

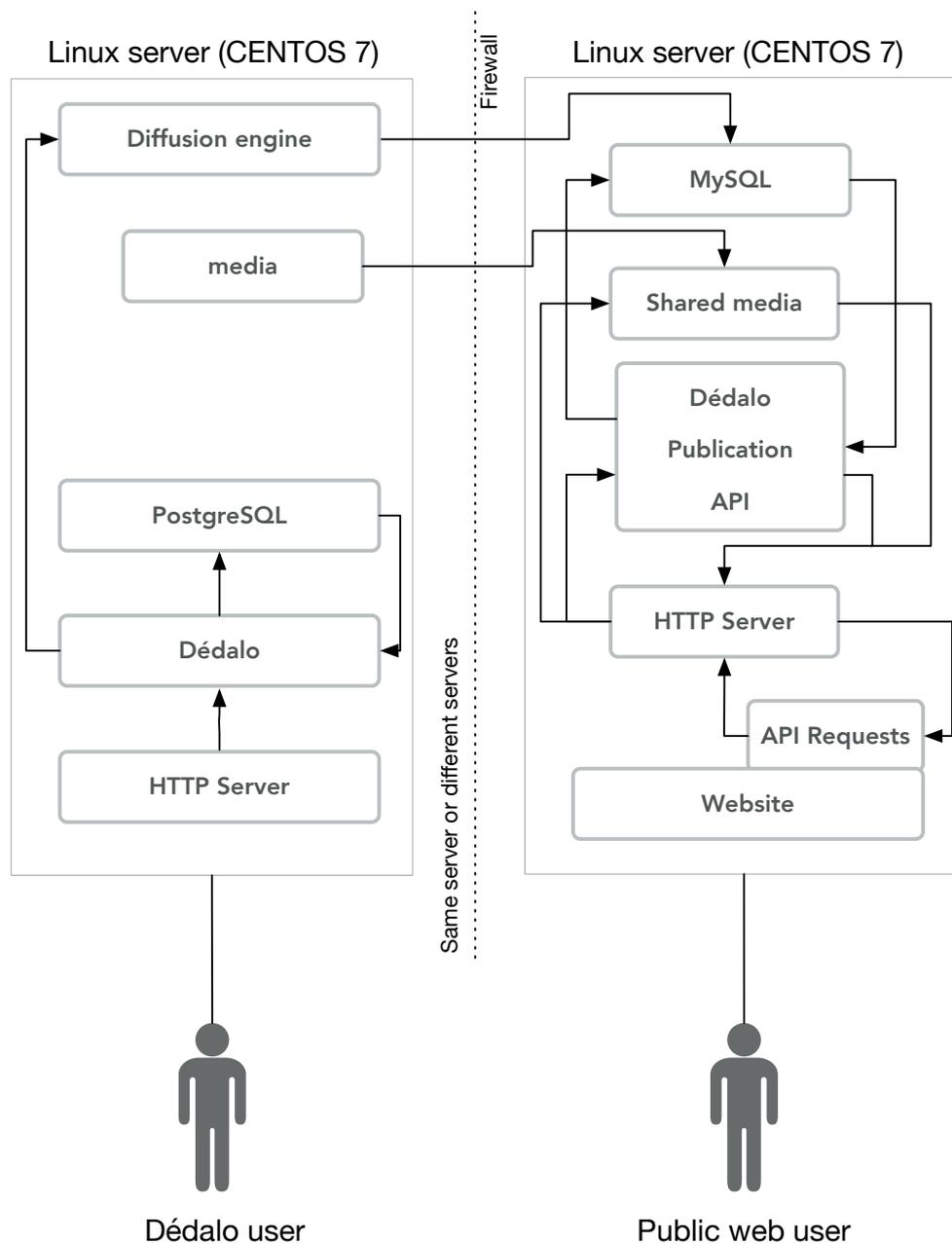
DÉDALO DIFFUSION FLOW

Dédalo utiliza un sistema de 'volcado' externo para la difusión/publicación de contenidos. Esto significa que los datos de trabajo y los datos públicos están totalmente separados, en soportes diferentes y a veces en servidores físicos diferentes.

