

Package com.mapmyjourney.backend.repository

Interface ActivityRepository

All Superinterfaces:

org.springframework.data.repository.CrudRepository<Activity,String[↗]>,
org.springframework.data.jpa.repository.JpaRepository<Activity,String[↗]>,
org.springframework.data.repository.ListCrudRepository<Activity,String[↗]>,
org.springframework.data.repository.ListPagingAndSortingRepository<Activity,String[↗]>,
org.springframework.data.repository.PagingAndSortingRepository<Activity,String[↗]>,
org.springframework.data.repository.query.QueryByExampleExecutor<Activity>,
org.springframework.data.repository.Repository<Activity,String[↗]>

@Repository
public interface **ActivityRepository**
extends org.springframework.data.jpa.repository.JpaRepository<Activity,String[↗]>

Repositorio para la entidad Activity. Proporciona métodos de acceso a datos para las actividades.

Method Summary

All Methods	Instance Methods	Abstract Methods
Modifier and Type		
Method		
Description		
boolean		
existsByTripIdAndActivityDate (Long [↗] tripId, LocalDate [↗] date)		
Verifica si existe una actividad para un viaje y una fecha.		
Optional [↗] <Activity>		
findByIdAndTripId (String [↗] id, Long [↗] tripId)		
Obtiene una actividad por ID y verifica que pertenece a un viaje específico.		
List [↗] <Activity>		
findByTripId (Long [↗] tripId)		
Obtiene todas las actividades de un viaje, ordenadas por fecha y hora.		
List [↗] <Activity>		
findByTripIdAndDate (Long [↗] tripId, LocalDate [↗] date)		
Obtiene las actividades de un viaje para una fecha específica.		
List [↗] <Activity>		
findByTripIdAndDateRange (Long [↗] tripId, LocalDate [↗] startDate, LocalDate [↗] endDate)		
Obtiene las actividades de un viaje para un rango de fechas.		
List [↗] <Activity>		
findByTripIdAndDayIndex (Long [↗] tripId, Integer [↗] dayIndex)		
Obtiene las actividades de un viaje para un día índice específico.		
Integer [↗]		
findMaxOrderByTripIdAndDate (Long [↗] tripId, LocalDate [↗] date)		
Obtiene el máximo orden de una actividad para una fecha.		

Methods inherited from interface org.springframework.data.repository.CrudRepository

count, delete, deleteAll, deleteAll, deleteAllById, deleteById, existsById, findById, save

Methods inherited from interface org.springframework.data.jpa.repository.JpaRepository

deleteAllByIdInBatch, deleteAllInBatch, deleteAllInBatch, deleteInBatch, findAll, findAll, flush, getById, getOne, getReferenceById, saveAllAndFlush, saveAndFlush

Methods inherited from interface org.springframework.data.repository.ListCrudRepository

findAll, findById, saveAll

Methods inherited from interface org.springframework.data.repository.ListPagingAndSortingRepository

findAll

Methods inherited from interface org.springframework.data.repository.PagingAndSortingRepository

findAll

Methods inherited from interface org.springframework.data.repository.query.QueryByExampleExecutor

count, exists, findAll, findBy, findOne

Method Details

findByTripId

```
@Query("SELECT a FROM Activity a WHERE a.trip.id = :tripId ORDER BY a.activityDate, a.startTime")
List<Activity> findByTripId(@Param("tripId")
                           Long tripId)
```

Obtiene todas las actividades de un viaje, ordenadas por fecha y hora.

findByTripIdAndDate

```
@Query("SELECT a FROM Activity a WHERE a.trip.id = :tripId AND a.activityDate = :date ORDER BY a.orderIndex, a.startTime")
List<Activity> findByTripIdAndDate(@Param("tripId")
                                   Long tripId,
                                   @Param("date")
                                   LocalDate date)
```

Obtiene las actividades de un viaje para una fecha específica.

findByTripIdAndDateRange

```
@Query("SELECT a FROM Activity a WHERE a.trip.id = :tripId AND a.activityDate BETWEEN :startDate AND :endDate ORDER BY a.activityDate, a.startTime")
List<Activity> findByTripIdAndDateRange(@Param("tripId")
                                       Long tripId,
                                       @Param("startDate")
                                       LocalDate startDate,
                                       @Param("endDate")
                                       LocalDate endDate)
```

Obtiene las actividades de un viaje para un rango de fechas.

findByTripIdAndDayIndex

```
@Query("SELECT a FROM Activity a WHERE a.trip.id = :tripId AND a.dayIndex = :dayIndex ORDER BY a.orderIndex, a.startTime")
List<Activity> findByTripIdAndDayIndex(@Param("tripId")
                                       Long tripId,
                                       @Param("dayIndex")
                                       Integer dayIndex)
```

Obtiene las actividades de un viaje para un día índice específico.

existsByTripIdAndActivityDate

```
boolean existsByTripIdAndActivityDate(Long tripId,
                                       LocalDate date)
```

Verifica si existe una actividad para un viaje y una fecha.

findMaxOrderByTripIdAndDate

```
@Query("SELECT MAX(a.orderIndex) FROM Activity a WHERE a.trip.id = :tripId AND a.activityDate = :date")
Integer findMaxOrderByTripIdAndDate(@Param("tripId")
                                    Long tripId,
                                    @Param("date")
                                    LocalDate date)
```

Obtiene el máximo orden de una actividad para una fecha.

findByIdAndTripId

```
@Query("SELECT a FROM Activity a WHERE a.id = :id AND a.trip.id = :tripId")
Optional<Activity> findByIdAndTripId(@Param("id")
                                    String id,
                                    @Param("tripId")
                                    Long tripId)
```

Obtiene una actividad por ID y verifica que pertenece a un viaje específico.