defined, override default beha

parents - Optional comma separated parents used as roots like

En ocasiones es útil definir puntos de partida o padres concretos, en lugar de los valores reales. Esto puede hacerse con el parámetro 'parents', añadiendo los términos deseados separados por ',' como: 'hierarchy1\_245,hierarchy1\_253'.

**POST** /thesaurus\_random\_term A random thesaurus term name

La petición POST 'thesaurus\_random\_term' retorna un término aleatorio del tesoro. Puede definir la tabla/s de tesoro origen del término y excluir términos no deseados.

Puede ser útil, por ejemplo para generar un término random de referencia en una búsqueda temática y orientar así al visitante.

Ejemplo de respuesta JSON para la petición thesaurus\_root\_list:

```
{
  "result": {
    "hp1": [
      {
        "term_id": "hp1_3",
        "term": "Second Republic",
        "scope_note": "",
        "indexation": "[{\type:\dd96",
        "tag_id": "39", "section_id": "7",
        "section_tipo": "rsc167",
        "component_tipo": "rsc36",
        "section_top_id": "5",
        "section_top_tipo": "oh1",
        "from_component_tipo": "hierarchy40"}],
        "time": "",
        "space": "{\alt:16,\lat": "39.462571",\lon": "-0.376295",
        "zoom": "15",
        "lang": "lg-eng",
        "options": {},
        "highlight": null,
        "table": "periodos",
        "parent_term": null,
        "section_id": "3",
        "tld": "hp1",
        "model": null,
        "parent": "\hp1_1",
        "childrens": null,
        "norder": "0",
        "ar_childrens": []
      }
    ],
    {
      "term_id": "hp1_25",
      "term": "Civil War",
      "scope_note": "",
      "indexation": "[{\type:\dd96",
      "tag_id": "55", "section_id": "2",
      "section_tipo": "rsc167",
      "component_tipo": "rsc36",
      "section_top_id": "2",
      "section_top_tipo": "oh1",
      "from_component_tipo": "hierarchy40"}],
      "time": "",
      "space": "{\alt:16,\lat": "39.462571",\lon": "-0.376295",
      "zoom": "12",
      "lang": "lg-eng",
      "options": {},
      "highlight": null,
      "table": "periodos",
      "parent_term": null,
      "section_id": "25",
      "tld": "hp1",
      "model": null,
      "parent": "\hp1_1",
      "childrens": null,
      "norder": "1",
      "ar_childrens": []
    }
  ]
}
```

Ejemplo de respuesta JSON para la petición thesaurus\_random\_term:

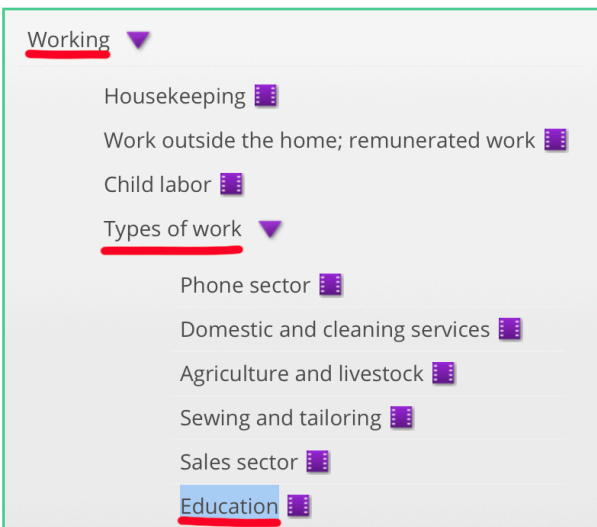
```
{
  "term": "Playing",
  "term_id": "aa1_90",
  "indexation": "[{\type:\dd96,\tag_id": "15",\section_id": "1",\section_tipo": "rsc167",\component_tipo": "rsc36",\section_top_id": "1",\section_top_tipo": "oh1",\from_component_tipo": "hierarchy40"}]
}
```

**POST** /thesaurus\_search Search into thesaurus term

La petición POST 'thesaurus\_search' ejecuta un búsqueda parametrizada sobre todas las tablas que conforman el tesoro (habitualmente nombradas con el prefijo 'ts\_') y retorna un listado de items (objetos) con los datos de los registros encontrados en el tesoro, incluyendo datos de indexaciones, y padre e hijos en la jerarquía.

