

Package com.mapmyjourney.backend.controller

Class ActivityController

java.lang.Object
com.mapmyjourney.backend.controller.ActivityController

```
@RestController
@RequestMapping("/api/trips/{tripId}/activities")
public class ActivityController
extends Object
```

Controlador REST para gestionar las actividades del itinerario de un viaje. Endpoints: GET /api/trips/{tripId}/activities - Obtener todas las actividades de un viaje GET /api/trips/{tripId}/activities/{activityId} - Obtener una actividad específica POST /api/trips/{tripId}/activities - Crear una nueva actividad PUT /api/trips/{tripId}/activities/{activityId} - Actualizar una actividad DELETE /api/trips/{tripId}/activities/{activityId} - Eliminar una actividad

Constructor Summary

Constructors

Constructor

Description

ActivityController()

Method Summary

All Methods

Instance Methods

Concrete Methods

Modifier and Type

Method

Description

```
org.springframework.http.ResponseEntity<ActivityDTO>
createActivity(Long tripId, @Valid CreateActivityRequestDTO dto)
```

Crea una nueva actividad en un viaje.

```
org.springframework.http.ResponseEntity<Void>
deleteActivity(Long tripId, String activityId)
```

Elimina una actividad.

```
org.springframework.http.ResponseEntity<List<ActivityDTO>>
```

```
getActivitiesByDate(Long tripId, LocalDate date)
```

Obtiene las actividades de un viaje para una fecha específica.

```
org.springframework.http.ResponseEntity<List<ActivityDTO>>
```

```
getActivitiesByTrip(Long tripId)
```

Obtiene todas las actividades de un viaje.

```
org.springframework.http.ResponseEntity<ActivityDTO>
```

```
getActivity(Long tripId, String activityId)
```

Obtiene una actividad específica.

```
org.springframework.http.ResponseEntity<Void>
```

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

Reordena las actividades de una fecha específica.

```
org.springframework.http.ResponseEntity<ActivityDTO>
```

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

Actualiza una actividad existente.

Methods inherited from class java.lang.Object

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

Constructor Details

ActivityController

```
public ActivityController()
```

Method Details

getActivitiesByTrip

```
@GetMapping  
public org.springframework.http.ResponseEntity<List<ActivityDTO>> getActivitiesByTrip(@PathVariable  
Long tripId)
```

Obtiene todas las actividades de un viaje.

Parameters:

tripId - ID del viaje

Returns:

Lista de actividades

getActivitiesByDate

```
@GetMapping("/date")  
public org.springframework.http.ResponseEntity<List<ActivityDTO>> getActivitiesByDate(@PathVariable  
Long tripId,  
@RequestParam  
LocalDate date)
```

Obtiene las actividades de un viaje para una fecha específica.

Parameters:

tripId - ID del viaje

date - Fecha en formato ISO (YYYY-MM-DD)

Returns:

Lista de actividades para esa fecha

getActivity

```
@GetMapping("/{activityId}")  
public org.springframework.http.ResponseEntity<ActivityDTO> getActivity(@PathVariable  
Long tripId,  
@PathVariable  
String activityId)
```

Obtiene una actividad específica.

Parameters:

tripId - ID del viaje

activityId - ID de la actividad

Returns:

Datos de la actividad

createActivity

```
@PostMapping  
public org.springframework.http.ResponseEntity<ActivityDTO> createActivity
```

```
(@PathVariable  
Long tripId,  
@Valid @RequestBody  
@Valid CreateActivityRequestDTO dto)
```

Crea una nueva actividad en un viaje.

Parameters:

tripId - ID del viaje

dto - Datos de la nueva actividad

Returns:

Actividad creada

updateActivity

```
@PutMapping("/{activityId}")  
public org.springframework.http.ResponseEntity<ActivityDTO> updateActivity  
(@PathVariable  
Long tripId,  
@PathVariable  
String activityId,  
@Valid @RequestBody  
@Valid UpdateActivityRequestDTO dto)
```

Actualiza una actividad existente.

Parameters:

tripId - ID del viaje

activityId - ID de la actividad a actualizar

dto - Datos a actualizar

Returns:

Actividad actualizada

deleteActivity

```
@DeleteMapping("/{activityId}")  
public org.springframework.http.ResponseEntity<Void> deleteActivity(@PathVariable  
Long tripId,  
@PathVariable  
String activityId)
```

Elimina una actividad.

Parameters:

tripId - ID del viaje

activityId - ID de la actividad a eliminar

Returns:

Respuesta sin contenido

reorderActivities

```
@PutMapping("/reorder")  
public org.springframework.http.ResponseEntity<Void> reorderActivities(@PathVariable  
Long tripId,  
@RequestParam  
LocalDate date,  
@RequestBody  
List<String> activityIds)
```

Reordena las actividades de una fecha específica.

Parameters:

tripId - ID del viaje

date - Fecha en formato ISO (YYYY-MM-DD)

activityIds - Lista de IDs de actividades en el nuevo orden

Returns:

Respuesta sin contenido