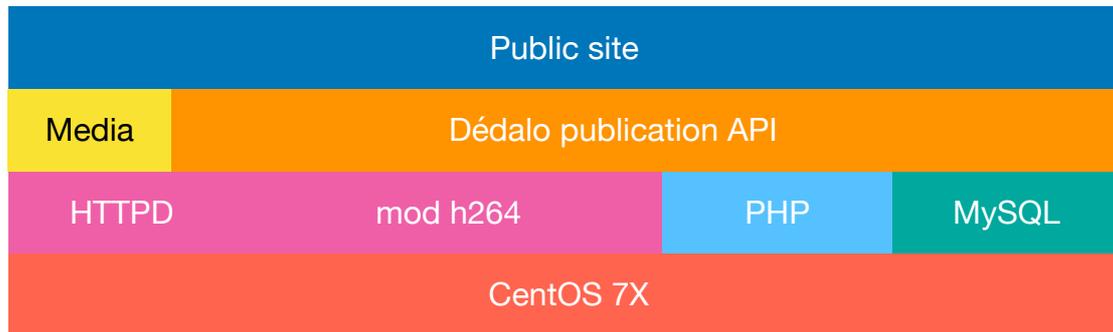
Los investigadores deciden qué datos son públicos y qué datos no lo son, y al estar separados, un ataque al servidor público o un problema técnico en el área de publicación no afectará al sistema de trabajo.

CONFIGURACIÓN

La configuración más habitual es la siguiente:

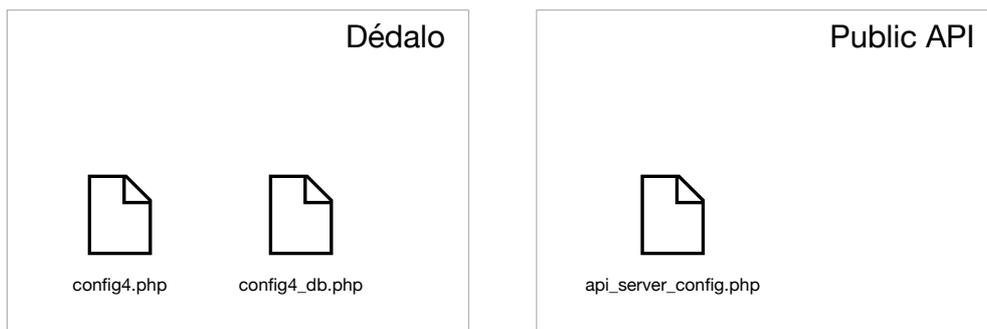


En esta configuración, Dédalo exporta/publica los datos desde el sistema de trabajo (datos JSON en PostgreSQL) a una base de datos externa (datos planos en tablas de MySQL).



Para establecer los parámetros básicos de publicación existen 3 ficheros principales que albergan las variables de configuración:

[config4.php](#) | [config4_db.php](#) | [api_server_config.php](#)



[/dedalo/lib/dedalo/config/config4.php](#)

Aquí debe configurar variables generales de publicación como el dominio de difusión (DEDALO_DIFFUSION_DOMAIN) o los lenguajes de salida (DEDALO_DIFFUSION_LANGS)

[/dedalo/lib/dedalo/config/config4_db.php](#)

Aquí debe configurar los valores relativos a la conexión con la base de datos de publicación: servidor, usuario, password, puerto, etc.

[/dedalo/lib/dedalo/publication/server_api/v1/config_api/api_server_config.php](#)

Aquí debe configurar valores específicos para el API de consulta. Recuerde que puede estar en un servidor separado y que por tanto tendrá que repetir algunos valores ya existentes en los ficheros de configuración de Dédalo, como los de conexión con la base de datos o el código de usuario.

