

Основы Java EE

Spring и базы данных

Основы Java EE

Автор курса



Бондаренко Андрей

Основы Java EE

После урока обязательно



Повторите этот урок в видео формате на
[ITVDN.com](http://itvdn.com)



Проверьте как Вы усвоили данный материал на
[TestProvider.com](http://testprovider.com)

Spring и базы данных

Spring

Spring & JDBC

Действие	Spring	Вы
Находит параметры Connection		X
Открывает соединения	X	
Встроенные SQL выражения		X
Подготовить и выполнить запрос	X	
Работа с перебором значений при выборке в цикле		X
Перехват транзакций	X	
Заккрытие Connection, Statement, ResultSet	X	

Spring

Spring Data

- JDBC
- Hibernate
- JPA

Spring

Spring Data

- Более высокий уровень программирования.
- Улучшенная работа с исключениями.
- Большую часть рутинной работы Spring берет на себя.
- Концентрация именно на бизнес процессах.
- Используются механизмы DI и AOP.
- Учтены все минусы технологий.

Шаблоны

- **Шаблон** – оболочка, которая делает всю рутинную работу.
- Типы шаблонов:
 - JDBC Template;
 - Hibernate Template;
 - JPA Template;
 - JMS Template.

Spring Data

DataSource

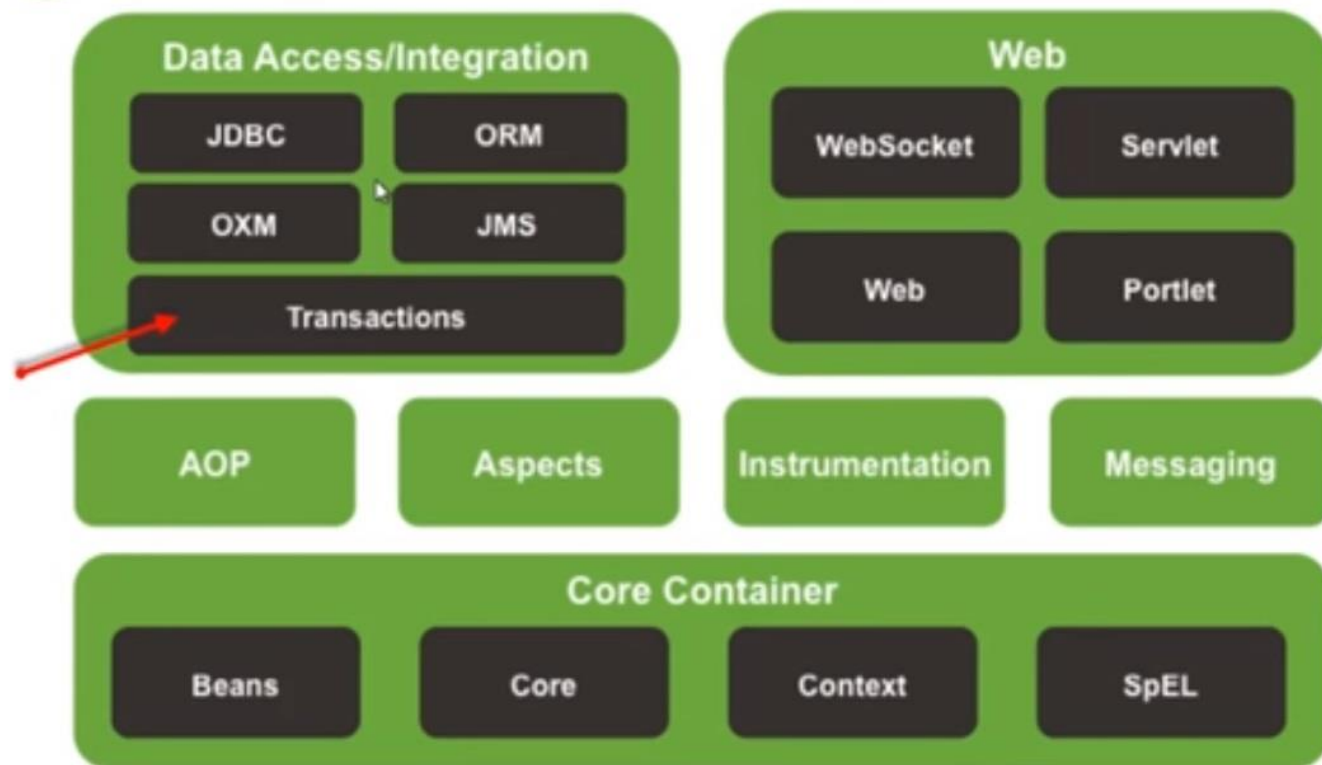
- DriverManagerDataSource – при каждом запросе новое соединение.
- SingleConnectionDataSource – одно соединение на все запросы (использовать желательно для тестирования).
- На проектах использовать пул соединений (используя JNDI).

