



Informe de Implementación de la API RESTful

DESARROLLO BACKEND - IF201IINF

CONFIGURACIONES Y URLS PARA POSTMAN

- Daniela Droguett
- Nathalie Coyopae
- Patricio Melo
- Carolina Bohle

Informe de Implementación de la API RESTful

Resumen

Este informe detalla la arquitectura de la solución implementada, los métodos disponibles en la API, las URL de prueba y los datos necesarios para probar la API utilizando Postman. La API RESTful ha sido desarrollada para ser consumida por una solución web y permite la gestión de varias entidades como `categoria_servicio`, `info_contacto`, `historia`, `pregunta_frecuente` y `parcela`.

Arquitectura de la Solución

La arquitectura de la solución sigue una estructura básica de carpetas y archivos en PHP, que facilita la gestión y escalabilidad de la API. La estructura de carpetas es la siguiente:

```
Eva2-w-Linces/
├── backend/
│   ├── api/
│   │   └── v1/
│   │       ├── categoria_servicio/
│   │       │   ├── delete.php
│   │       │   ├── get.php
│   │       │   ├── post.php
│   │       │   ├── put.php
│   │       │   └── patch.php
│   │       ├── info_contacto/
│   │       │   ├── get.php
│   │       │   ├── put.php
│   │       │   └── post.php
│   │       ├── historia/
│   │       │   └── get.php
│   │       ├── pregunta_frecuente/
│   │       │   ├── delete.php
│   │       │   ├── get.php
│   │       │   ├── post.php
│   │       │   ├── put.php
│   │       │   └── patch.php
│   │       └── parcela/
│   │           ├── delete.php
│   │           ├── get.php
│   │           ├── post.php
│   │           ├── put.php
│   │           └── patch.php
│   ├── config/
│   │   └── config.php
│   ├── includes/
│   │   ├── auth.php
│   │   ├── controller.php
│   │   ├── db.php
│   │   └── funciones.php
│   └── index.php
```

Detalle de los Métodos y Sus Usos

A continuación se describe cada endpoint, los métodos HTTP disponibles y su uso:

categoria_servicio

URL Base: `http://localhost/Eva2-w-Linces/backend/api/v1/categoria_servicio`

- **GET:** Obtiene todas las categorías de servicio.
 - **URL:** `/get.php`
 - **Authorization:** `Bearer get`
- **POST:** Crea una nueva categoría de servicio.
 - **URL:** `/post.php`
 - **Authorization:** `Bearer post`
 - **Body:**

```
{
  "nombre": "Nombre del servicio",
  "descripcion": "Descripción del servicio"
}
```

- **PUT:** Actualiza una categoría de servicio existente.
 - **URL:** `/put.php`
 - **Authorization:** `Bearer put`
 - **Body:**

```
{
  "id": 1,
  "nombre": "Nombre del servicio actualizado",
  "descripcion": "Descripción del servicio actualizada"
}
```

- **PATCH:** Actualiza parcialmente una categoría de servicio.
 - **URL:** `/patch.php`
 - **Authorization:** `Bearer patch`
 - **Body:**

```
{
  "id": 1,
  "descripcion": "Descripción parcial actualizada"
}
```

- **DELETE:** Elimina una categoría de servicio.
 - **URL:** `/delete.php`
 - **Authorization:** `Bearer delete`
 - **Body:**

```
{
  "id": 1
}
```

info_contacto

URL Base: `http://localhost/Eva2-w-Linces/backend/api/v1/info_contacto`

- ♦ **GET:** Obtiene la información de contacto.

- **URL:** /get.php
- **Authorization:** Bearer get
- **POST:** Crea nueva información de contacto.
 - **URL:** /post.php
 - **Authorization:** Bearer post
 - **Body:**

```
{
  "nombre": "Nombre de contacto",
  "telefono": "Número de teléfono",
  "correo": "Correo electrónico"
}
```

- **PUT:** Actualiza la información de contacto.
 - **URL:** /put.php
 - **Authorization:** Bearer put
 - **Body:**

```
{
  "id": 1,
  "nombre": "Nombre de contacto actualizado",
  "telefono": "Número de teléfono actualizado",
  "correo": "Correo electrónico actualizado"
}
```

historia

URL Base: http://localhost/Eva2-w-Linces/backend/api/v1/historia

- **GET:** Obtiene todas las historias con imágenes.
 - **URL:** /get.php
 - **Authorization:** Bearer get

pregunta_frecuente

URL Base: http://localhost/Eva2-w-Linces/backend/api/v1/pregunta_frecuente

- **GET:** Obtiene todas las preguntas frecuentes.
 - **URL:** /get.php
 - **Authorization:** Bearer get
- **POST:** Crea una nueva pregunta frecuente.
 - **URL:** /post.php
 - **Authorization:** Bearer post
 - **Body:**

```
{
  "pregunta": "Pregunta frecuente",
  "respuesta": "Respuesta a la pregunta frecuente"
}
```

- **PUT:** Actualiza una pregunta frecuente existente.
 - **URL:** /put.php
 - **Authorization:** Bearer put

- **Body:**

```
{
  "id": 1,
  "pregunta": "Pregunta frecuente actualizada",
  "respuesta": "Respuesta actualizada a la pregunta frecuente"
}
```

- **PATCH:** Actualiza parcialmente una pregunta frecuente.

- **URL:** /patch.php
- **Authorization:** Bearer patch
- **Body:**

```
{
  "id": 1,
  "respuesta": "Respuesta parcialmente actualizada"
}
```

- **DELETE:** Elimina una pregunta frecuente.

- **URL:** /delete.php
- **Authorization:** Bearer delete
- **Body:**

```
{
  "id": 1
}
```

parcela

URL Base: http://localhost/Eva2-w-Linces/backend/api/v1/parcela

- **GET:** Obtiene todas las parcelas.

- **URL:** /get.php
- **Authorization:** Bearer get

- **POST:** Crea una nueva parcela.

- **URL:** /post.php
- **Authorization:** Bearer post
- **Body:**

```
{
  "tipo": "Tipo de la parcela",
  "lote": "Lote de la parcela",
  "servicio_id": 1
}
```

- **PUT:** Actualiza una parcela existente.

- **URL:** /put.php
- **Authorization:** Bearer put
- **Body:**

```
{
  "id": 1,
  "tipo": "Tipo de la parcela actualizado",
}
```

```
"lote": "Lote de la parcela actualizado",
"servicio_id": 2
}
```

- **PATCH:** Actualiza parcialmente una parcela.

- **URL:** /patch.php
- **Authorization:** Bearer patch
- **Body:**

```
{
  "id": 1,
  "tipo": "Tipo de la parcela parcialmente actualizado"
}
```

- **DELETE:** Elimina una parcela.

- **URL:** /delete.php
- **Authorization:** Bearer delete
- **Body:**

```
{
  "id": 1
}
```

Datos Necesarios para Probar en Postman

Para probar cada endpoint en Postman, utiliza los siguientes datos de ejemplo:

- **Headers:**
 - **Authorization:** (El token correspondiente al método que estás utilizando, por ejemplo, Bearer get , Bearer post , etc.)
 - **Content-Type:** application/json
- **Body** (para métodos POST, PUT y PATCH):

Ver ejemplos proporcionados en la sección de métodos.

Conclusión

Este informe proporciona una visión general de la arquitectura de la API RESTful implementada, los métodos disponibles y cómo probarlos utilizando Postman. Asegúrate de que los tokens correctos se utilicen para cada solicitud y de que los datos necesarios estén correctamente formateados.

