@Transactional propagation cheatsheat! 👇

Lost in propagation? Save this!

```
@Transactional(propagation = Propagation.REQUIRED)
@Transactional(propagation = Propagation.REQUIRES_NEW)
@Transactional(propagation = Propagation.NESTED)
@Transactional(propagation = Propagation.SUPPORTS)
@Transactional(propagation = Propagation.NOT_SUPPORTED)
@Transactional(propagation = Propagation.NEVER)
@Transactional(propagation = Propagation.MANDATORY)
```

spring

Follow for more tech insights!

Required(default)

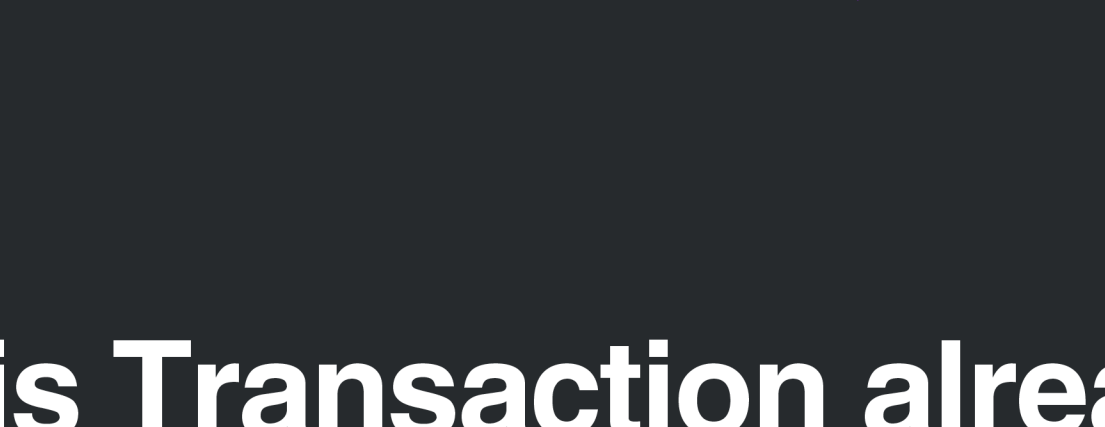
```
@Transactional
```

1. If no Transaction

Create a new transaction

2. If there is Transaction already

Join Current Transaction



Swipe Left

spring

Requires new

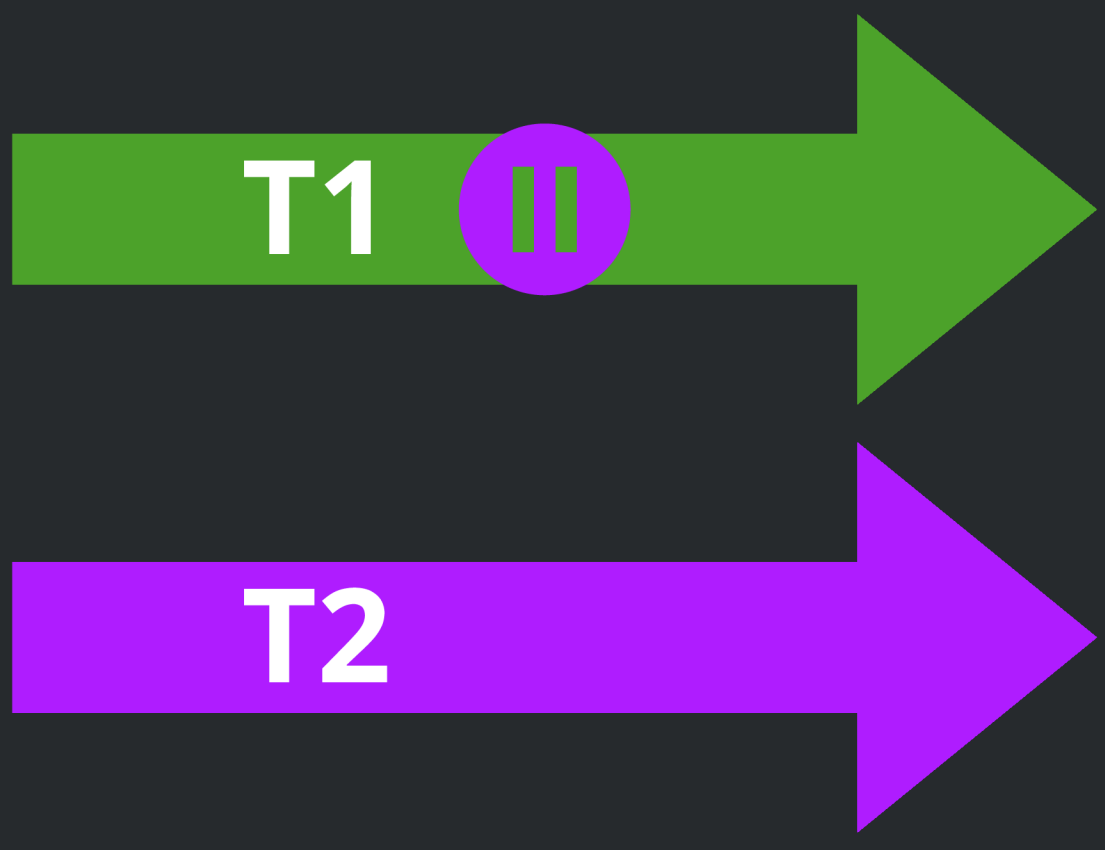
```
@Transactional(propagation = Propagation.REQUIRES_NEW)
```

