

# All Classes and Interfaces

All Classes and Interfaces	Interfaces	Classes	Enum Classes	Exception Classes
Class				
Description				
<b>AccessDeniedException</b>				
Excepción para acceso denegado (falta de permisos).				
<b>AddMemberRequestDTO</b>				
DTO para agregar un nuevo miembro a un viaje.				
<b>BackendApplication</b>				
Clase principal para iniciar la aplicación Spring Boot.				
<b>ChangeMemberRoleRequestDTO</b>				
DTO para cambiar el rol de un miembro en un viaje.				
<b>CorsConfig</b>				
Configuración de CORS (Cross-Origin Resource Sharing) Permite que el frontend pueda hacer solicitudes al backend				
<b>CreateSplitRequestDTO</b>				
DTO para crear una nueva división de gasto.				
<b>CustomUserDetailsService</b>				
Servicio para cargar detalles de usuario desde la base de datos.				
<b>DuplicateResourceException</b>				
Excepción para operaciones duplicadas (ej: usuario ya existe).				
<b>ErrorResponseDTO</b>				
DTO para respuestas de error de la API.				
<b>Expense</b>				
Entidad que representa un gasto en un viaje.				
<b>ExpenseController</b>				
Controlador REST para gestionar gastos en viajes.				
<b>ExpenseCreateRequestDTO</b>				
DTO para crear un nuevo gasto.				
<b>ExpenseDTO</b>				
DTO para información completa de un gasto.				
<b>ExpenseRepository</b>				
<b>ExpenseService</b>				
<b>ExpenseSplit</b>				
Entidad que representa la división de un gasto entre un usuario.				
<b>ExpenseSplitController</b>				
Controlador REST para gestionar divisiones de gastos.				
<b>ExpenseSplitDTO</b>				
DTO para una división de gasto.				
<b>ExpenseSplitRepository</b>				
<b>ExpenseSplitService</b>				
Servicio para gestionar las divisiones de gastos.				

<b>ExpenseSplitType</b>
Enum que define cómo se divide un gasto entre los participantes.
<b>GlobalExceptionHandler</b>
Manejador global de excepciones para la API REST.
<b>HealthController</b>
Controlador para verificar el estado de los servicios del backend Usado por la página de estado para comprobar que el backend está disponible
<b>JwtAuthenticationFilter</b>
<b>JwtTokenProvider</b>
<b>LoginRequestDTO</b>
DTO para solicitud de login.
<b>LoginResponseDTO</b>
DTO para respuesta de login (contiene el token JWT y datos del usuario).
<b>OpenApiConfig</b>
Configuración de OpenAPI/Swagger para la API REST.
<b>ResourceNotFoundException</b>
Excepción base para errores de recursos no encontrados.
<b>SecurityConfig</b>
Configuración de Spring Security con JWT y CORS
<b>Trip</b>
Entidad que representa un viaje colaborativo.
<b>TripController</b>
Controlador REST para gestionar viajes.
<b>TripCreateRequestDTO</b>
DTO para crear un nuevo viaje.
<b>TripDTO</b>
DTO para información completa de un viaje.
<b>TripMember</b>
Entidad que representa la relación entre un Usuario y un Viaje.
<b>TripMemberController</b>
Controlador REST para gestionar miembros de viajes.
<b>TripMemberDTO</b>
DTO para información de un miembro de un viaje.
<b>TripMemberRepository</b>
<b>TripMemberRole</b>
Enumeración que define los roles de un usuario dentro de un viaje.
<b>TripMemberService</b>
Servicio para gestionar la membresía de usuarios en viajes.
<b>TripRepository</b>
<b>TripService</b>
<b>User</b>
Entidad que representa un usuario en el sistema.
<b>UserController</b>

Controlador REST para gestionar usuarios.
<b>UserCreateRequestDTO</b> DTO para crear/registrar un nuevo usuario.
<b>UserDTO</b> DTO para información pública de un usuario.
<b>UserRepository</b>
<b>UserRole</b> Enum que define los roles de los usuarios en el sistema.
<b>UserService</b>
<b>ValidationException</b> Excepción para errores de negocio (ej: viaje con fechas inválidas).