

**Package** com.mapmyjourney.backend.service

## Class ExpenseSplitService

java.lang.Object<sup>↗</sup>  
com.mapmyjourney.backend.service.ExpenseSplitService

@Service  
public class **ExpenseSplitService**  
extends Object<sup>↗</sup>

Servicio para gestionar las divisiones de gastos. Maneja la creación, actualización y liquidación de divisiones de gastos.

### Constructor Summary

Constructors
Constructor
Description
<a href="#">ExpenseSplitService()</a>

### Method Summary

All Methods	Instance Methods	Concrete Methods
Modifier and Type		
Method		
Description		
<b>ExpenseSplitDT0</b>		
<b>createSplit</b> ( <a href="#">Long</a> <sup>↗</sup> expenseId, <a href="#">Long</a> <sup>↗</sup> participantUserId, <a href="#">BigDecimal</a> <sup>↗</sup> amount, <a href="#">BigDecimal</a> <sup>↗</sup> percentage)		
Crea una nueva división de gasto.		
void		
<b>deleteSplit</b> ( <a href="#">Long</a> <sup>↗</sup> splitId)		
Elimina una división de gasto.		
<b>List</b> <sup>↗</sup> < <a href="#">ExpenseSplitDT0</a> >		
<b>getExpenseSplits</b> ( <a href="#">Long</a> <sup>↗</sup> expenseId)		
Obtiene todas las divisiones de un gasto.		
<b>ExpenseSplitDT0</b>		
<b>getSplit</b> ( <a href="#">Long</a> <sup>↗</sup> splitId)		
Obtiene una división específica por ID.		
<b>BigDecimal</b> <sup>↗</sup>		
<b>getTotalPendingDebt</b> ( <a href="#">Long</a> <sup>↗</sup> userId)		
Calcula el total de deudas pendientes de un usuario.		
<b>List</b> <sup>↗</sup> < <a href="#">ExpenseSplitDT0</a> >		
<b>getUserPendingDebts</b> ( <a href="#">Long</a> <sup>↗</sup> userId)		
Obtiene todas las deudas pendientes de un usuario.		
<b>ExpenseSplitDT0</b>		
<b>markAsPaid</b> ( <a href="#">Long</a> <sup>↗</sup> splitId)		
Marca una división como pagada.		
<b>ExpenseSplitDT0</b>		
<b>markAsUnpaid</b> ( <a href="#">Long</a> <sup>↗</sup> splitId)		

Marca una división como no pagada.

**ExpenseSplitDTO**

`updateSplitAmount(Long splitId, BigDecimal newAmount)`

Actualiza el monto de una división.

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

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

**Constructor Details**

**ExpenseSplitService**

`public ExpenseSplitService()`

**Method Details**

**createSplit**

```
@Transactional
public ExpenseSplitDTO createSplit(Long expenseId,
                                   Long participantUserId,
                                   BigDecimal amount,
                                   BigDecimal percentage)
```

Crea una nueva división de gasto.

**Parameters:**

expenseId - ID del gasto

participantUserId - ID del usuario participante

amount - Monto que debe pagar este usuario

percentage - Porcentaje (opcional, para divisiones por porcentaje)

**Returns:**

DTO de la división creada

**getExpenseSplits**

```
@Transactional(readOnly=true)
public List<ExpenseSplitDTO> getExpenseSplits(Long expenseId)
```

Obtiene todas las divisiones de un gasto.

**Parameters:**

expenseId - ID del gasto

**Returns:**

Lista de DTOs de divisiones

**getUserPendingDebts**

```
@Transactional(readOnly=true)
public List<ExpenseSplitDTO> getUserPendingDebts(Long userId)
```

Obtiene todas las deudas pendientes de un usuario.

**Parameters:**

userId - ID del usuario

**Returns:**

Lista de DTOs de divisiones pendientes

## markAsPaid

```
@Transactional
public ExpenseSplitDTO markAsPaid(Long splitId)
```

Marca una división como pagada.

**Parameters:**

splitId - ID de la división

**Returns:**

DTO de la división actualizada

## markAsUnpaid

```
@Transactional
public ExpenseSplitDTO markAsUnpaid(Long splitId)
```

Marca una división como no pagada.

**Parameters:**

splitId - ID de la división

**Returns:**

DTO de la división actualizada

## getSplit

```
@Transactional(readonly=true)
public ExpenseSplitDTO getSplit(Long splitId)
```

Obtiene una división específica por ID.

**Parameters:**

splitId - ID de la división

**Returns:**

DTO de la división

## deleteSplit

```
@Transactional
public void deleteSplit(Long splitId)
```

Elimina una división de gasto.

**Parameters:**

splitId - ID de la división

## getTotalPendingDebt

```
@Transactional(readonly=true)
public BigDecimal getTotalPendingDebt(Long userId)
```

Calcula el total de deudas pendientes de un usuario.

**Parameters:**

userId - ID del usuario

**Returns:**

Total de deudas pendientes

## updateSplitAmount

```
@Transactional
public ExpenseSplitDTO updateSplitAmount(Long splitId,
                                           BigDecimal newAmount)
```

Actualiza el monto de una división.

**Parameters:**

splitId - ID de la división

newAmount - Nuevo monto

**Returns:**

DTO de la división actualizada