1. If no Transaction

Create a new transaction

2. If there is Transaction already

Create a new transaction (T2) and suspends existing one(T1) (Till the end of T2)



Swipe Left

spring

Nested

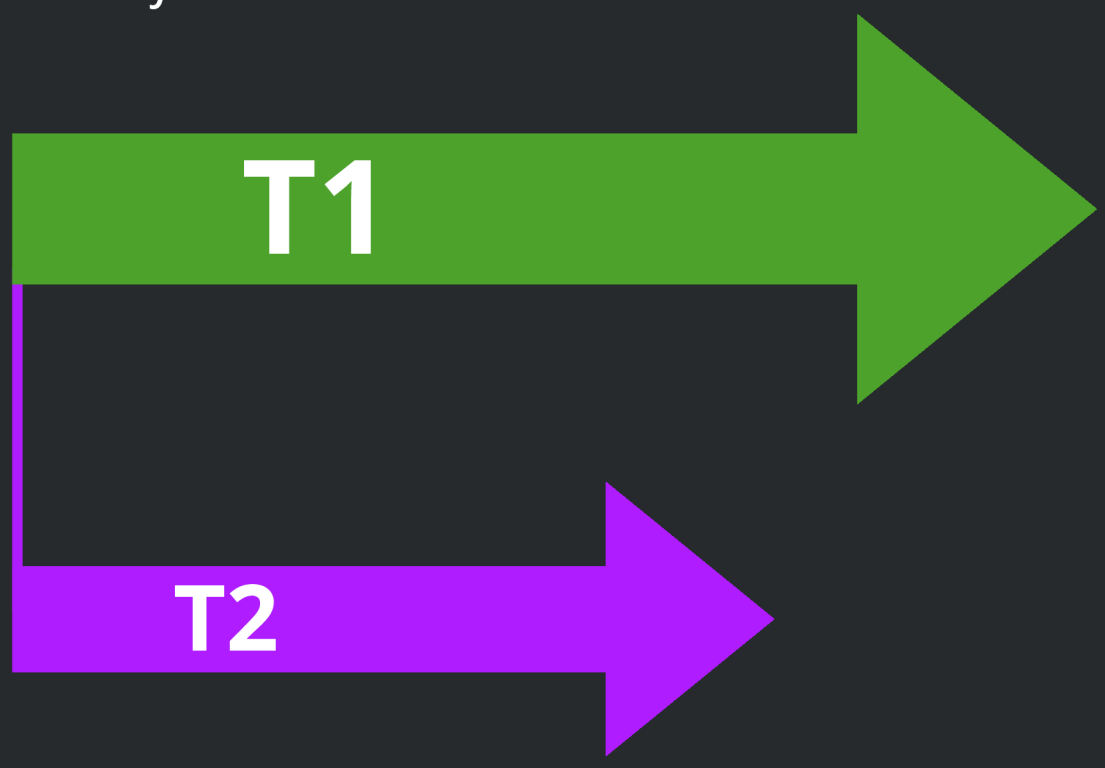
```
@Transactional(propagation = Propagation.NESTED)
```