Transactions

Транзакции



Spring Framework Runtime



Transactions

Транзакции

- Использование аспектов для внедрения транзакций.
- Поддержка аннотаций.

Transactions

Виды транзакций

- **Локальные** – одна система (например, только базой данных).
- **Глобальные** – несколько систем (база данных, JMS и пр.) – транзакция на уровне всех систем одновременно.

Transactions

Транзакции

- **Программная поддержка** – управление транзакциями напрямую в коде (сложнее в реализации, большая гибкость, готовые шаблоны в Spring).
- **Декларативная поддержка** – настройки и аннотации (используются чаще всего, проще в реализации, ограниченные возможности, аспекты или аннотации).

Transactions

Поддержка транзакций в Spring

- Поддержка транзакционности современных методов доступа к данным (JDBC, Hibernate, JPA).
- Настройка (XML – аспекты, аннотации).
- Транзакционность можно подключать уже после создания всего кода.

Transactions

Менеджеры транзакций

- `DataSourceTransactionManager` – для работы с JDBC.
- С помощью аннотаций можно использовать разные менеджеры транзакций.

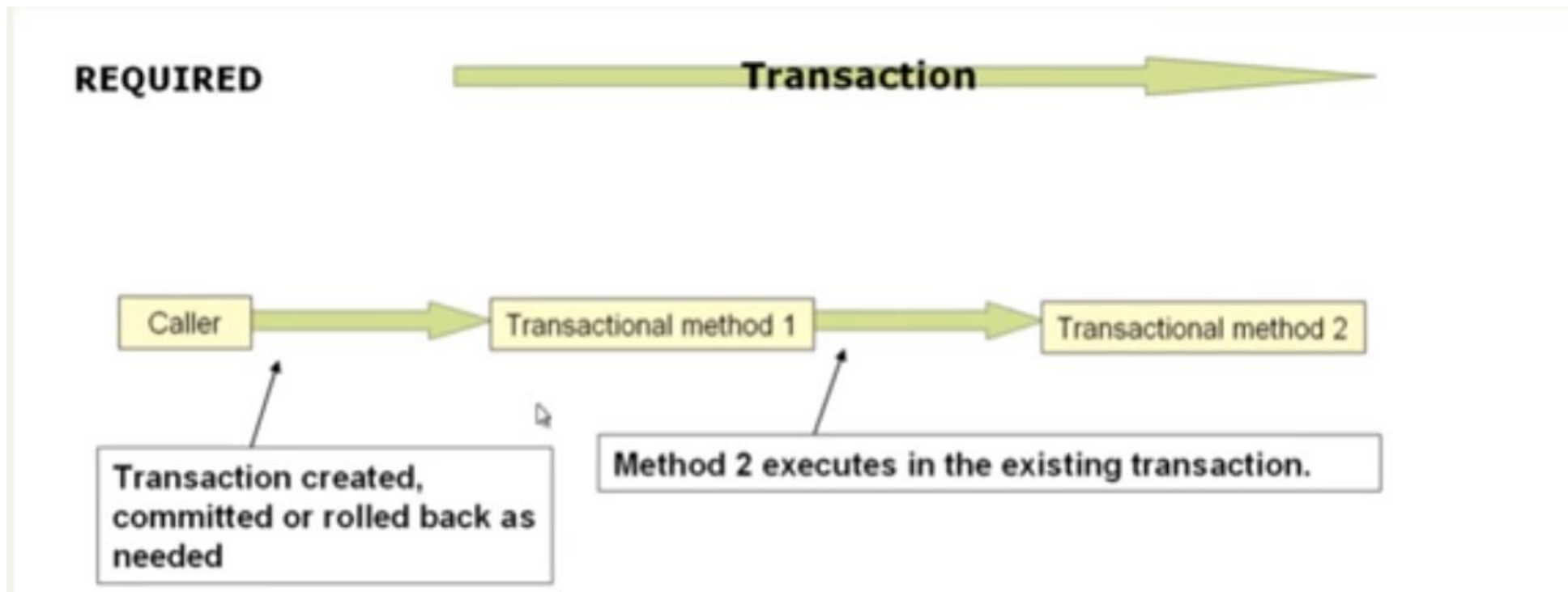
Transactions

Типы транзакций

- **Physical** – единая транзакция, которая действует сразу для нескольких методов.
- **Logical** – транзакция каждого метода.

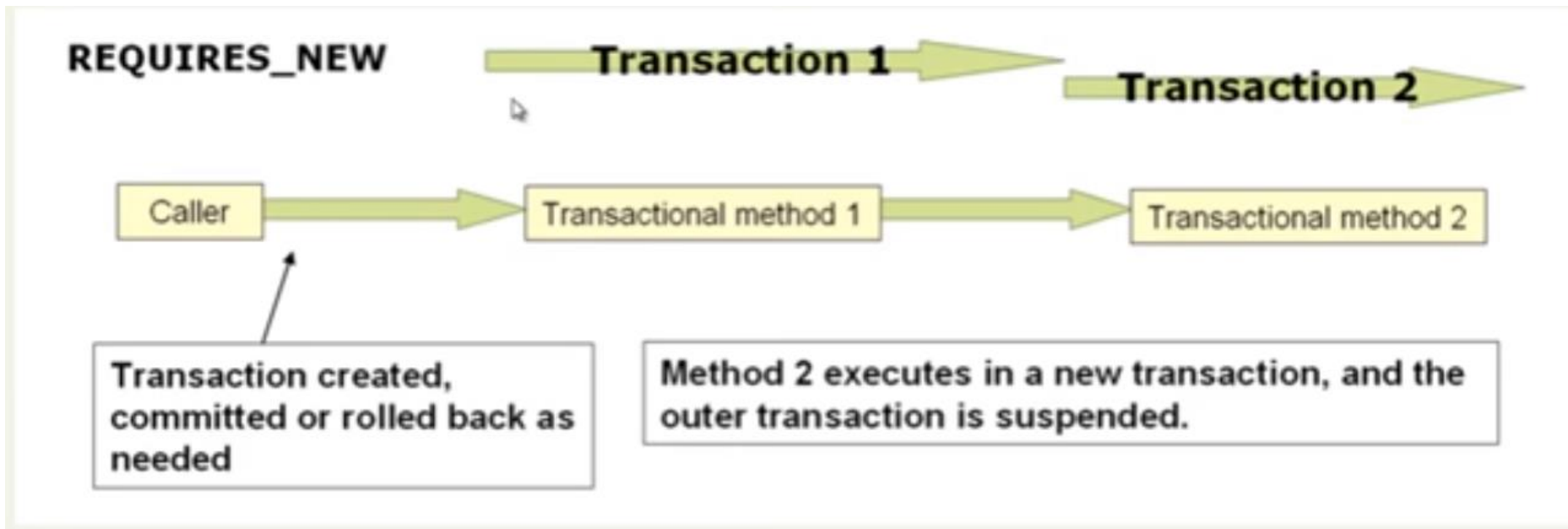
Transactions

Транзакции



Transactions

Транзакции



Transactions

Propagation

- Очень важный атрибут транзакций – от его правильного значения может зависеть вся логика работы приложения.
- Не оставлять значение по-умолчанию.
- Нужно следить, повлияет ли ошибка во внутренней транзакции на внешнюю.

