CONCEPTUALIZACIÓN DE ENTORNOS DE DESARROLLO DE APLICACIONES Y SERVICIOS

2.6. PUT y DELETE

UDGVIRTUAL®

Axel Mauricio Barraza Cárdenas

06/10/2024

Editar las Rutas en routes/api.php

Primero, agregamos las rutas necesarias para manejar las solicitudes HTTP que permiten interactuar con los datos de películas.

- PUT /movies/{id}: Actualiza una película existente.
- DELETE /movies/{id}: Elimina una película.

```
15  Route::put('/movies/{id}', [MovieController::class, 'updateMovie']);
16  Route::delete("/movies/{id}", [MovieController::class, 'delete']);
17
18
```

Editar el Controlador MovieController

Luego, agregamos las funciones necesarias en el controlador MovieController para manejar las operaciones de películas:

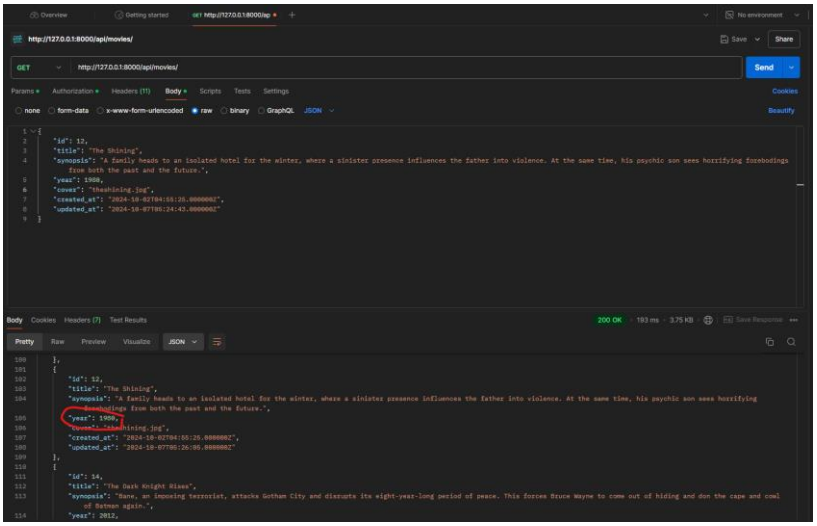
1 updateMovie:

- **Propósito**: Actualizar una película existente.
- **Ruta**: PUT /api/movies/{id}

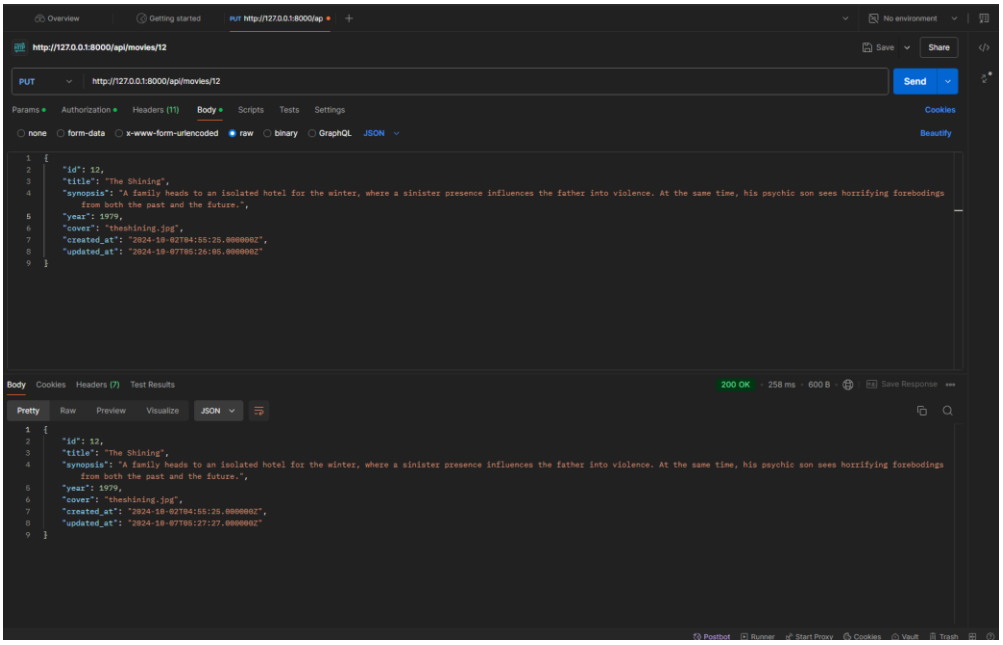
```
38      public function updateMovie(Request $request, $id)
39      {
40          $movie = Movie::find($id);
41          if ($movie) {
42              $movie->update([
43                  'title' => $request->title,
44                  'synopsis' => $request->synopsis,
45                  'year' => $request->year,
46                  'cover' => $request->cover,
47              ]);
48              return response()->json($movie, 200);
49          } else {
50              return response()->json(['message' => 'Movie not found'], 404);
51          }
52      }
53
```

Evidencia API:

Todas las peliculas



Se le cambio la fecha de 1980 a 1979:



2 delete:

- **Propósito**: Eliminar una película de la base de datos.
- **Ruta**: DELETE /api/movies/{id}

```php
public function delete($id)
{
    // Find the movie by ID
    $movie = Movie::find($id);

    // Check if the movie exists
    if (!$movie) {
        return response()->json(['message' => 'Movie not found'], 404);
    }

    // Delete the movie
    $movie->delete();

    // Return a success response
    return response()->json(['message' => 'Movie deleted successfully'], 200);
}
```

Evidencia API: