

|  |
| --- |
| **Departamento: Tecnología** |
| **Responsables: Desarrollo Digital** |
| **Central Salamanca; España** |
| **Fecha: 23/08/2022** |

**Manual de Actuación**

**– Directus 9 –**

**Acortador de URLs**

**ÍNDICE**

[1. – INTRODUCCIÓN 2](#_Toc112229202)

[2. – OBJETIVO 2](#_Toc112229203)

[3. – REQUISITOS 2](#_Toc112229204)

[4. – PRIMEROS PASOS 2](#_Toc112229205)

[4.1 – Instalación de Directus 2](#_Toc112229206)

[4.2 – Instalación de Extensiones 3](#_Toc112229207)

[5. – CREACIÓN DEL MODELO DE DATOS 4](#_Toc112229208)

[6. – CASOS DE USO 5](#_Toc112229209)

[6.1 – [CU-01] - Solicitud API Rest 5](#_Toc112229210)

[6.2 – [CU-02] - Creación de la URL acortada 6](#_Toc112229211)

[6.3 – [CU-03] - Creación del custom hook 6](#_Toc112229212)

[6.4 – [CU-04] - Redireccionamiento 7](#_Toc112229213)

[7. – VÍDEO FUNCIONAMIENTO 7](#_Toc112229214)

# – INTRODUCCIÓN

Directus es un CMS que nos permite crear contenidos y una estructura y base de datos para luego exponer toda esa arquitectura mediante su API y así poder conectarlo con otras tecnologías y aplicaciones. Al ser un Headless CMS implica que el front de tu web es independiente a las bases de datos y el back. Esto te permite ser más ágil en los cambios del front y además es mucho más eficaz para sincronizar en tiempo real distintas bases de datos.

# – OBJETIVO

Se solicita la creación de un acortador de URLs. Este manual te guiará paso a paso en la realización del proyecto.

# – REQUISITOS

* Se usará Directus como gestor de contenidos para la creación de la base de datos y de la API de conexión. La API que se usará será API Rest.
* La solicitud de acortar la URL se realizará mediante Postman.
* Al realizar la petición:
  + Se almacenará la URL original
  + Se creará y almacenará la nueva URL acortada

El formato de la nueva URL será: string: **gex**- + string: **7 caracteres alfanuméricos** (*ej: gex-tve8d9d*)

* + Se hará la redirección de la URL corta a la URL original

# – PRIMEROS PASOS

## – Instalación de Directus

* Crear el directorio del proyecto

mkdir <nombre del proyecto>

cd <nombre del proyecto>

* Instalar Directus 9

npm init directus-project <nombre del directorio de la api>

Durante el proceso de instalación hay que seleccionar la base de datos e introducir los datos de administración de la herramienta.

* Lanzar Directus 9

cd <ruta del directorio de la api>

npx directus start

Servidor lanzado en <http://localhost:8055>

## – Instalación de Extensiones

Directus es modular, lo que permite la extensión con JavaScript personalizado. Usaremos la lógica de eventos para extender la API y así poder crear nuestra URL acortada y almacenarla en la base de datos.

* Crear la extensión

npm init directus-extension

Se solicitará el tipo de extensión que desearemos crear. Para nuestro proyecto usaremos una extensión de tipo hook.

? Choose the extensión type **hook**

? Choose a name for the extensión **url-shortener**

? Choose the language to use **javascript**

Una vez finalizada la instalación se nos habrá creado una carpeta llamada **url-shortener** en nuestro directorio raíz como proyecto independiente. Esta carpeta viene con algunos paquetes preinstalados según el tipo de extensión y el idioma que elija. El más importante es @directus/extensions-sdk. Este paquete incluye una CLI, que nos permitirá crear nuestra extensión.