Retorna, además, la información completa de los padres recursivos para así poder construir el path jerárquico hasta llegar desde el término raíz al término de resultado, como en el ejemplo:

[http://mydomain/dedalo/lib/dedalo/publication/server\\_api/v1/json/thesaurus\\_search?code=XXX&lang=lg-eng&q=Education](http://mydomain/dedalo/lib/dedalo/publication/server_api/v1/json/thesaurus_search?code=XXX&lang=lg-eng&q=Education)



Parameters	
Name	Description
<b>code</b> * required string (query)	Auth code (mandatory)
db_name string (query)	Database name. If not defined, the default database will be used
lang string (query)	Query lang filter. Use Dédaló lang format like lg-eng for english. If undefined, API server settings default lang is used
table string (query)	Optional comma separated names of thesaurus tables. If undefined, server config thesaurus tables are used
exclude_tld string (query)	Optional comma separated list of tld to exclude like 'xx,rt'
<b>q</b> * required string (query)	string to search in database. Minimum 3 chars for full text search
rows_per_page integer (query)	Pagination preference. One row per page is the default value
page_number integer (query)	Pagination preference. Current page number. One is default
tree_root string (query)	Defines first element from build the hierarchy tree when show the results. Accepted values first_parent   last_parent . Default is last_parent

Ejemplo de respuesta JSON para la petición thesaurus\_search:

```
{
  "result": [
    {
      "table": "periodos,antropologia",
      "section_id": "121",
      "lang": "lg-eng",
      "publication": "yes",
      "descriptor": "yes",
      "tld": "aa1",
      "term_id": "aa1_121",
      "term": "Sexual education",
      "model": null,
      "parent": "[aa1_120]",
      "childrens": null,
      "indexation": "[type:dd96]",
      "tag_id": "41", "section_id": "2",
      "section_tipo": "rsc167",
      "component_tipo": "rsc36",
      "from_component_tipo": "hierarchy40",
      "related": "",
      "time": null,
      "code": "",
      "space": "",
      "norder": "0",
      "scope_note": ""
    }
  ],
  "total": 2,
  "page_number": 1,
  "rows_per_page": 1
},
"ar_ts_terms": {
  "aa1": [
    {
      "term_id": "aa1_120",
      "term": "Desiring",
      "scope_note": "",
      "indexation": "",
      "time": "",
      "space": "",
      "lang": "lg-eng",
      "options": {},
      "highlight": null,
      "table": "periodos,antropologia",
      "section_id": "120",
      "publication": "yes",
      "descriptor": "yes",
      "tld": "aa1",
      "model": null,
      "parent": "[aa1_1]",
      "childrens": "[type:dd48]",
      "section_id": "121", "section_tipo": "aa1", "from_component_tipo": "hierarchy49",
      "related": "",
      "code": "",
      "norder": "4",
      "ar_childrens": [
        "aa1_121",
        "aa1_123",
        "aa1_125",
        "aa1_127"
      ]
    }
  ]
},
"ar_highlight": [
  "aa1_121"
],
"ar_parent": [
  "aa1_1",
  "aa1_120"
]
}
```

POST

/thesaurus\_autocomplete Predictive autocomplete search

La petición POST 'thesaurus\_autocomplete' ejecuta una búsqueda de texto que contenga el string pasado sobre el tesauo.

El comando SQL es de tipo 'contiene': `term` LIKE '%edu%'

Es útil para formularios con elementos de búsqueda predictiva.

Por defecto está limitado a 25 resultados y se puede seleccionar el formato del resultado:

simple : array de strings con los términos hallados

full: array de objetos con el término y el identificador (term\_id) de cada término hallado.

POST

/thesaurus\_term Get thesaurus term objects (one or various)

La petición POST 'thesaurus\_term' devuelve la información completa de una lista de términos del tesauo filtrados por sus correspondientes 'term\_id'.

