Основы Java EE

Spring авто связывание. АОП

Автор курса

Бондаренко Андрей

После урока обязательно

Повторите этот урок в видео формате на ITVDN.com

Проверьте как Вы усвоили данный материал на TestProvider.com

Тема

Spring авто связывание. АОП

Basic Bean Configuration

- Все объекты должны быть загружены через контейнер (поэтому желательно не создавать вручную через new).
- Все объекты должны быть созданы с помощью bean.
- Отличие bean от объекта Java.
- Контейнер – ApplicationContext.
- Обращаемся к контейнеру из кода (но можем работать без контейнера вообще).

Autowiring

- Цель – сокращения кофигурации XML.
- Автоматический поиск бина внутри контейнера для внедрения.
- Можно использовать в XML через аннотации (более предпочтительно).
- В XML автоматическое связывание нужно использовать осторожно (или вообще не использовать.
- Лучше через аннотацию @Autowired (больше возможностей).
- Минусы: читаемость; путаница, какой объект будет использоваться.

Autowiring

- Атрибут autowire не наследует дочерние бины.
- Можно исключить бины, чтобы их не использовали при автоматическом поиске.
- Явное указание бинов в XML переопределяет бины поиска.

Annotations

Аннотации

- **@Required** – обезательное заполнение на этапе конфигурации.
- **@Autowired** – похож на атрибут autowire в XML, но имеет возможность уточнять поиск бинов для внедрения.
- **@Qualifier** – для утонения автоматического связывания.

Annotations

Аннотации

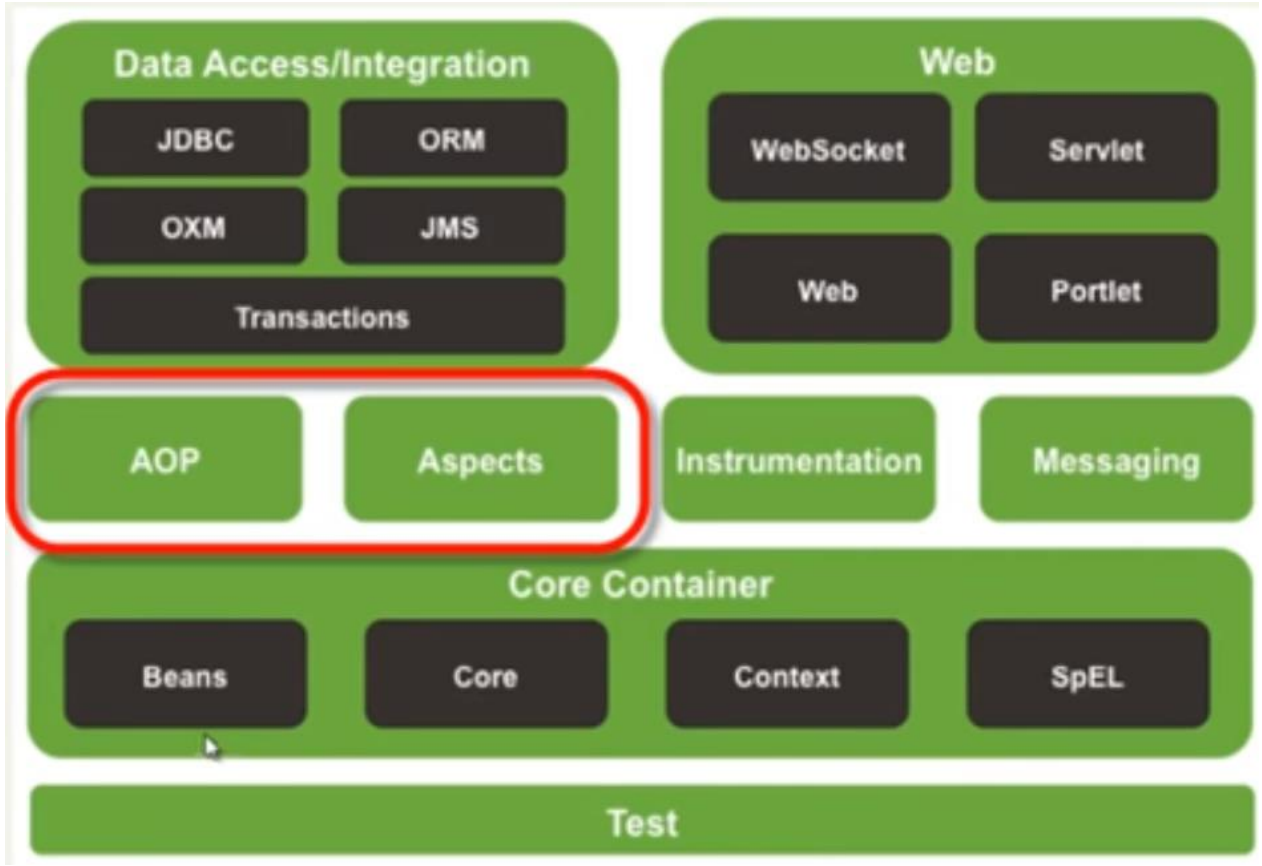
- Автоматически создаем компоненты на основе @Component или фильтра.
- Создаем нужное количество бинов с параметрами.
- @Autowired также работает.
- Меньше наглядности, нужно больше искать по классам.