* Para empezar a implementar nuestra extensión:

cd url-shortener

npm run dev

* Para llevar nuestra extensión a producción

npm run build

* Una vez creado el fichero en producción tendremos que seguir los siguientes pasos para que Directus ejecute el hook:
  + Copiar archivo *index.js* creado en producción y que se encuentra en la carpeta **url-shortener > dist**
  + En el directorio raíz acceder a la carpeta de hooks: **extensiones > hooks**. Una vez allí crear una carpeta llamada url-shortener y dentro pegar el archivo *index.js*
  + Cada vez que realicemos un cambio en nuestro hook y lo llevamos a producción, es necesario realizar estos pasos. Podemos automatizarlos.

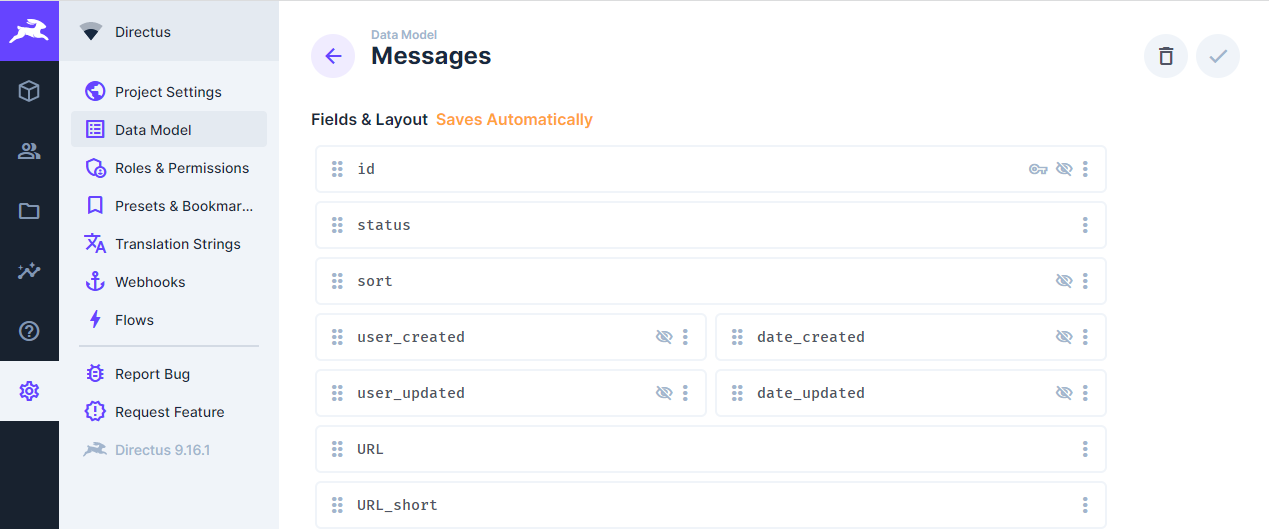
# – CREACIÓN DEL MODELO DE DATOS

A través de la propia interfaz que nos proporciona Directus crearemos una colección que llamaremos **messages.** Es aquí donde se almacenará la información de las URLs que se envíen. Esta colección por lo tanto será una tabla en la base de datos de Directus. Para la creación de la colección seguiremos estos pasos:

* Accedemos a las colecciones: **Settings > Data model**
* Añadimos una nueva colección y la nombramos como **messages**
* Poteriormente marcamos todos los campos que nos aparecen por defecto

Una vez tenemos la tabla creada, añadiremos los campos correspondientes a la misma:

* Dentro de los Data models seleccionamos la tabla que acabamos de crear
* Observamos que además de los campos marcados anteriormente, Directus nos crea por defecto el ID y el estado de la colección.
* En los requisitos no se especifican los campos a crear, por lo tanto, únicamente crearemos los campos necesarios. En nuestro caso crearemos:
  + Un campo llamado **URL**: de tipo Input, único y que no permita valores nulos
  + Un campo llamado **URL\_short**: de tipo Input y que permita valores nulos.



**NOTA**: Las colecciones creadas en Directus, por defecto se marcan únicamente con permisos de administración. Para que podamos leer, actualizar, crear y borrar cualquier campo de la colección que acabamos de crear, es necesario que cambiemos esos permisos para que sean accesibles por todo el mundo.