Transactions

Свойство Propagation

- **MANDATORY** – использовать существующую транзакцию, если ее нет – выбрасывать исключение.
- **NESTED** – вложенная транзакция (подтранзакция), подтверждается вместе с внешней транзакцией. Если нет существующей транзакции – работает как **REQUIRED**.
- **NEVER** – не выполнять в текущей транзакции, если транзакция запущена – выбрасывать исключение.
- **NOT_SUPPORTED** – не выполнять в текущей транзакции, если транзакция запущена – она останавливается на время выполнения метода.
- **REQUIRED** – если запущена транзакция – выполнять внутри нее, иначе создает новую транзакцию.
- **REQUIRED_NEW** – создает новую транзакцию, если запущена существующая – она останавливается на время выполнения метода.
- **SUPPORTS** – может выполняться внутри транзакция, если она запущена, иначе выполнять без транзакции.

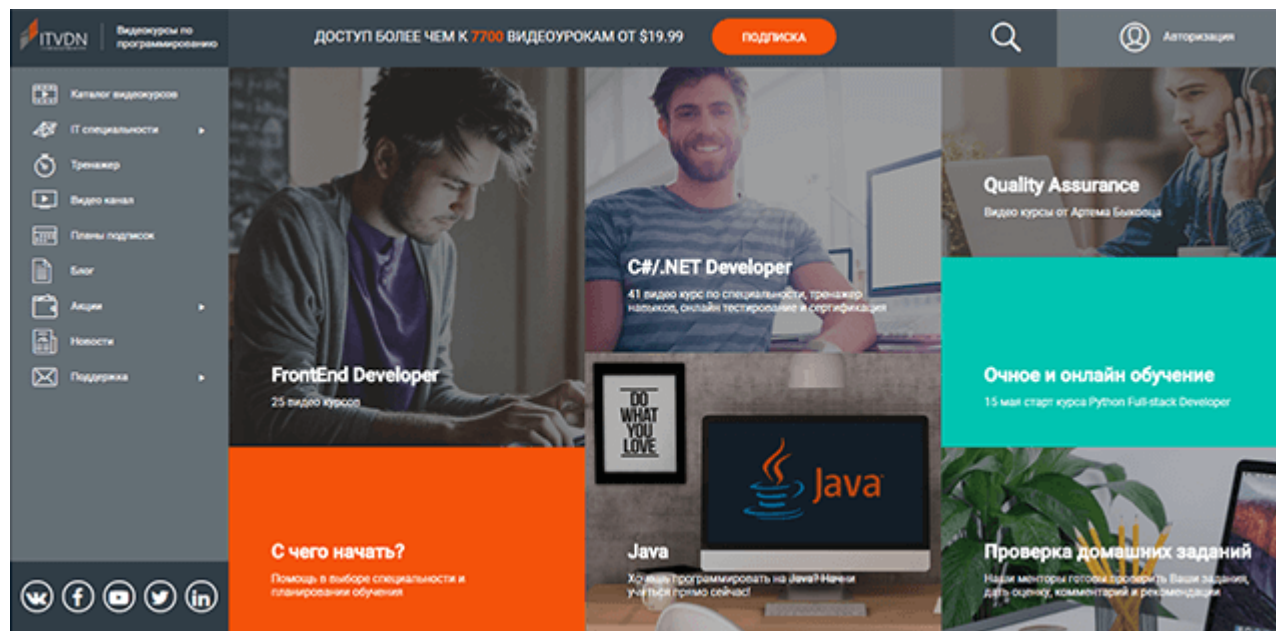
Transactions

Свойство Isolation

- **Default** – не рекомендуется, используется значение по-умолчанию для текущей базы данных.
- **READ_UNCOMMITTED** – читает неподтвержденные данные, самый быстрый, но небезопасный уровень, возможны все проблемы.
- **READ_COMMITTED** – чтение только подтвержденных данных, проблемы неповторяемых и фантомных строк сохраняется.
- **REPEATABLE_READ** – решает проблему неподтвержденных и неповторяемых строк, проблема фантомных строк остается.
- **SERIALIZABLE** – самый строгий уровень, все транзакции выполняются последовательно.