Name	Description
<b>code</b> * required string (query)	Auth code (mandatory)
db_name string (query)	Database name. If not defined, the default database will be used
lang string (query)	Query lang filter. Use Dédaló lang format like lg-eng for english. If undefined, API server settings default lang is used
<b>ar_term_id</b> * required string (query)	Comma separated list of one or various term_id like 'ts1_1,ts1_2'
combine string (query)	Experimental (!). Optional combined result mode. Options: false   combined   cumulative. Default is false. When is set to combine, only indexations returned

Ejemplo de respuesta JSON para la petición `thesaurus_term`:

```
{
  "result": [
    {
      "term_id": "aa1_1",
      "term": "Social Antropology",
      "scope_note": "",
      "indexation": "",
      "time": "",
      "space": "",
      "lang": "lg-eng",
      "options": {},
      "highlight": null,
      "table": "antropologia",
      "section_id": "1",
      "publication": "yes",
      "descriptor": "yes",
      "tld": "aa1",
      "model": null,
      "parent": "[\"hierarchy1_246\"]",
      "childrens": "[[\"type\": \"dd48\", \"section_id\": \"6\", \"section_tipo\": \"aa1\", \"from_component_tipo\": \"hierarchy49\"]]",
      "related": "",
      "code": "",
      "norder": "0",
      "ar_childrens": [
        "aa1_6"
      ]
    }
  ],
  "matching_terms": false
}
```



**POST** /thesaurus\_indexation\_node Indexation object with all required info

La petición POST 'thesaurus\_indexation\_node' devuelve la información completa de un 'nodo de indexación'.

Un 'nodo de indexación' es un objeto que contiene la información de todas las indexaciones (etiquetas) del termino dado agrupadas por entrevista.

Esto es útil, por ejemplo, si dentro de una misma entrevista hay 3 referencias (indexaciones) a un término. Obtenemos entonces los datos de todas ellas, lo que nos permite navegar sobre las diferentes etiquetas, dentro de la misma entrevista.

Además obtenemos la url del posterframe de la entrevista lo que facilita la gestión de las miniaturas.

Ejemplo de uso:



Ejemplo de respuesta JSON para la petición thesaurus\_indexation\_node:

```
{
  "result": [
    {
      "term_id": "ts1_1",
      "locator": {
        "type": "dd96",
        "tag_id": "1",
        "section_id": "1",
        "section_tipo": "rsc167",
        "component_tipo": "rsc36",
        "section_top_id": "1",
        "section_top_tipo": "oh1",
        "from_component_tipo": "hierarchy40"
      },
      "node_id": "1",
      "image_type": "posterframe",
      "lang": "lg-eng",
      "image_url": "/dedalo/media/av/posterframe/rsc35_rsc167_1.jpg",
      "group_locators": [
        {
          "type": "dd96",
          "tag_id": "1",
          "section_id": "1",
          "section_tipo": "rsc167",
          "component_tipo": "rsc36",
          "section_top_id": "1",
          "section_top_tipo": "oh1",
          "from_component_tipo": "hierarchy40"
        }
      ]
    },
    {
      "term_id": "ts1_1",
      "locator": {
        "type": "dd96",
        "tag_id": "1",
        "section_id": "4",
        "section_tipo": "rsc167",
        "component_tipo": "rsc36",
        "section_top_id": "3",
        "section_top_tipo": "oh1",
        "from_component_tipo": "hierarchy40"
      },
      "node_id": "3",
      "image_type": "posterframe",
      "lang": "lg-eng",
      "image_url": "/dedalo/media/av/posterframe/rsc35_rsc167_4.jpg",
      "group_locators": [
        {
          "type": "dd96",
          "tag_id": "1",
          "section_id": "4",
          "section_tipo": "rsc167",
          "component_tipo": "rsc36",
          "section_top_id": "3",
          "section_top_tipo": "oh1",
          "from_component_tipo": "hierarchy40"
        }
      ]
    }
  ]
}
```



POST

/thesaurus\_video\_view\_data Video player data

**Obsoleto (!)**. La petición POST 'thesaurus\_video\_view\_data' devuelve la información necesaria para mostrar un fragmento de video a partir de un id de término (term\_id) y un array de locators .

