

# Uso de la API almacén:

Para la documentación siguiente supondremos que la api se encuentra albergada en la url:

[http://localhost/DWES/dwes06-API\\_REST/apirestcrud](http://localhost/DWES/dwes06-API_REST/apirestcrud)

Esta URL será referenciada de aquí en adelante como **<base\_url>** en caso de albergar el servidor en otra carpeta se habrá de sustituir por la url apropiada.

## Categorías:

### POST de Categorías:

Se ha de enviar una petición de tipo POST a **<base\_url>/categoria.php**

El body de la petición ha de ser un objeto de tipo JSON con los siguientes campos:

`cat_nombre`: El nombre de la nueva categoría

Ejemplo:

```
{
  "cat_nombre": "Estufas Eléctricas",
}
```

La respuesta del servidor en caso de éxito será un objeto de tipo JSON con los siguientes campos:

```
result: ok,
insert_id: El campo ID del nuevo registro
```

En caso de error nos devolverá

```
result: error,
details: Mensaje de detalle del error
```

## GET de Categorías:

Se ha de enviar una petición de tipo GET a [<base\\_url>/categoria.php](base_url/categoria.php)

El body de la petición, en caso de ser añadido ha de ser un objeto de tipo JSON con alguno de los siguientes campos:( Si se añade un body el resultado vendrá filtrado por los campos del JSON )

```
cat_id: El ID de la categoría
cat_nombre: El nombre de la categoría
```

La respuesta del servidor en caso de éxito será un objeto de tipo JSON con los siguientes campos:

```
result: ok,
categorias: Una lista de las diferentes categorías
```

Cada categoría será un objeto de tipo JSON con los siguientes campos:

```
cat_id: El ID de la categoría
cat_nombre: El nombre de la categoría
```

En caso de error nos devolverá

```
result: error,
details: Mensaje de detalle del error
```

## PUT de Categorías:

Se ha de enviar una petición de tipo PUT a [<base\\_url>/categoria.php](base_url/categoria.php)

El body de la petición ha de ser un objeto de tipo JSON con los siguientes campos:

```
cat_nombre: El nuevo nombre de la categoría
```

Ejemplo:

```
{
  "cat_nombre": "Estufas Eléctricas enchufables",
}
```

También se ha de especificar en la url el query param: `cat_id` y este debe ser el ID de una categoría existente

La respuesta del servidor en caso de éxito será un objeto de tipo JSON con los siguientes campos:

```
result: ok,
```

En caso de error nos devolverá

```
result: error,
details: Mensaje de detalle del error
```

## DELETE de Categorías:

Se ha de enviar una petición de tipo DELETE a <base\_url>/categoria.php

Se ha de especificar en la url el query param: `cat_id` y este debe ser el ID de una categoría existente

La respuesta del servidor en caso de éxito será un objeto de tipo JSON con los siguientes campos:

```
result: ok,
```

En caso de error nos devolverá

```
result: error,
```

```
details: Mensaje de detalle del error
```

# Artículos:

## POST de Artículos:

Se ha de enviar una petición de tipo POST a [<base\\_url>/articulo.php](base_url/articulo.php)

El body de la petición ha de ser un objeto de tipo JSON con los siguientes campos:

art\_nombre: El nombre del nuevo artículo  
art\_categoria: el id de la categoría del artículo  
art\_cantidad: la cantidad de artículos de este tipo en stock

Ejemplo:

```
{
  "art_nombre": "Nintendo Swich",
  "art_categoria": "5",
  "art_cantidad": "10"
}
```

La respuesta del servidor en caso de éxito será un objeto de tipo JSON con los siguientes campos:

result: ok,  
insert\_id: El campo ID del nuevo registro

En caso de error nos devolverá

result: error,  
details: Mensaje de detalle del error

## GET de Artículos:

Se ha de enviar una petición de tipo GET a [<base\\_url>/articulo.php](base_url/articulo.php)

El body de la petición, en caso de ser añadido ha de ser un objeto de tipo JSON con alguno de los siguientes campos:( Si se añade un body el resultado vendrá filtrado por los campos del JSON )

```
art_id: El ID del artículo.  
art_nombre: El nombre del artículo.  
art_categoria: El id de la categoría del artículo.
```

La respuesta del servidor en caso de éxito será un objeto de tipo JSON con los siguientes campos:

```
result: ok,  
articulos: Una lista de los diferentes artículos.
```

Cada categoría será un objeto de tipo JSON con los siguientes campos:

```
art_id: El ID del artículo.  
art_nombre: El nombre del artículo.  
art_categoria: El id de la categoría del artículo.  
art_cantidad: Cantidad de artículos en stock.
```

En caso de error nos devolverá

```
result: error,  
details: Mensaje de detalle del error
```

## PUT de Artículos:

Se ha de enviar una petición de tipo PUT a [<base\\_url>/articulo.php](base_url/articulo.php)

El body de la petición ha de ser un objeto de tipo JSON con los siguientes campos:

```
art_nombre: El nuevo nombre del artículo.  
art_categoria: El id de la categoría del artículo.  
art_cantidad: Cantidad de artículos en stock.
```

Ejemplo:

```
{  
  "art_nombre": "Nintendo Swich azul",  
  "art_categoria": "5",  
  "art_cantidad": "10"  
}
```

También se ha de especificar en la url el query param: `art_id` y este debe ser el ID de un artículo existente.

La respuesta del servidor en caso de éxito será un objeto de tipo JSON con los siguientes campos:

```
result: ok,
```

En caso de error nos devolverá

```
result: error,  
details: Mensaje de detalle del error
```

## DELETE de Artículos:

Se ha de enviar una petición de tipo DELETE a [<base\\_url>/articulo.php](base_url/articulo.php)

Se ha de especificar en la url el query param: `art_id` y este debe ser el ID de un artículo existente.

La respuesta del servidor en caso de éxito será un objeto de tipo JSON con los siguientes campos:

```
result: ok,
```

En caso de error nos devolverá:

```
result: error,  
details: Mensaje de detalle del error
```

# Artículos de una categoría:

## GET de Artículos por categoría:

Se ha de especificar en la url el query param: `categoria` y este debe ser el ID de una categoría existente.

La respuesta del servidor en caso de éxito será un objeto de tipo JSON con los siguientes campos:

```
result: ok,  
artículos: Lista de los diferentes artículos pertenecientes a  
la categoria indicada.
```

Cada categoría será un objeto de tipo JSON con los siguientes campos:

```
art_id: El ID del artículo.  
art_nombre: El nombre del artículo.  
art_categoria: El id de la categoría del artículo.  
art_cantidad: Cantidad de artículos en stock.
```

En caso de error nos devolverá

```
result: error,  
details: Mensaje de detalle del error
```