**Settings > Roles & Permissions > Public > Messages**

Como no se ha especificado nada de permisos en los requisitos, se hará de esta manera para poder realizar las pruebas permitentes. Para la puesta en producción se tendrán que definir los requisitos de permisos para adaptar el desarrollo.

# – CASOS DE USO

1. [CU-01] Creación de un nuevo registro en la tabla **messages** de la base de datos a través de una petición vía API Rest.
2. [CU-02] Creación de un string aleatorio para usarlo como URL corta.
3. [CU-03] Actualización del registro creado anteriormente con el nuevo string en el campo **URL\_short**.
4. [CU-04] Ser hará una solicitud 301 de redirección con la url corta

## – [CU-01] - Solicitud API Rest

La entrada de datos a nuestro proyecto se realizará a través de una solicitud a la API Rest de Directus.

Según el [CU-01], a través de esta petición debemos crear un nuevo registro en la tabla **messages**. Para obtener acceso de escritura a esta tabla haremos uso del método **POST**. La integración se hará vía HTTP.

Para realizar esta solicitud usaremos el programa Postman. Estos son los pasos:

* Indicamos el endpoint:

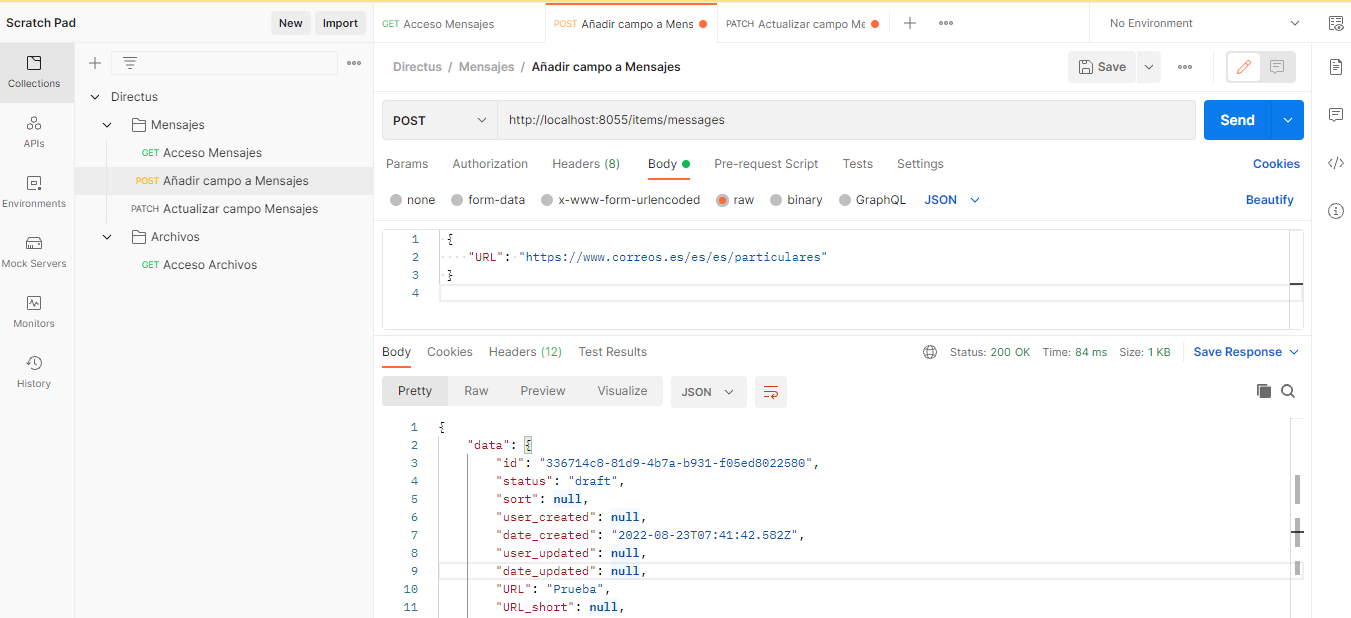
http://localhost:8055/items/messages

* Enviamos el campo a añadir a través del body en formato JSON:

{"URL": <https://www.correos.es/es/es/particulares>}

Tras recibir la solicitud correcta, el backend de Directus creará un nuevo registro en la tabla **messages** con los datos enviados a través del body.