Parameters	
Name	Description
<b>code</b> * required string (query)	Auth code (mandatory)
db_name string (query)	Database name. If not defined, the default database will be used
lang string (query)	Query lang filter. Use Dédaló lang format like lg-eng for english. If undefined, API server settings default lang is used
<b>term_id</b> * required string (query)	Term id to get data like ts1_1
<b>ar_locators</b> * required string (query)	JSON encoded array of locator objects like [{"type":"dd96","tag_id":"2","section_id":"1","section_tipo":"rsc167","component_tipo":"rsc36","section_top_id":"1","section_top_tipo":"oh1","from_component_tipo":{"type":"dd96","tag_id":"6","section_id":"1","section_tipo":"rsc167","component_tipo":"rsc36","section_top_id":"1","section_top_tipo":"oh1","from_component_tipo":
ar_locators_key integer (query)	current array key selector. Default is '0'

Ejemplo de respuesta JSON para la petición *thesaurus\_video\_view\_data*:

```
{
  "lang": "lg-eng",
  "add_subtitles": false,
  "result": true,
  "interview_id": "1",
  "av_section_id": "1",
  "image_url": "/dedalo/mediat/av/posterframe/rsc35_rsc167_1.jpg",
  "terms": [],
  "informant": [
    {
      "table": "informant",
      "section_id": "1",
      "lang": "lg-eng",
      "name": "Pilar [Madre]",
      "surname": "Librero Colmenero",
      "birthdate": "1943-09-30",
      "birthplace": "Huelva, Huelva, Andalucía, Reino de España",
      "birthplace_id": "[\"es1_3410\"]",
      "gender": "Mujer",
      "location": "Valencina de la Concepción, Sevilla, Andalucía, Reino de España",
      "location_id": "[\"es1_7248\"]",
      "profession": "Ama de casa",
      "observations": "Pilar es"
    },
    {
      "table": "informant",
      "section_id": "2",
      "lang": "lg-eng",
      "name": "Celeste [Hija]",
      "surname": "Delgado-Librero",
      "birthdate": "1966-02-27",
      "birthplace": "Sevilla, Sevilla, Andalucía, Reino de España",
      "birthplace_id": "[\"es1_6519\"]",
      "gender": "Mujer",
      "location": "",
      "location_id": null,
      "profession": "Profesora",
      "observations": "Vive entre EEspaña. nació en.."
    }
  ],
  "summary": "The mother was born in the year 1943.",
  "interview_date": "2012-02-15",
  "interview_code": "PLC-1",
  "total": 2,
  "key": 0,
  "ar_key": [
    0,
    1
  ],
  "term_id": "ts1_1",
  "term": null,
  "fragments": [
    null
  ],
  "ar_restricted_fragments": [
    {
      "fragment": "TODAY IT IS THREE.",
      "video_url": "?vbegin=1&vend=38",
      "subtitles_url": "/dedalo/lib/dedalo/publication/server_api/v1/subtitles/?section_id=1&lang=lg-eng&tc_in=1&tc_out=38",
      "tcin_secs": 1,
      "tcout_secs": 38
    }
  ]
}
```

**POST**

`/thesaurus_childrens` Search all childrens of current term

La petición POST 'thesaurus\_childrens' devuelve el listado de registros con los hijos jerárquicos del término pasado encontrados en el tesoro.

Por defecto, devuelve lo hijos directos, pero se pueden solicitar también todos los hijos recursivamente, filtrar por los que son de tipo 'descriptor' y eliminar los definidos en configuración como restringidos del resultado.