Аннотации

- @Bean – аналог тега bean, возможность создания различных бинов на основе класса.
- @Bean рекомендуется использовать вместе с @Configuaration, а не с @Component.
- Практически все конфигурации XML можно повторить с помощью аннотаций.
- Полностью без XML обойтись не получится, так как надо объявить <context:component-scan>.
- <context:component-scan> - более расширенная версия <context:annotation-config>, делает тоже самое, но имеет больше возможностей.

Aspect Oriented Programming

AOP



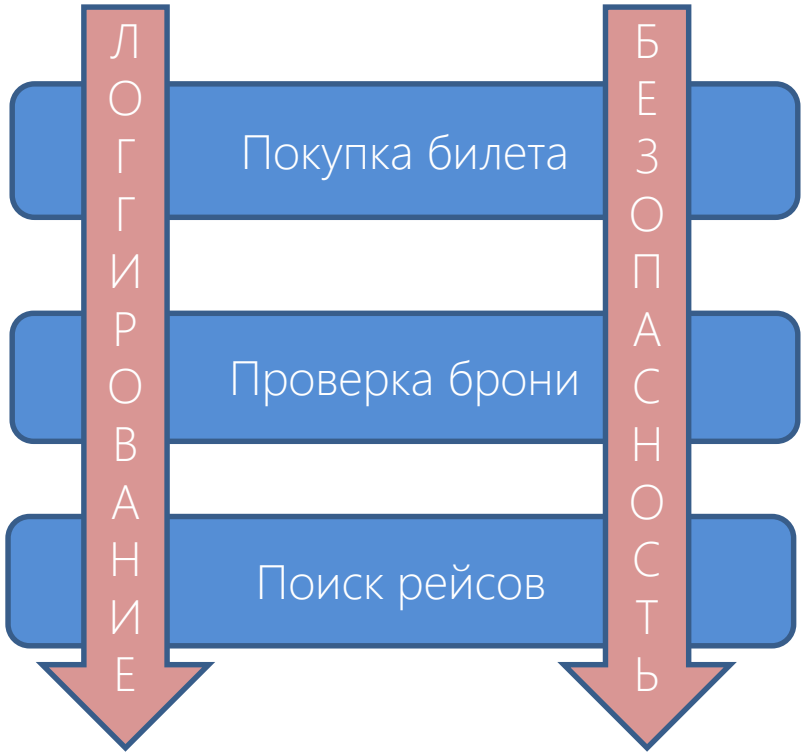
Aspect Oriented Programming

AOP

- Цель – разделения основного функционала от второстепенного.
- Центральное понятие - «аспект».
- «Сквозной» функционал (логирование, транзакции, безопасность).
- Внедрение «сквозного» функционала в основной.

Aspect Oriented Programming

AOP



AOP

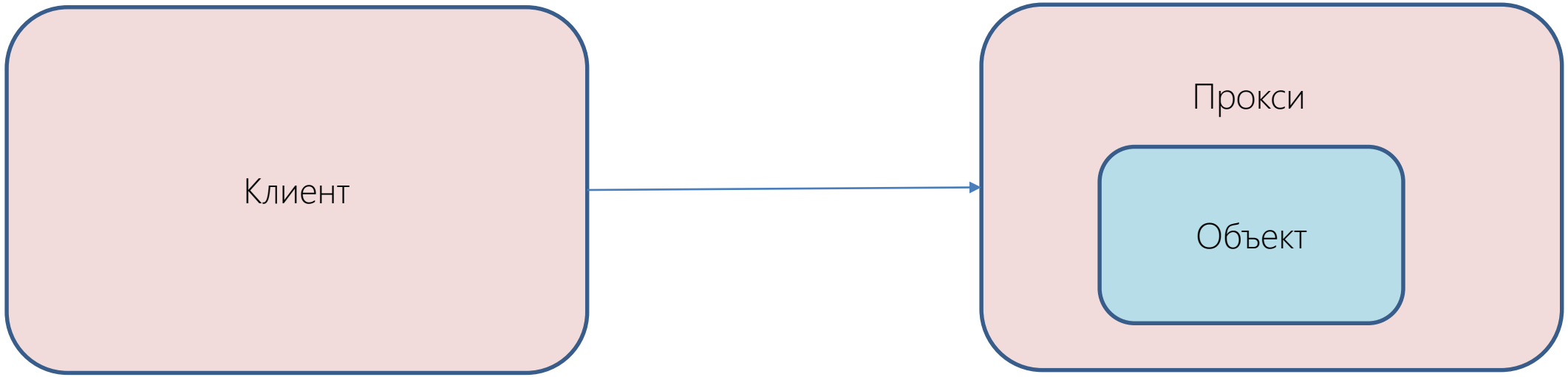
- Aspect – «сквозной» функционал.
- Advice – «совет», когда и что именно (какой код) нужно выполнить в точке соединения.
- Join Point – «Точка сопряжения», «Точка соединения» – в каком месте (при каком действии) нужно выполнить совет.
- Pointcut – «срез точек соединения» - область применения (поиск, сужение) точек соединения.

