

# Obtener Comentarios Post (1) - [obtain-comments2]

## ? Descripción

Obtiene los comentarios de una publicación específica del módulo **Publicaciones** (**Frappe/ERPNext**) junto con información adicional del empleado que realizó cada comentario. El servicio:

1. Consulta los comentarios del documento `Publicaciones/{post_name}`.
  2. Obtiene la lista de empleados activos desde ERP para enlazar `owner → nombre_completo`.
  3. Ordena todos los comentarios por fecha de creación en orden ascendente.
  4. Devuelve la lista final enriquecida con el nombre completo del usuario comentarador.
- 

## ? Endpoint

`POST /obtain-comments-by-post`

(La ruta final depende de la configuración en `routes/api.php`)

---

## ? Seguridad

- Requiere **cookie de sesión Frappe** válida (el usuario debe estar logueado).
  - Sin cookie → fallo en la consulta al endpoint de ERP.
- 

## ? Request Body

```
{
  "cookie": "string",
  "post_name": "string",
  "start": 0,
  "limit": 10
}
```

## Campos

Campo	Tipo	Requerido	Descripción
cookie	string	Sí	Cookie de sesión enviada por ERP para realizar consultas autenticadas.
post_name	string	Sí	Nombre (ID) de la publicación en ERP, ejemplo: "PUB-0001".
start	number	No	Índice inicial para paginación de comentarios.
limit	number	No	Cantidad máxima de comentarios a obtener.

## ? Responses

?? 200 – Comentarios obtenidos correctamente

```
{
  "data": [
    {
      "owner": "user@example.com",
      "name": "COM-0001",
      "content": "Buen trabajo equipo!",
      "creation": "2025-01-01 10:30:00",
      "nombre_completo": "Juan Pérez"
    }
  ]
}
```

# ? 400/500 – Error en la solicitud a ERPNext

Si ocurre una excepción desde Guzzle (cookie inválida, publicación inexistente, etc.):

**Respuesta directa del error de ERPNext**, ejemplo:

```
{"exc": ["Traceback...", "..."]}
```

## ? Lógica Interna (Resumen Técnico)

### 1. Validación de entradas

- Se extraen: `cookie`, `post_name`, `start`, `limit`.

### 2. Consulta de comentarios

- **Endpoint Frappe llamado:** `GET APICAPACITACION/resource/Publicaciones/{post_name}`

- **Envía parámetros:**

```
fields: []
filters: []
limit_start: start
limit_page_length: limit
```

### 3. Consulta de empleados activos

- **Solicita:** `Employee?fields=["name","nombre_completo","user_id"]&filters=[[{"status","=","Active"}]]`
- **Construye un mapa:** `user_id → nombre_completo`

### 4. Enriquecimiento de comentarios

- **A cada comentario se le agrega:** `nombre_completo = jsonEmployee[owner] ?? owner`

### 5. Ordenamiento

- Orden ascendente por la fecha `creation`.

### 6. Resultado final

```
{ "data": [ ...comentarios_ordenados... ] }
```

## ? Schema de respuesta final

Comentario:

```
{
  "owner": "string",
  "name": "string",
  "content": "string",
  "creation": "datetime",
  "nombre_completo": "string"
}
```

## ? Ejemplo de uso (curl)

```
curl -X POST https://midominio.com/api/obtain-comments-by-post \
-H "Content-Type: application/json" \
-d '{
  "cookie": "sid=abc123",
  "post name": "PUB-0005",
  "start": 0,
  "limit": 20
}'
```

Revisión #1

Creado 2025-11-24 18:12:13 -05 por Elian

Actualizado 2025-11-24 18:17:39 -05 por Elian