1. If no Transaction

Create a new transaction

2. If there is Transaction already

Creates a nested transaction that is logically independent but still part of the main transaction



Swipe Left

spring

Supports

```
@Transactional(propagation = Propagation.SUPPORTS)
```

1. If no Transaction

Run code without any transaction

T1

2. If there is Transaction already

Join Current Transaction

T1 + T2

Swipe Left

spring

Not supported

```
@Transactional(propagation = Propagation.NOT_SUPPORTED)
```

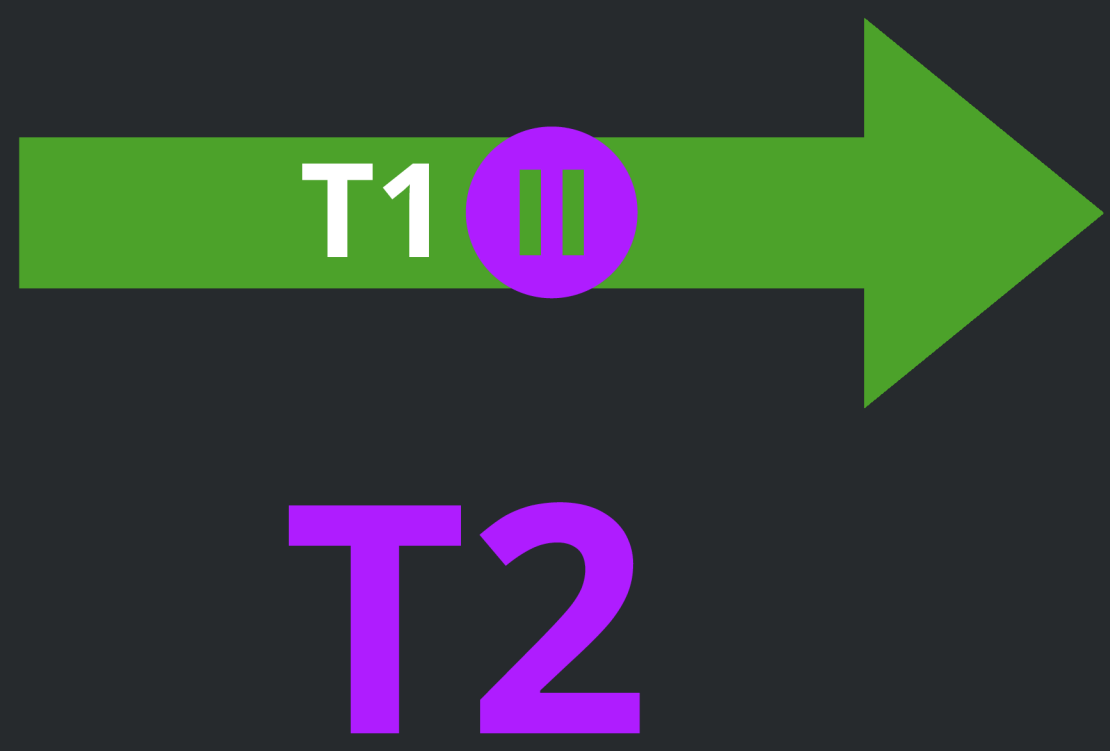
1. If no Transaction

Run code without any transaction

T1

2. If there is Transaction already

Run code without any transaction and suspend existing transaction(till the execution of T1 code)



