

Universidad Autónoma Juan Misael Saracho
Facultad de Ciencias Integradas del Gran Chaco
Carrera Ingeniería Informática



DOCUMENTACION API REST FULL

MATERIA: TECNOLOGIA DE PROGRAMACION EN RED

SIGLA: INF423

ESTUDIANTES:

- José Eduardo Reyes Flores
- Cesar Ernesto Mosquez Cabrera

DOCENTE: Elias Cassal Baldiviezo

Fecha de presentación: 22/10/2024

Contenido

Documentación API RESTful - Usuarios, Libros, Géneros	4
Descripción.....	4
Recursos Disponibles	4
Endpoints Usuarios	4
1. Obtener todos los usuarios	4
2. Obtener un usuario por ID	4
3. Crear un nuevo usuario	4
4. Actualizar un usuario existente.....	4
5. Eliminar un usuario	4
Endpoints Libros	4
1. Obtener todos los libros	4
2. Obtener un libro por ID.....	4
3. Crear un nuevo libro	4
4. Actualizar un libro existente	4
5. Eliminar un libro.....	4
Endpoints Géneros	4
1. Obtener todos los géneros.....	4
2. Obtener un género por ID	4
3. Crear un nuevo género	4
4. Actualizar un género existente	4
5. Eliminar un género.....	4
Endpoints Clientes	4
1. Obtener todos los clientes	4
2. Obtener un cliente por ID.....	4
3. Crear un nuevo cliente	4
4. Actualizar un cliente existente	4
5. Eliminar un cliente	4
Endpoints Préstamos	4
1. Obtener todos los préstamos	4
2. Obtener un préstamo por ID.....	4
3. Crear un nuevo préstamo	4
4. Actualizar un préstamo existente.....	4
5. Eliminar un préstamo	4
Endpoints Autores.....	4
1. Obtener todos los autores	4

2. Obtener un autor por ID	5
3. Crear un nuevo autor.....	5
4. Actualizar un autor existente	6
5. Eliminar un autor.....	7

Documentación API RESTful - Usuarios, Libros, Géneros

Descripción

Esta API permite gestionar usuarios, libros y géneros mediante operaciones CRUD (Crear, Leer, Actualizar y Eliminar) en una base de datos. Los datos se almacenan en MySQL, y las operaciones se realizan utilizando PHP.

Base URL: `http://localhost/{nombre de la carpeta}/biblioteca`

Recursos Disponibles

- `usuario`: Recurso que representa a los usuarios.
- `libro`: Recurso que representa los libros.
- `genero`: Recurso que representa los géneros literarios.
- `cliente`: Recurso que representa a los clientes en el sistema.
- `prestamo`: Recurso que representa los préstamos en el sistema.
- `autor`: Recurso que representa a los autores en el sistema.

Endpoints Usuarios

Endpoints Libros

Endpoints Géneros

Endpoints Clientes

Endpoints Préstamos

Endpoints Autores

1. Obtener todos los autores

- **Descripción:** Obtiene una lista de todos los autores registrados en el sistema.
- **Método:** GET
- **Endpoint:** `/autores`
- **Autenticación requerida:** No.

- **Respuesta exitosa:**

```
[
  {
    "id_autor": 1,
    "nombre": "Gabriel",
    "apellido": "García Márquez",
    "nacionalidad": "Colombiana",
    "fecha_nacimiento": "1927-03-06",
    "biografia": "Escritor y periodista...",
    "imagen_autor": "gabriel.jpg"
  },
  {
    "id_autor": 2,
    "nombre": "Isabel",
    "apellido": "Allende",
    "nacionalidad": "Chilena",
    "fecha_nacimiento": "1942-08-02",
    "biografia": "Autora de novelas...",
    "imagen_autor": "isabel.jpg"
  }
]
```

2. Obtener un autor por ID

- **Descripción:** Obtiene la información de un autor específico por su ID.
- **Método:** GET
- **Endpoint:** /autores/{id}
- **Parámetros de ruta:**
 - id: ID del autor que se desea obtener.
- **Autenticación requerida:** No.
- **Respuesta exitosa:**

```
{
  "status": 200,
  "detalle": {
    "id_autor": 1,
    "nombre": "Gabriel",
    "apellido": "García Márquez",
    "nacionalidad": "Colombiana",
    "fecha_nacimiento": "1927-03-06",
    "biografia": "Escritor y periodista...",
    "imagen_autor": "gabriel.jpg"
  }
}
```

- **Respuesta de error:**
 - Si el autor no se encuentra:

```
{
  "status": 404,
  "detalle": "El autor no existe"
}
```

3. Crear un nuevo autor

- **Descripción:** Crea un nuevo autor en la base de datos.
- **Método:** POST
- **Endpoint:** /autores
- **Cuerpo de la solicitud:**

```
{
  "nombre": "Gabriel",
  "apellido": "García Márquez",
  "nacionalidad": "Colombiana",
  "fecha_nacimiento": "1927-03-06",
  "biografia": "Escritor y periodista...",
  "imagen_autor": "gabriel.jpg"
}
```

- **Autenticación requerida:** No.
- **Respuesta exitosa:**

```
{
  "status": 200,
  "detalle": "registro exitoso",
  "autor": {
    "nombre": "Gabriel",
    "apellido": "García Márquez",
    "nacionalidad": "Colombiana",
    "fecha_nacimiento": "1927-03-06",
    "biografia": "Escritor y periodista...",
    "imagen_autor": "gabriel.jpg"
  }
}
```

- **Respuesta de error:**
 - Si faltan datos obligatorios o el autor ya existe:

```
{
  "status": 400,
  "detalle": "Faltan datos obligatorios"
}
```

4. Actualizar un autor existente

- **Descripción:** Actualiza los datos de un autor específico.
- **Método:** PUT
- **Endpoint:** /autores/{id}
- **Parámetros de ruta:**
 - id: ID del autor que se desea actualizar.
- **Cuerpo de la solicitud:**

```
{
  "nombre": "Gabriel",
  "apellido": "García Márquez",
  "nacionalidad": "Colombiana",
  "fecha_nacimiento": "1927-03-06",
  "biografia": "Escritor y periodista destacado...",
  "imagen_autor": "gabriel_actualizado.jpg"
}
```

- **Autenticación requerida:** No.
- **Respuesta exitosa:**

```
{
  "status": 200,
  "detalle": "actualizacion completa",
  "autor": {
    "nombre": "Gabriel",
    "apellido": "García Márquez",
    "nacionalidad": "Colombiana",
    "fecha_nacimiento": "1927-03-06",
    "biografia": "Escritor y periodista destacado...",
    "imagen_autor": "gabriel_actualizado.jpg"
  }
}
```

- **Respuesta de error:**
 - Si el ID es incorrecto o faltan datos:

```
{
  "status": 400,
  "detalle": "ID requerido para actualizar"
}
```

5. Eliminar un autor

- **Descripción:** Elimina un autor específico de la base de datos.
- **Método:** DELETE
- **Endpoint:** /autores/{id}
- **Parámetros de ruta:**
 - id: ID del autor que se desea eliminar.
- **Autenticación requerida:** No.
- **Respuesta exitosa:**

```
{
  "status": 200,
  "detalle": "Eliminacion completa"
}
```

- **Respuesta de error:**
 - Si el ID no es válido o no se proporciona:

```
{
  "status": 400,
  "detalle": "ID requerido para eliminar"
}
```