Parameters	
Name	Description
<b>code</b> <small>* required</small> string <i>(query)</i>	Auth code (mandatory)
db_name string <i>(query)</i>	Database name. If not defined, the default database will be used
lang string <i>(query)</i>	Query lang filter. Use Dédaló lang format like lg-eng for english. If undefined, API server settings default lang is used
ar_fields string <i>(query)</i>	Comma separated list of required columns in table
term_id string <i>(query)</i>	Term id to get data like ts1_1
recursive boolean <i>(query)</i>	set deep hierarchy search on true
only_descriptors boolean <i>(query)</i>	Exclude no descriptors in the result
remove_restricted boolean <i>(query)</i>	Exclude restricted terms in the result

Ejemplo de respuesta JSON para la petición `thesaurus_childrens`:

```
{
  "result": [
    {
      "table": "antropologia",
      "section_id": "3",
      "lang": "lg-eng",
      "descriptor": "yes",
      "tld": "aa1",
      "term_id": "aa1_3",
      "term": "Learning",
      "model": null,
      "parent": "\\aa1_1\\",
      "childrens": "[\\type\\:dd48\\",
      "\\section_id\\:139\\,\\section_tipo\\:aa1\\,\\from_component_tipo\\:\\hierarchy49\\,\\type\\:dd48\\",
      "\\section_id\\:140\\,\\section_tipo\\:aa1\\,\\from_component_tipo\\:\\hierarchy49\\]",
      "indexation": null,
      "related": "",
      "space": "\\lat\\:39.462571\\,\\lon\\:-0.376295\\,\\zoom\\:12\\,\\alt\\:16}",
      "norder": "1",
      "scope_note": ""
    },
    {
      "table": "antropologia",
      "section_id": "4",
      "lang": "lg-eng",
      "descriptor": "yes",
      "tld": "aa1",
      "term_id": "aa1_4",
      "term": "Working",
      "model": null,
      "parent": "\\aa1_1\\",
      "childrens": "[\\type\\:dd48\\",
      "\\section_id\\:29\\,\\section_tipo\\:aa1\\,\\from_component_tipo\\:\\hierarchy49\\,\\type\\:dd48\\",
      "\\section_id\\:31\\,\\section_tipo\\:aa1\\,\\from_component_tipo\\:\\hierarchy49\\]",
      "indexation": null,
      "related": "",
      "space": "\\alt\\:16\\,\\lat\\:39.462571\\,\\lon\\:-0.376295\\,\\zoom\\:12}",
      "norder": "2",
      "scope_note": ""
    },
    {
      "table": "antropologia",
      "section_id": "5",
      "lang": "lg-eng",
      "descriptor": "yes",
      "tld": "aa1",
      "term_id": "aa1_5",
      "term": "Believing",
      "model": null,
      "parent": "\\aa1_1\\",
      "childrens": "[\\type\\:dd48\\",
      "\\section_id\\:99\\,\\section_tipo\\:aa1\\,\\from_component_tipo\\:\\hierarchy49\\,\\type\\:dd48\\",
      "\\section_id\\:72\\,\\section_tipo\\:aa1\\,\\from_component_tipo\\:\\hierarchy49\\]",
      "indexation": null,
      "related": "",
      "space": "\\lat\\:39.462571\\,\\lon\\:-0.376295\\,\\zoom\\:12\\,\\alt\\:16}",
      "norder": "8",
      "scope_note": ""
    }
  ],
  "total": 10
}
```

**POST** /thesaurus\_parents Search all parents of current term

La petición POST 'thesaurus\_parents' devuelve el listado de registros con los padres jerárquicos del término pasado encontrados en el tesauero.

Por defecto, devuelve lo padres directos, pero se pueden solicitar también todos los padres recursivamente.

Parameters	
Name	Description
code <small>required</small> string <i>(query)</i>	Auth code (mandatory)
db_name string <i>(query)</i>	Database name. If not defined, the default database will be used
lang string <i>(query)</i>	Query lang filter. Use Dédaló lang format like lg-eng for english. If undefined, API server settings default lang is used
ar_fields string <i>(query)</i>	Comma separated list of required columns in table
term_id string <i>(query)</i>	Term id to get data like ts1_2
recursive boolean <i>(query)</i>	Set deep hierarchy search single or multi level

