## **REQUIRED (Default Transaction Propagation):**

1. **If there is @Transactional above insertEmployee method and if it is called directly then it will create its own new transaction.**
2. **If the calling service has a transaction then method makes use of exiting transaction:**

Suppose you have @Transnational above joinOrganization method inside OrganzationServiceImpl where insertEmployee method is also called and if there is @Transnational above insertEmployee method inside EmployeeServiceImpl then it makes use of existing transaction ie of method joinOrganization .

1. **If the calling service does not have transaction then the method will create new transaction:**

Suppose you have not specified @Transnational above joinOrganization method inside OrganzationServiceImpl where insertEmployee method is also called and if there is @Transnational above insertEmployee method inside EmployeeServiceImpl then it will create new transaction ie of method insertEmployee .

## **SUPPORTS:**

1. **If there is @Transactional(propagation=Propagation.SUPPORTS) above insertEmployee method and if it is called directly then it will not create its own new transaction.**
2. **If the calling service has a transaction then method makes use of exiting transaction:**

Suppose you have @Transnational above joinOrganization method inside OrganzationServiceImpl where insertEmployee method is also called and if there is @Transactional(propagation=Propagation.SUPPORTS) above insertEmployee method inside EmployeeServiceImpl then it makes use of existing transaction ie of method joinOrganization .

1. **If the calling service does not have transaction then the method will not create a new transaction:**

Suppose you have not specified @Transnational above joinOrganization method inside OrganzationServiceImpl where insertEmployee method is also called and if there is @Transactional(propagation=Propagation.SUPPORTS) above insertEmployee method inside EmployeeServiceImpl then it will not create new transaction.

## **NOT\_SUPPORTED:**

1. **If there is @Transactional(propagation=Propagation.NOT\_SUPPORTED) above insertEmployee method and if it is called directly then it will not create its own new transaction.**
2. **If the calling service has a transaction then method does not make use of exiting transaction neither does it create its own transaction. It run without transaction:**

Suppose you have @Transnational above joinOrganization method inside OrganzationServiceImpl where insertEmployee method is also called and if there is @Transactional(propagation=Propagation.NOT\_SUPPORTED) above insertEmployee method inside EmployeeServiceImpl then it suspends the existing transaction, execute insertEmployee method without transaction and once execution of the method is completed then it resumes with suspended transaction.

1. **If the calling service does not have transaction then the method will not create a new transaction and run without transaction:**

Suppose you have not specified @Transnational above joinOrganization method inside OrganzationServiceImpl where insertEmployee method is also called and if there is @Transactional(propagation=Propagation.NOT\_SUPPORTED) above insertEmployee method inside EmployeeServiceImpl then it will not create new transaction.

## **REQUIRES\_NEW:**

1. **If there is @Transactional(propagation=Propagation.REQUIRES\_NEW) above insertEmployee method and if it is called directly then it will create its own new transaction.**
2. **If the calling service has a transaction then method does not make use of exiting transaction but creates its own transaction :**

Suppose you have @Transnational above joinOrganization method inside OrganzationServiceImpl where insertEmployee method is also called and if there is @Transactional(propagation=Propagation.REQUIRES\_NEW) above insertEmployee method inside EmployeeServiceImpl then it makes use of new transaction ie of method insertEmployee, suspends previous transaction of the calling service and always gives priority to the new transaction.

1. **If the calling service does not have transaction then the method will create a new transaction:**

Suppose you have not specified @Transnational above joinOrganization method inside OrganzationServiceImpl where insertEmployee method is also called and if there is @Transactional(propagation=Propagation.REQUIRES\_NEW) above insertEmployee method inside EmployeeServiceImpl then it will create new transaction.

## **NEVER:**

1. **If there is @Transactional(propagation=Propagation.NEVER) above insertEmployee method and if it is called directly then it will not create its own new transaction.**
2. **If the calling service has a transaction then method throws exception:**

Suppose you have @Transnational above joinOrganization method inside OrganzationServiceImpl where insertEmployee method is also called and if there is @Transactional(propagation=Propagation.NEVER) above insertEmployee method inside EmployeeServiceImpl then it will throw exception ie *Exception in thread "main" org.springframework.transaction.IllegalTransactionStateException: Existing transaction found for transaction marked with propagation 'never'*

1. **If the calling service does not have transaction then the method will not create a new transaction and run without transaction:**

Suppose you have not specified @Transnational above joinOrganization method inside OrganzationServiceImpl where insertEmployee method is also called and if there is @Transactional(propagation=Propagation.NEVER) above insertEmployee method inside EmployeeServiceImpl then it will not create new transaction.

## **MANDATORY:**

1. **If there is @Transactional(propagation=Propagation.MANDATORY) above insertEmployee method and if it is called directly then it will throw an exception.**
2. **If the calling service has a transaction then method makes use of exiting transaction:**

Suppose you have @Transnational above joinOrganization method inside OrganzationServiceImpl where insertEmployee method is also called and if there is @Transactional(propagation=Propagation.MANDATORY) above insertEmployee method inside EmployeeServiceImpl then it makes use of existing transaction ie of method joinOrganization .

1. **If the calling service does not have transaction then the method will throw an exception:**

Suppose you have not specified @Transnational above joinOrganization method inside OrganzationServiceImpl where insertEmployee method is also called and if there is @Transactional(propagation=Propagation.MANDATORY) above insertEmployee method inside EmployeeServiceImpl then it will throw an exception ie *Exception in thread "main" org.springframework.transaction.IllegalTransactionStateException: No existing transaction found for transaction marked with propagation 'mandatory'*