Смотрите наши уроки в видео формате

ITVDN.com



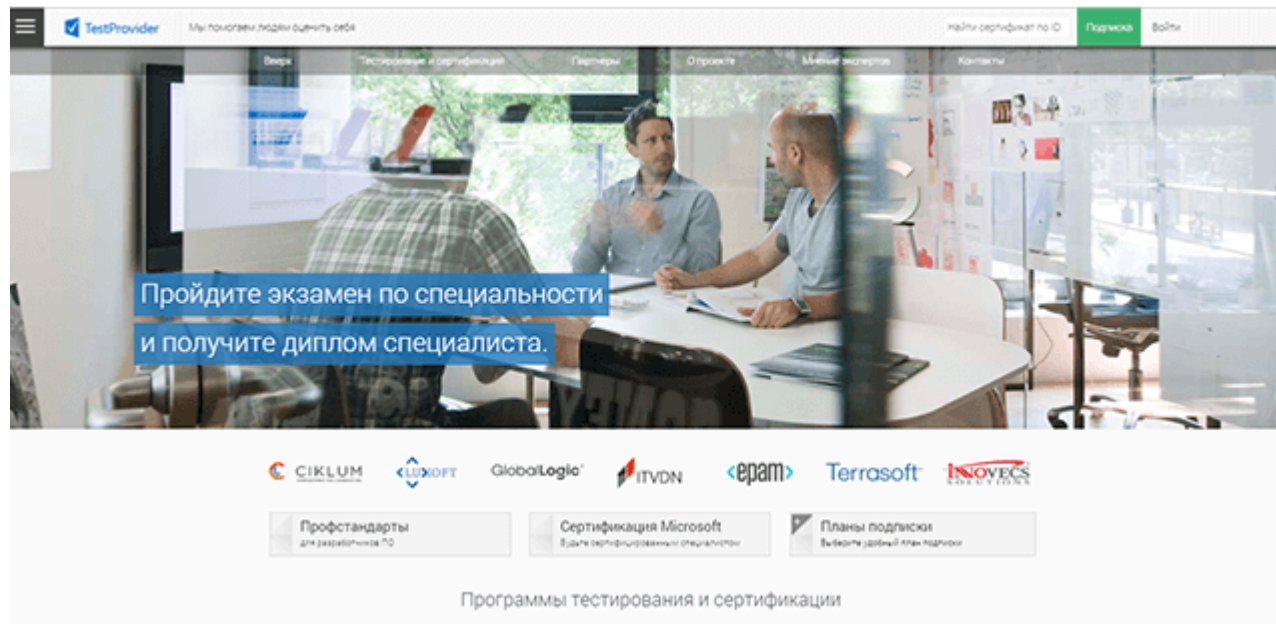
Посмотрите этот урок в видео формате на образовательном портале [ITVDN.com](http://itvdn.com) для закрепления пройденного материала.

Курсы записаны сертифицированными тренерами, которые работают в учебном центре CyberBionic Systematics и другими высококвалифицированными разработчиками.



Проверка знаний

TestProvider.com



TestProvider – это online сервис проверки знаний по информационным технологиям. С его помощью Вы можете оценить Ваш уровень и выявить слабые места. Он будет полезен как в процессе изучения технологии, так и для общей оценки знаний IT специалиста.

После каждого урока проходите тестирование для проверки знаний на [TestProvider.com](https://testprovider.com)

Успешное прохождение финального тестирования позволит Вам получить соответствующий Сертификат.



Q&A

Информационный видеосервис для разработчиков программного обеспечения