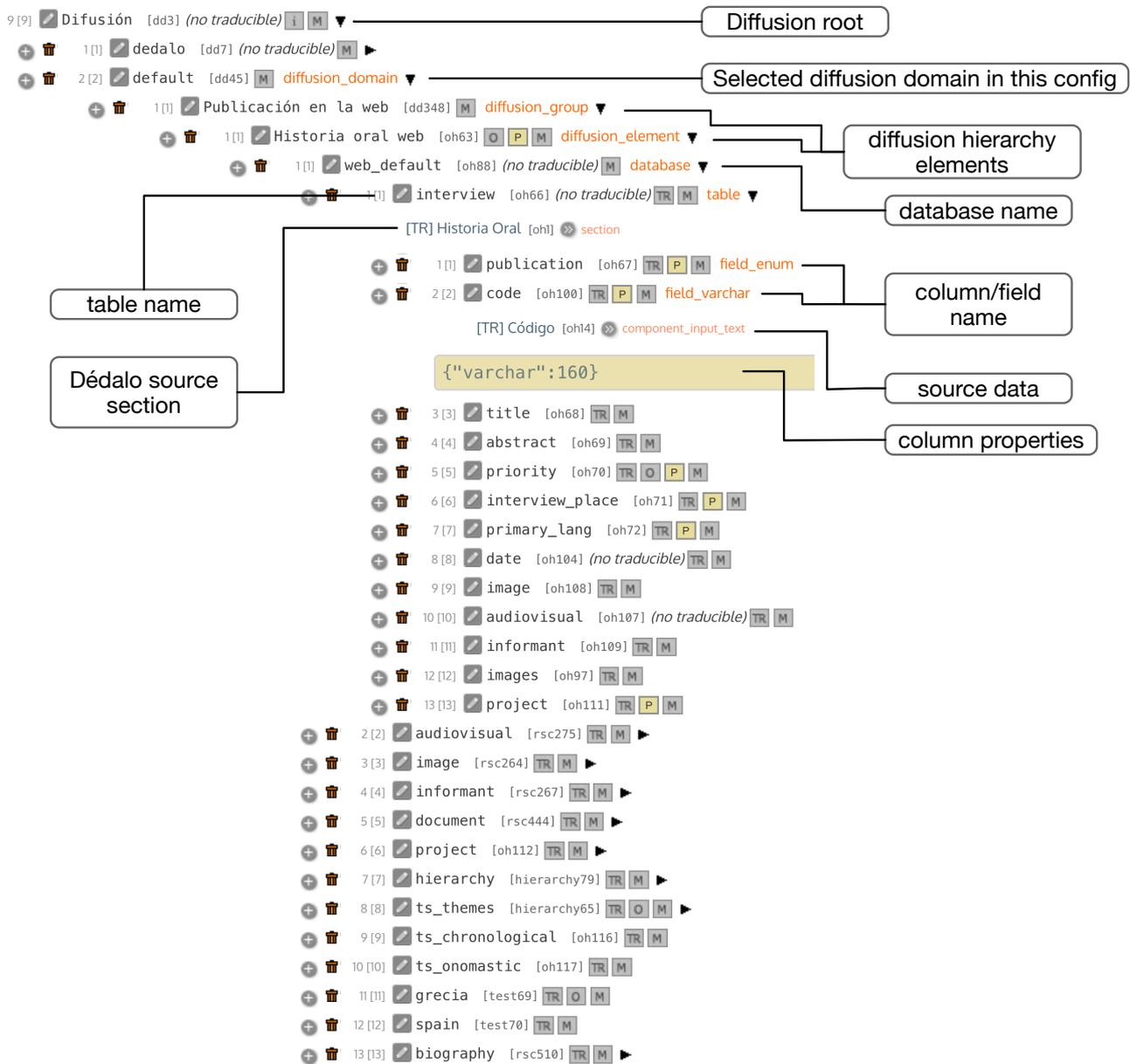
En los directorios implicados encontrará ficheros 'sample.config..' que le ayudarán a configurar los valores en cada caso.

Encontrará información detallada en el documento: DÉDALO PUBLICATION API REFERENCE

ONTOLOGÍA

En la publicación, Dédalo convierte los datos y resuelve las relaciones internas sobre tablas sql estándar siguiendo el mapa definido en la ontología de difusión.

Ejemplo:



Esta ontología de difusión define las características, relaciones y nomenclaturas de las tablas de destino y sus columnas, así como el formato de los datos a publicar. Existen ontologías de difusión pre-configuradas como la de Historia Oral que se pueden extender y modificar según las necesidades.

Para cada elemento de destino existe un objeto o elemento en la ontología (término) que lo representa y sobre el que se pueden configurar parámetros específicos. Este elemento tiene un 'modelo' que define qué es, lo que determina las opciones que puede utilizar y cómo se interpretarán dinámicamente por Dédalo.

Por ejemplo, para representar una tabla estándar y sus columnas en MySQL, definiríamos en la ontología un elemento de modelo 'table' configurado con un término relacionado (TR) que apunta a la sección Historia Oral (oh1) de donde saldrán los datos.

Como hijo de este término, crearíamos un término de modelo 'field_varchar' (note que se corresponde con el tipo de columna estándar MySQL precedido del prefijo 'field_') y en sus propiedades JSON, en este caso, definiríamos el valor de la longitud de caracteres deseada, por ejemplo, {"varchar":160}.