Ejemplo de respuesta JSON para la petición *thesaurus\_parents*:

```
{
  "result": [
    {
      "table": "antropologia",
      "section_id": "12",
      "lang": "lg-eng",
      "descriptor": "yes",
      "tld": "aa1",
      "term_id": "aa1_12",
      "term": "Being",
      "model": null,
      "parent": "\\aa1_1\\",
      "childrens": "[\\type\\:\"dd48\",
        \\section_id\\:\"192\\,\\section_tipo\\:
        \\aa1\\,\\from_component_tipo\\:
        \\hierarchy49\\,\\type\\:\"dd48\",
        \\section_id\\:\"160\\,\\section_tipo\\:
        \\aa1\\,\\from_component_tipo\\:
        \\hierarchy49\\,\\type\\:\"dd48\",
        \\section_id\\:\"95\\,\\section_tipo\\:
        \\aa1\\,\\from_component_tipo\\:
        \\hierarchy49\\]",
      "indexation": null,
      "related": "",
      "space": "\\lat\\:\"39.462571\\,\\lon\\:
        \\-0.376295\\,\\zoom\\:12,\\alt\\:16\",
      "norder": "9",
      "scope_note": ""
    },
    {
      "table": "antropologia",
      "section_id": "1",
      "lang": "lg-eng",
      "descriptor": "yes",
      "tld": "aa1",
      "term_id": "aa1_1",
      "term": "Social Antropology",
      "model": null,
      "parent": "\\hierarchy1_246\\",
      "childrens": "[\\type\\:\"dd48\",
        \\section_id\\:\"6\\,\\section_tipo\\:\"aa1\"
        \\from_component_tipo\\:\"hierarchy49\",
        \\type\\:\"dd48\\,\\section_id\\:\"3\",
        \\section_tipo\\:\"aa1\",
        \\from_component_tipo\\:\"hierarchy49\",
        \\type\\:\"dd48\\,\\section_id\\:\"4\",
        \\section_tipo\\:\"aa1\",
        \\from_component_tipo\\:\"hierarchy49\",
        \\type\\:\"dd48\\,\\section_id\\:\"7\",
        \\section_tipo\\:\"aa1\",
        \\from_component_tipo\\:
        \\hierarchy49\\]",
      "indexation": null,
      "related": "",
      "space": "\\alt\\:16,\\lat\\:
        \\39.462571\\,\\lon\\:\"-0.376295\",
        \\zoom\\:15\",
      "norder": "0",
      "scope_note": ""
    }
  ]
}
```

**FREE**

Para gestionar las búsqueda de texto libre contra el texto de las transcripciones de las entrevistas, puede usar las siguientes peticiones.

**POST**`/free_search` Find word/s in DDBB

La petición POST 'free\_search' devuelve un listado de objetos con los datos de los elementos encontrados al ejecutar la búsqueda.

Estos datos incluyen la información de la entrevista, las imágenes, los informantes y las cintas (reels) asociados, además de las url de los posterframes, de los ficheros de video 'cortados' y los subtítulos, así como los fragmentos restringidos relevantes (incluidos en los fragmentos encontrados).

Name	Description
<b>code</b> * required string <i>(query)</i>	Auth code (mandatory)
db_name string <i>(query)</i>	Database name. If not defined, the default database will be used
lang string <i>(query)</i>	Query lang filter. Use Dédaló lang format like lg-eng for english. If undefined, API server settings default lang is used
<b>q</b> * required string <i>(query)</i>	String to search
search_mode string <i>(query)</i>	SQL type of search. Default is 'full_text_search'
rows_per_page integer <i>(query)</i>	Number of records returned by page. Default is 10
page_number integer <i>(query)</i>	Current page number (pagination params)
offset integer <i>(query)</i>	Offset used to paginate results. Default is 0
appearances_limit integer <i>(query)</i>	Max number of appearances (matches) calculated. Default is 1
match_select boolean <i>(query)</i>	Selects specific match inside results. Default = false . Optional
count boolean <i>(query)</i>	Exec total records count. Default is true
image_type string <i>(query)</i>	Type of image returned. Options are posterframe   identify_image. Default is posterframe
list_fragment boolean <i>(query)</i>	Resolve text fragment to show in list mode (small text, without video cut). Default is true
video_fragment boolean <i>(query)</i>	Activates video fragment resolution (with extended transcription fragment and video cut). Default is false
fragment_terms boolean <i>(query)</i>	Calculates terms (indexations) for current fragment of transcription. Default is false
filter string <i>(query)</i>	Apply a filter to result like 'section_tipo = 1'

Puede ver el listado completo de opciones en la ficha de información técnica.