Postman nos devolverá el resultado autor de la llamada.



## – [CU-02] - Creación de la URL acortada

Crearemos un nuevo archivo llamado *shortUrl.js* para generar un número aleatorio de 7 caracteres alfanuméricos, para posteriormente usarlo como URL corta.

Cabe recordar que todo lo que incluyamos en nuestra carpeta src de pruebas tendrá que ir también en la carpeta de hooks > url-shortener del proyecto. Por lo tanto, crearemos ambos ficheros con el mismo nombre y con el siguiente código:

export const shortUrl = () => Math.random().toString(36).substr(2,7)

Lo que hemos hecho ha sido crear un número aleatorio con la función nativa de javascript que nos devolverá un número entre 0 y 1 (*ej.: 0.31313212331*). Posteriormente el número lo convertiremos en un número en base 36 ya que al usar tanto letras como número para su representación, nos devolverá un string alfanumérico. Para finalizar usaremos el método substr para eliminar el 0. de nuestro número y para coger los 7 caracteres posteriores tal como se indica en el [CU-02].

## – [CU-03] - Creación del custom hook

Los Custom Hooks en Directus permiten ejecutar una lógica personalizada cuando ocurre un evento específico dentro de su proyecto. Veremos los diferentes eventos disponibles según la documentación y cuál es el que mejor se adapta a nuestros requisitos.

**Entrypoint de nuestra extensión**

Una vez tenemos ya todo instalado y configurado, es hora de crear la extensión personalizada para poder acortar las URL que se soliciten.

El punto de entrada de la extensión como hemos comentado anteriormente será el *index.js* ubicado dentro de la carpeta hooks de nuestras extensiones, pero la implementación de la función se realizará inicialmente en el index.js ubicado en la carpeta src de nuestro acortador.

El primer paso será exportar una función para registrar el detector de eventos. Esta función recibe un objeto que contiene las funciones de registro específicas del tipo como primer parámetro: **filter, action, init y schedule** y como segundo parámetro es un objeto de contexto con algunas de las siguientes propiedades: **services, getSchema, database, exceptions…**

*Tanto la documentación de cada propiedad como del resto de propiedades se pueden consultar en la documentación oficial de Directus 9.*

En nuestro caso esta función será:

export default ({filter, action}, {services, database, getSchema}) => {}

**Eventos**

Nuestro hook podrá activarse en función del evento disparado. Un evento se define por su tipo y su nombre. Como hemos descrito anteriormente, los tipos de eventos disponibles son: filter, action, init y schedule.

Usaremos **filter** cuando deseemos que el hook se dispare antes del evento y **action** cuando queramos que el hook se dispare después del evento.

Según el [CU-03], una vez se dispare el hook, se almacenará el string en el campo **URL\_short** del registro creado a través de la petición a la API. El flujo será:

**HOOK**

**URL\_short**

**URL**

ACTION

BASE DE DATOS

DIRECTUS

Dentro de los eventos de tipo **action** existe (<collection>.)items.create que se ejecuta cuando se crea un nuevo registro en una colección. En nuestro caso, se ejecutará cuando añadamos un nuevo registro a la tabla messages.

Una vez activado el evento simplemente tendremos que actualizar el campo URL\_short del registro añadido, con la url generada en el [CU-02].

## – [CU-04] - Redireccionamiento

Pendiente de implementación

# – VÍDEO FUNCIONAMIENTO

A continuación, se muestra un pequeño vídeo del proceso de generación de una URL corta en Directus a partir de una solicitud API Rest con Postman.

[Vídeo Acortador URLs en Directus](https://www.loom.com/share/046178926ab54516acd4d8698269db56)