Este elemento apuntaría (término relacionado) al componente origen del dato, 'Código' (oh14). Así para todas las columnas deseadas (lo más cómodo es ver elementos ya definidos y copiar las opciones que nos interesen, cambiando los parámetros específicos).



Para que una ontología de publicación sea operativa, es necesario crear siempre una columna 'field_enum' relacionada con el componente de publicación de la sección. De esta forma tendremos control de si un registro es publicable o no.



PUBLICACIÓN DE DATOS

Cuando el usuario publica datos desde la sección 'Historia Oral' (oh1) con el botón que se habrá activado en esa sección al crear la ontología relacionada, comienza el flujo de publicación.



Dédalo entonces recoge los datos requeridos de la sección (oh1), recorre cada uno de los términos definidos en el árbol de la ontología de publicación de la sección ('code' en este caso) transformando el dato original en valores planos de exportación. Esta conversión implica, por ejemplo, que los datos traducibles en Dédalo, que contienen todos los idiomas simultáneamente, sean desglosados en el momento de la publicación en un valor para cada idioma que generará un registro diferente para cada uno de ellos.

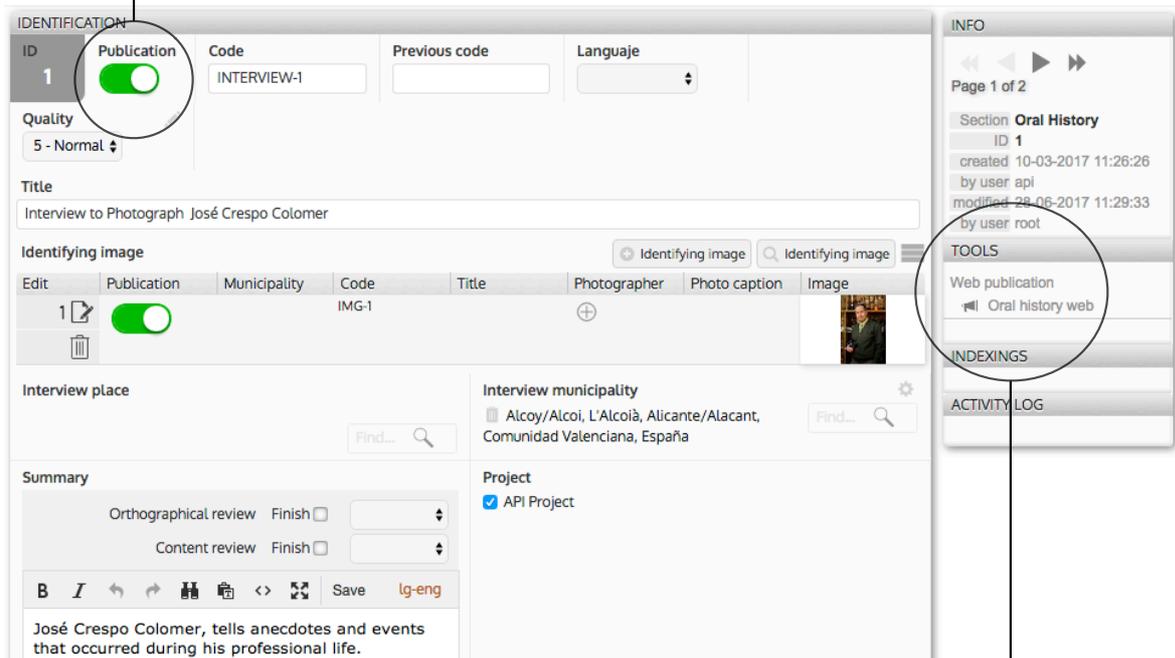
Publication component



Set current section record as public or not. If activated, when user makes a single record publication or multiple section publication, this record is included to publish.

If is not activated, this record will be deleted in the next publication action.

NOTE: Only to set the publication on green NOT publish nothing. Only when user press the special button "Web Publication" the record/s is published.



Publication button



Edit mode: Publish (copy to public database or publication element) only current record.

If publication component is set to off, this record is removed in target publication element (if exists).

