

Package com.mapmyjourney.backend.service

Class ExpenseSplitService

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

```
@Service  
public class ExpenseSplitService  
extends Object
```

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

Constructor Summary

Constructors

Constructor

Description

ExpenseSplitService()

Method Summary

All Methods

Instance Methods

Concrete Methods

Modifier and Type

Method

Description

ExpenseSplitDTO

createSplit(Long expenseId, Long participantUserId, BigDecimal amount, BigDecimal percentage)

Crea una nueva división de gasto.

void

deleteSplit(Long splitId)

Elimina una división de gasto.

List<ExpenseSplitDTO>

getExpenseSplits(Long expenseId)

Obtiene todas las divisiones de un gasto.

ExpenseSplitDTO

getSplit(Long splitId)

Obtiene una división específica por ID.

BigDecimal

getTotalPendingDebt(Long userId)

Calcula el total de deudas pendientes de un usuario.

List<ExpenseSplitDTO>

getUserPendingDebts(Long userId)

Obtiene todas las deudas pendientes de un usuario.

ExpenseSplitDTO

markAsPaid(Long splitId)

Marca una división como pagada.

ExpenseSplitDTO

markAsUnpaid(Long splitId)

Marca una división como no pagada.

ExpenseSplitDTO

`updateSplitAmount(Long2 splitId, BigDecimal2 newAmount)`

Actualiza el monto de una división.

Methods inherited from class java.lang.Object²

`clone2, equals2, finalize2, getClass2, hashCode2, notify2, notifyAll2, toString2, wait2, wait2, wait2`

Constructor Details

ExpenseSplitService

`public ExpenseSplitService()`

Method Details

createSplit

```
@Transactional  
public ExpenseSplitDTO createSplit(Long2 expenseId,  
                                    Long2 participantUserId,  
                                    BigDecimal2 amount,  
                                    BigDecimal2 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 List2<ExpenseSplitDTO> getExpenseSplits(Long2 expenseId)
```

Obtiene todas las divisiones de un gasto.

Parameters:

`expenseId` - ID del gasto

Returns:

Lista de DTOs de divisiones

getUserPendingDebts

```
@Transactional(readOnly=true)  
public List2<ExpenseSplitDTO> getUserPendingDebts(Long2 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(Long2 splitId)
```

Marca una división como pagada.

Parameters:

splitId - ID de la división

Returns:

DTO de la división actualizada

markAsUnpaid

```
@Transactional  
public ExpenseSplitDTO markAsUnpaid(Long2 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(Long2 splitId)
```

Obtiene una división específica por ID.

Parameters:

splitId - ID de la división

Returns:

DTO de la división

deleteSplit

```
@Transactional  
public void deleteSplit(Long2 splitId)
```

Elimina una división de gasto.

Parameters:

splitId - ID de la división

getTotalPendingDebt

```
@Transactional(readOnly=true)  
public BigDecimal2 getTotalPendingDebt(Long2 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(Long2 splitId,  
                                         BigDecimal2 newAmount)
```

Actualiza el monto de una división.

Parameters:

splitId - ID de la división

newAmount - Nuevo monto

Returns:

DTO de la división actualizada