Ejemplo de respuesta JSON para la petición `free_search`:

```
{
  "result": [
    {
      "av_section_id": "14",
      "q": "house",
      "appearances_limit": "1",
      "match_select": false,
      "image_type": "posterframe",
      "video_fragment": false,
      "list_fragment": true,
      "fragment_terms": false,
      "lang": "lg-eng",
      "video": "/dedalo/media/av/404/rsc35_rsc167_14.mp4",
      "interview_section_id": "11",
      "code": "EMJ-11",
      "title": "",
      "abstract": "Crió a su hijo sin ayuda del padre...",
      "priority": "5",
      "interview_place": "Murcia, Reino de España",
      "date": "2001-02-25",
      "image": [
        {
          "table": "image",
          "section_id": "11",
          "lang": "lg-eng",
          "image": "/dedalo/lib/dedalo/themes/default/0.jpg",
          "title": "",
          "footprint": "",
          "description": ""
        }
      ],
      "audiovisual": "[^14^]",
      "informant": [
        {
          "table": "informant",
          "section_id": "22",
          "lang": "lg-eng",
          "name": "María",
          "surname": "Pérez Jiménez",
          "nickname": "",
          "birthdate": "1947-09-16",
          "birthplace": "Ivi., Reino de España",
          "birthplace_id": "[^es1_4515^]",
          "gender": "Mujer",
          "profession": "Agricultura | Servicio doméstico | Ama de casa",
        },
        {
          "table": "informant",
          "section_id": "23",
          "lang": "lg-eng",
          "name": "Luis",
          "surname": "Albero Roma",
          "nickname": "",
          "birthdate": "1947-04-07",
          "birthplace": "Logroño, Reino de España",
          "birthplace_id": "[^es1_4515^]",
          "gender": "Hombre",
          "profession": "Albañil",
        }
      ],
      "project": "[^1^]",
      "image_url": "/dedalo/av/posterframe/rsc35_rsc167_14.jpg",
      "ar_restricted_fragments": [],
      "fragments": [
        {
          "word": "house",
          "match": 1,
          "list_fragment": ".. begin. as... it begins, we go to begin since you were small... how you lived...? as in your house .."
        }
      ]
    }
  ],
  "page_number": 1,
  "rows_per_page": "1",
  "total": 54
}
```

## VIDEO VIEW

Gestiona la recuperación de información de los videos de las entrevistas

**POST** /full\_reel Full transcriptions data

La petición POST 'full\_reel' devuelve la información completa de una cinta dada (reel) y los recursos asociados a la entrevista de la que forma parte.

Proporciona la transcripción completa, sin cortes (columna rs36) y los fragmentos restringidos contenidos en la misma si existen.

Además de los recurso referidos, el dato contiene las url de los ficheros de imagen y video referidos y los subtítulos.

Name	Description
<b>code</b> * required string (query)	Auth code (mandatory)
db_name string (query)	Database name. If not defined, the default database will be used
lang string (query)	Query lang filter. Use Dédaló lang format like lg-eng for english. If undefined, API server settings default lang is used
<b>av_section_id</b> * required integer (query)	Section id number to filter records
image_type string (query)	Type of image returned. Options are posterframe   identify_image. Default is posterframe
terms boolean (query)	Resolves current transcription indexed terms. Default is false