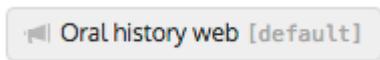
Multiple publication (list mode)

Page 1 of 1. Displayed records from 1 to 2 of 2

New interview View Statistics Print records Export Oral history web

Info...	Code	Title	Info...	Proj...	Lan...	Inte...	Tra...	Ind...	Aud...	Ide...
1 TR IN ST TL 00:52:06.000 Terms	INTERVIEW-1	Interview to Photograph José Crespo Colomer	José Crespo Colomer 25-3-1953	API Project		Alcoy/Alcoi, L'Alcoià, Alicante/Alacant, Comunidad Valenciana, España				
6 TR IN ST TL 00:22:46.000 Terms	API-TEST-2	Interview to Photograph Francisco Pérez Lucina	Francisco Pérez Lucina 1943	API Project	Catalan	Alcoy/Alcoi, L'Alcoià, Alicante/Alacant, Comunidad Valenciana, España				

Publication button

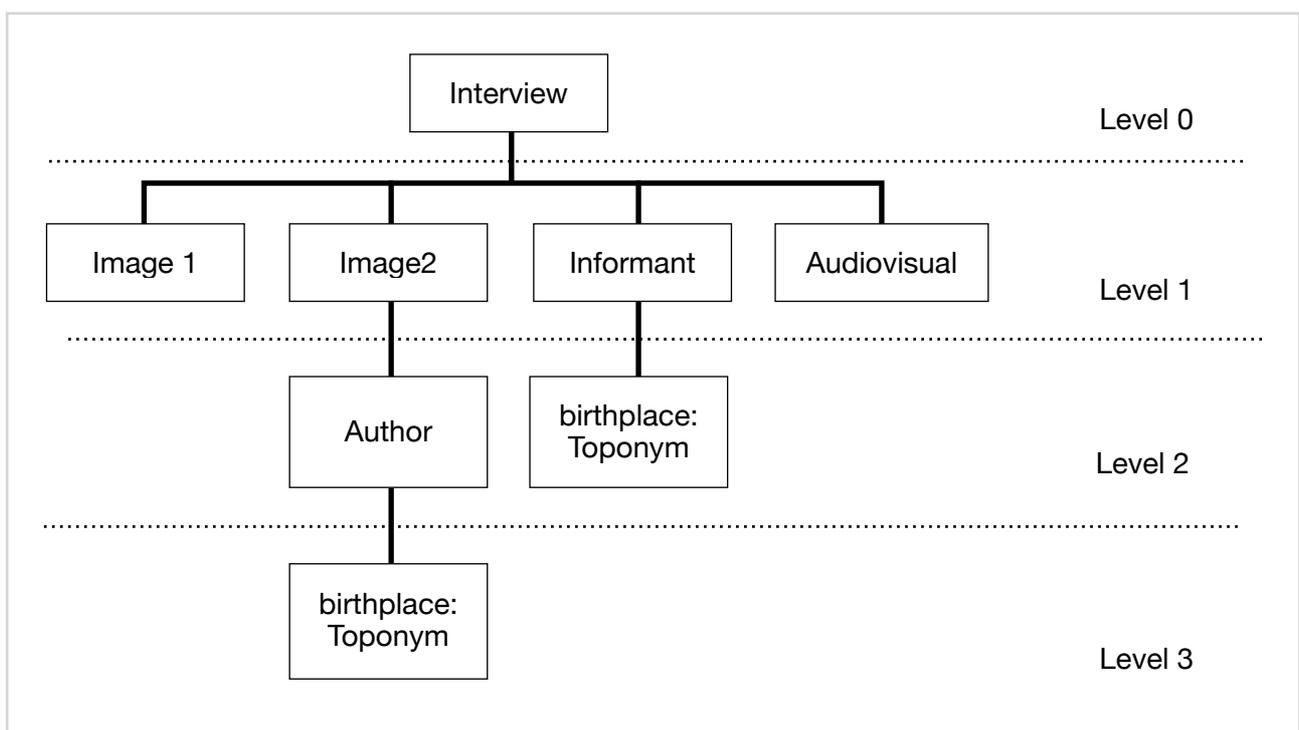


List mode: Publish (copy to public database or publication element) all viewed records (you can make a search to reduce the number of records to publish) respecting publication component value, in the same way as individual but massively

Además, si existen datos relacionados (por ejemplo para portales y componentes autocomplete) se resolverán los vínculos recorriendo el número de niveles definido en la configuración de Dédalo.

Ejemplo de resolución de relaciones en profundidad:

1 interview -> 1 image -> 1 person -> 1 toponym



Al publicar, se verifica si la tabla de destino existe. Si no es así, se crea automáticamente. Las columnas 'section_id' y 'lang' siempre se crean de forma automática en este proceso ya que son obligatorias.