Swipe Left

spring

Mandatory

```
@Transactional(propagation = Propagation.MANDATORY)
```

1. If no Transaction

Throw exception

2. If there is Transaction already

Join Current Transaction

T1 + T2

Swipe Left

NEVER

```
@Transactional(propagation = Propagation.NEVER)
```

1. If no Transaction

Run code without any transaction

T1

2. If there is Transaction already

Throw exception

Swipe Left

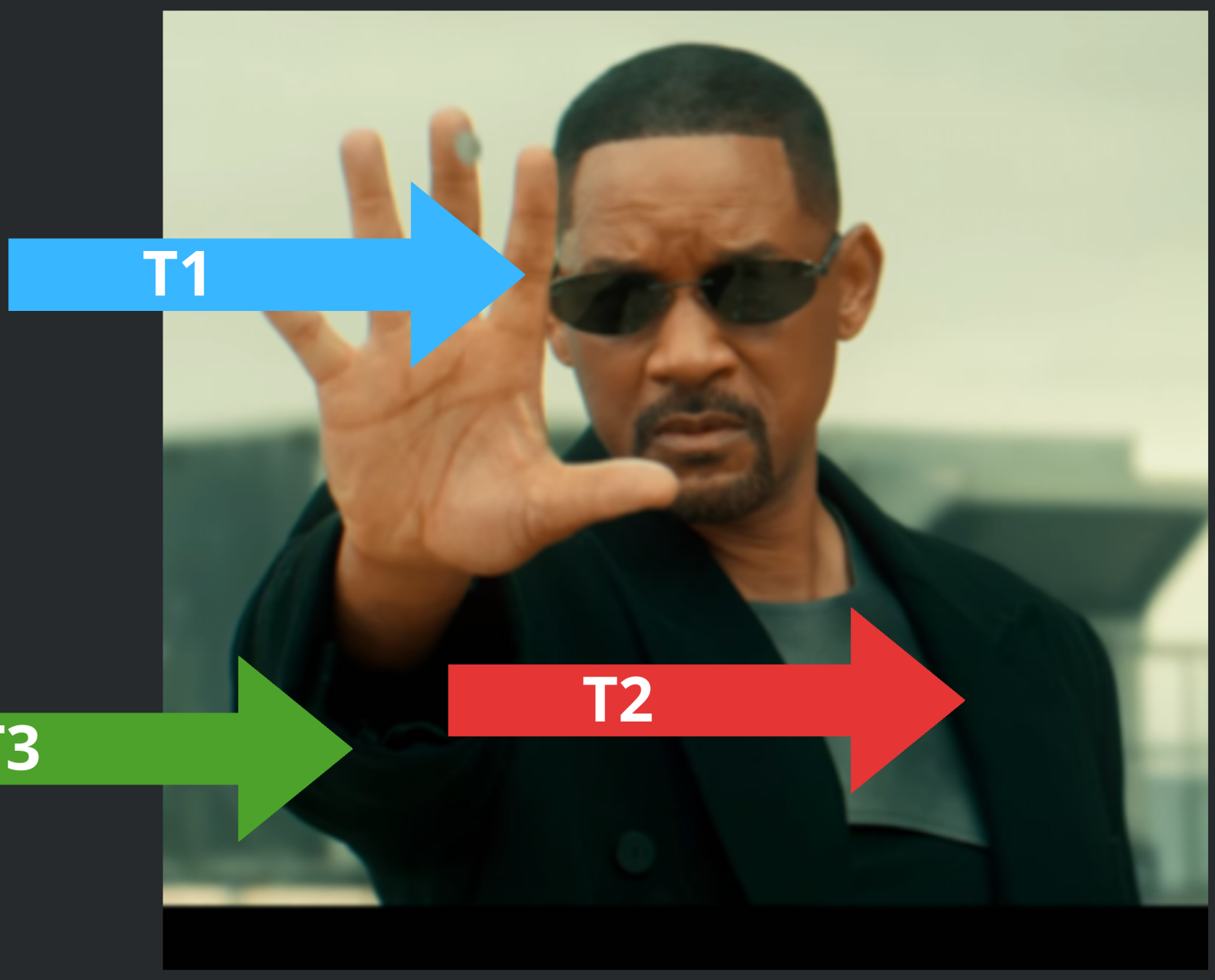
spring

Key Takeaways

✅Propagation defines how transactions interact.

✅Choose wisely: REQUIRED (default), REQUIRES_NEW (isolated), NESTED (savepoints), etc.

✅ Follow for more tech insights!



Control transactions like a boss!

spring