Ejemplo de respuesta JSON para la petición `full_reel`:

```
{
  "result": {
    "av_section_id": "1",
    "lang": "lg-eng",
    "image_type": "posterframe",
    "terms": false,
    "video": "/dedalo/media/av/404/rsc35_rsc167_1.mp4",
    "duration": "86",
    "interview_section_id": "1",
    "title": "",
    "abstract": "Lorem ipsum dolor sit amet...",
    "priority": "5",
    "interview_place": "Cádiz, Cádiz, Andalucía, Reino de España",
    "date": "2002-02-08",
    "image": [
      {
        "table": "image",
        "section_id": "1",
        "lang": "lg-eng",
        "image": "/dedalo/image/1.5MB/0/rsc29_rsc170_1.jpg",
        "title": ""
      }
    ],
    "audiovisual": "[\"1\"]",
    "informant": [
      {
        "table": "informant",
        "section_id": "1",
        "lang": "lg-eng",
        "name": "Alonso",
        "surname": "Pérez Gómez",
        "birthdate": "1943-09-30",
        "birthplace": "Huelva, Huelva, Andalucía, Reino de España",
        "birthplace_id": "[\"es1_3410\"]",
        "gender": "Mujer",
        "location": "Sevilla, Andalucía, Reino de España",
        "location_id": "[\"es1_7248\"]",
        "observations": "Pilar es la madre de Celeste. (A) 26/1/12"
      },
      {
        "table": "informant",
        "section_id": "2",
        "lang": "lg-eng",
        "name": "martín",
        "surname": "Darloger Lope",
        "birthdate": "1966-02-27",
        "birthplace": "Sevilla, Andalucía, Reino de España",
        "birthplace_id": "[\"es1_6519\"]",
        "gender": "Hombre",
        "location": "",
        "location_id": null,
        "observations": "Lorem ipsum dolor sit amet.."
      }
    ],
    "project": "[\"1\"]",
    "image_url": "/dedalo/av/posterframe/rsc35_rsc167_1.jpg",
    "ar_restricted_fragments": [
      {
        "fragm": "Lorem ipsum dolor sit amet ...",
        "video_url": "?vbegin=1&vend=38",
        "subtitles_url": "/dedalo/lib/dedalo/publication/server_api/v1/subtitles/?section_id=1&lang=lg-eng&tc_in=1&tc_out=38",
        "tcin_secs": 1,
        "tcout_secs": 38
      }
    ],
    "fragments": [
      {
        "video_url": "/dedalo/media/av/404/rsc35_rsc167_1.mp4",
        "subtitles_url": "/dedalo/lib/dedalo/publication/server_api/v1/subtitles/?section_id=1&lang=lg-eng",
        "fragm": "Lorem ipsum dolor sit amet, consectetur adipiscing elit"
      }
    ],
    "result": true,
    "ar_key": [
      "1"
    ],
    "key": "1"
  },
  "msg": "Ok. Request full_reel done successfully"
}
```

## UTILS

### Herramientas y útiles especiales

**POST** /combi Combined calls to api in one request

La petición POST 'combi' devuelve el resultado de la ejecución de varias llamadas al API agrupadas en una sola llamada.

Esto es especialmente útil en casos en los que necesitamos hacer muchas peticiones conocidas al API que generan mucho tráfico y tiempos de espera largos asociados a la latencia de las redes. Al agruparlos en una, se acelera el proceso de petición/respuesta y se reduce el tráfico considerablemente.

Las peticiones así encapsuladas pueden ser de cualquier tipo soportado por el API, sin límite de cantidad o repetición.

Ejemplo de array de objetos preparatorio para una consulta combinada con dos peticiones 'records' :

```
[
  {
    "id": "templates_all",
    "options": {
      "dedalo_get": "records",
      "table": "template_map",
      "ar_fields": [
        "data",
        "name"
      ],
      "lang": "lg-spa"
    }
  },
  {
    "id": "ts_web",
    "options": {
      "dedalo_get": "records",
      "table": "ts_web",
      "ar_fields": [
        "section_id",
        "term_id",
        "term",
        "web_path",
        "parent",
        "childrens",
        "template_name",
        "titulo",
        "entradilla",
        "cuerpo",
        "norder",
        "imagen",
        "casos"
      ],
      "lang": "lg-spa"
    }
  }
]
```

Al recuperar el resultado, usaremos el 'id' de cada petición para discriminarlos.