Ejemplo de creación automática de tabla MySQL:

Comando sql generado:

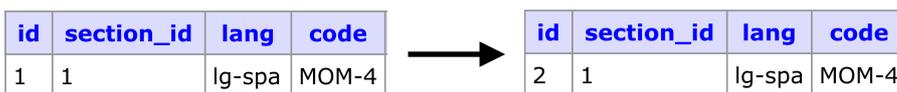
```
CREATE TABLE `interview` (
  `section_id` int NOT NULL,
  `lang` varchar(8) NOT NULL,
  `code` varchar(160) NOT NULL
) ENGINE='MyISAM';
```

Tabla generada:

section_id	lang	code
1	lg-spa	MOM-4

Si existen errores al definir las tablas, puede borrarlas y re-crearlas en cualquier momento.

Cuando el usuario publica un registro ya existente, se borrará el anterior y se creará uno nuevo con los datos actualizados.



CONVERSIÓN DE DATOS

Como hemos comentado, al publicar los datos se lleva a cabo una conversión del formato para adecuarlos a una consulta sencilla desde el API.

Cada idioma creará una registro con el código del idioma actual que se asignará a la columna fija 'lang'.

Por tanto habrá 3 registros físicos (rows) para un registro de Dédalo si publicamos en 3 idiomas (en este ejemplo Castellano, Inglés y Catalán para el registro 1).

Si no existe un dato en algún idioma se asignará el valor del idioma principal definido en el fichero config4.php (DEDALO_DATA_LANG_DEFAULT)

Si el component de origen no es traducible, su valor se repetirá en toda las filas.

section_id	lang	code	title
1	lg-spa	MOM-4	Entrevista etnológica
1	lg-eng	MOM-4	Ethnological interview
1	lg-cat	MOM-4	Entrevista etnològica

Para datos con referencias (locators) como los de portales, se generarán referencias en formato JSON del tipo ["1","2"] hacia la tabla de destino definida en el API

section_id	lang	code	title	informant_data
1	lg-spa	MOM-4	Entrevista etnológica	["1","2"]
1	lg-eng	MOM-4	Ethnological interview	["1","2"]
1	lg-cat	MOM-4	Entrevista etnològica	["1","2"]

Ejemplo de conversión de datos de components con referencias:

Data conversion on publication

DEDALO (PostgreSQL)

```

componen portal data
[
  {
    "type": "dd151",
    "section_id": "1",
    "section_tipo": "dd64",
    "from_component_tipo": "oh32"
  },
  {
    "type": "dd151",
    "section_id": "2",
    "section_tipo": "dd64",
    "from_component_tipo": "oh32"
  }
]

```

Web publication
 📄 Oral history web

Data is converted when the user publish data from Dédalo

Example of component_portal data publication

PUBLICATION API (MySQL)

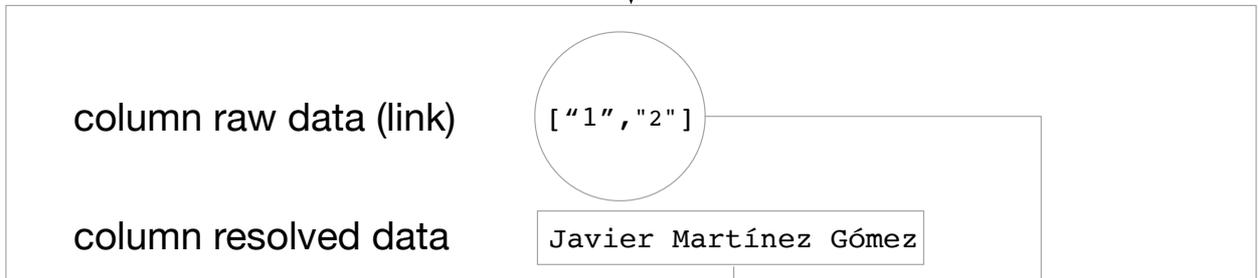


Table interview

id	section_id	lang	informant	informant_data
5	1	Ig-spa	Javier Martínez Gómez, Ana López Guerra	["1","2"] array in json format
6	1	Ig-eng	Javier Martínez Gómez, Ana López Guerra	["1","2"] array in json format

Table informant

id	section_id	lang	name	surname	birthdate
5	1	Ig-eng	Javier	Martínez Gómez	1943-09-30
6	2	Ig-eng	Ana	López Guerra	1943-09-30