

Package com.mapmyjourney.backend.service

## Class ActivityService

java.lang.Object  
com.mapmyjourney.backend.service.ActivityService

```
@Service
public class ActivityService
extends Object
```

Servicio para gestionar las operaciones de las actividades.

### Constructor Summary

#### Constructors

##### Constructor

##### Description

ActivityService()

### Method Summary

#### All Methods

#### Instance Methods

#### Concrete Methods

##### Modifier and Type

##### Method

##### Description

##### ActivityDTO

`createActivity(Long tripId, CreateActivityRequestDTO dto)`

Crea una nueva actividad en un viaje.

void

`deleteActivity(Long tripId, String activityId)`

Elimina una actividad.

##### List<ActivityDTO>

`getActivitiesByDate(Long tripId, LocalDate date)`

Obtiene las actividades de un viaje para una fecha específica.

##### List<ActivityDTO>

`getActivitiesByTrip(Long tripId)`

Obtiene todas las actividades de un viaje.

##### ActivityDTO

`getActivity(Long tripId, String activityId)`

Obtiene una actividad específica.

void

`reorderActivities(Long tripId, LocalDate date, List<String> activityIds)`

Reordena las actividades de una fecha.

##### ActivityDTO

`updateActivity(Long tripId, String activityId, UpdateActivityRequestDTO dto)`

Actualiza una actividad existente.

### Methods inherited from class java.lang.Object

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`

## ***Constructor Details***

## ActivityService

```
public ActivityService()
```

## ***Method Details***

## getActivitiesByTrip

```
@Transactional(readOnly=true)
public List<ActivityDTO> getActivitiesByTrip(Long tripId)
```

Obtiene todas las actividades de un viaje.

### **getActivitiesByDate**

Obtiene las actividades de un viaje para una fecha específica.

### **getActivity**

```
@Transactional(readOnly=true)
public ActivityDTO getActivity(Long tripId,
                               String activityId)
```

Obtiene una actividad específica

### **createActivity**

Crea una nueva actividad en un viaje

### **updateActivity**

Actualiza una actividad existente

### **deleteActivity**

```
@Transactional  
public void deleteActivity(Long tripId,  
                           String activityId)
```

#### Elimina una actividad

#### **reorderActivities**

Reordena las actividades de una fecha.