

## All Classes and Interfaces

[All Classes and Interfaces](#)

[Interfaces](#)

[Classes](#)

[Enum Classes](#)

[Exception Classes](#)

### Class

#### Description

##### **AccessDeniedException**

Excepción para acceso denegado (falta de permisos).

##### **AddMemberRequestDTO**

DTO para agregar un nuevo miembro a un viaje.

##### **BackendApplication**

Clase principal para iniciar la aplicación Spring Boot.

##### **ChangeMemberRoleRequestDTO**

DTO para cambiar el rol de un miembro en un viaje.

##### **CreateSplitRequestDTO**

DTO para crear una nueva división de gasto.

##### **CustomUserDetailsService**

Servicio para cargar detalles de usuario desde la base de datos.

##### **DuplicateResourceException**

Excepción para operaciones duplicadas (ej: usuario ya existe).

##### **Expense**

Entidad que representa un gasto en un viaje.

##### **ExpenseController**

Controlador REST para gestionar gastos en viajes.

##### **ExpenseCreateRequestDTO**

DTO para crear un nuevo gasto.

##### **ExpenseDTO**

DTO para información completa de un gasto.

##### **ExpenseRepository**

##### **ExpenseService**

##### **ExpenseSplit**

Entidad que representa la división de un gasto entre un usuario.

##### **ExpenseSplitController**

Controlador REST para gestionar divisiones de gastos.

##### **ExpenseSplitDTO**

DTO para una división de gasto.

##### **ExpenseSplitRepository**

##### **ExpenseSplitService**

Servicio para gestionar las divisiones de gastos.

##### **ExpenseSplitType**

Enum que define cómo se divide un gasto entre los participantes.

##### **GlobalExceptionHandler**

Manejador global de excepciones para la API REST.

## **JwtAuthenticationFilter**

### **JwtTokenProvider**

#### **LoginRequestDTO**

DTO para solicitud de login.

#### **LoginResponseDTO**

DTO para respuesta de login (contiene el token JWT).

#### **OpenApiConfig**

Configuración de OpenAPI/Swagger para la API REST.

#### **ResourceNotFoundException**

Excepción base para errores de recursos no encontrados.

#### **SecurityConfig**

Configuración de Spring Security con JWT y CORS

#### **Trip**

Entidad que representa un viaje colaborativo.

#### **TripController**

Controlador REST para gestionar viajes.

#### **TripCreateRequestDTO**

DTO para crear un nuevo viaje.

#### **TripDTO**

DTO para información completa de un viaje.

#### **TripMember**

Entidad que representa la relación entre un Usuario y un Viaje.

#### **TripMemberController**

Controlador REST para gestionar miembros de viajes.

#### **TripMemberDTO**

DTO para información de un miembro de un viaje.

#### **TripMemberRepository**

#### **TripMemberRole**

Enumeración que define los roles de un usuario dentro de un viaje.

#### **TripMemberService**

Servicio para gestionar la membresía de usuarios en viajes.

#### **TripRepository**

#### **TripService**

#### **User**

Entidad que representa un usuario en el sistema.

#### **UserController**

Controlador REST para gestionar usuarios.

#### **UserCreateRequestDTO**

DTO para crear/registrar un nuevo usuario.

#### **UserDTO**

DTO para información pública de un usuario.

#### **UserRepository**

**UserRole**

Enum que define los roles de los usuarios en el sistema.

**UserService****ValidationException**

Excepción para errores de negocio (ej: viaje con fechas inválidas).