Аспект – набор советов для каждого из которых определены точки соединения и область применения.

Типы советов

- Before – до выполнения точки сопряжения.
- After – после выполнения (даже в случае ошибки).
- After returning – в случае успешного выполнения.
- After throwing – в случае возникновения exception.
- Around advice – до и после.

Proxy



Aspect-oriented programming

AOP

- Разделение основного функционала и дополнительного – без перемешивания их между собой.
- Кеширование, логирование, транзакции, безопасность и пр.
- Аспект – функциональность не относящаяся на прямую к бизнес логике.
- Аспекты можно использовать в любых проектах.

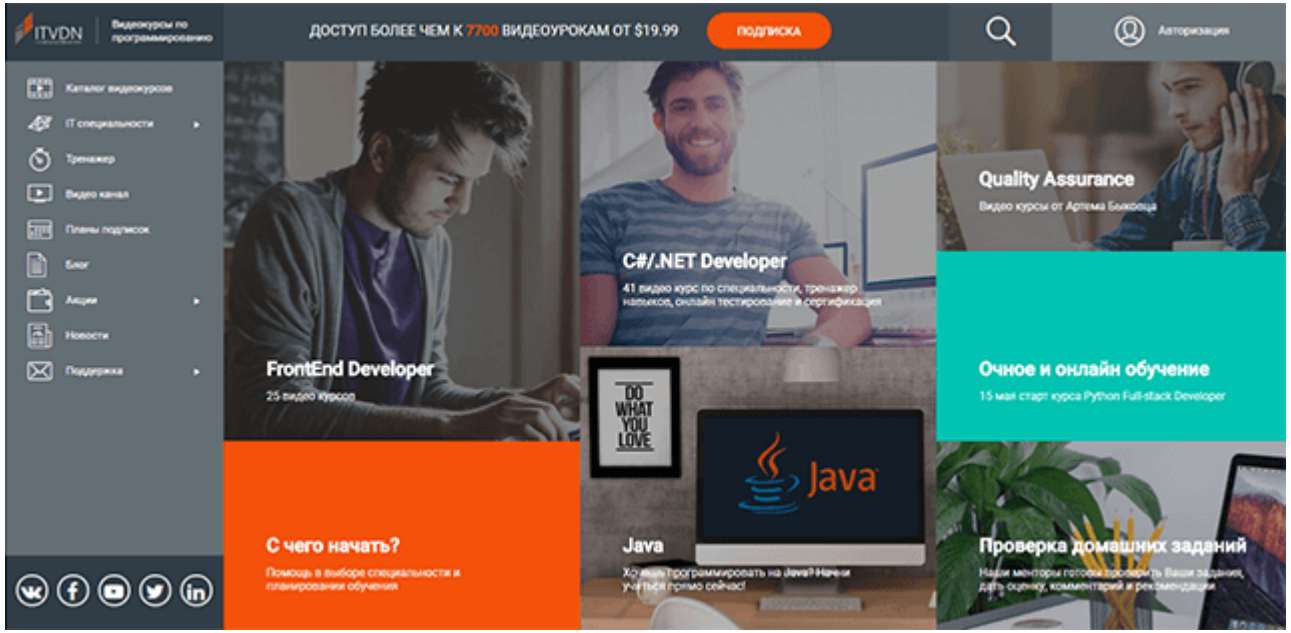
Aspect Oriented Programming

AOP

- В Spring Framework AOP точка сопряжения всегда равна «execution».
- AOP можно использовать и без Spring (в обычных Java программах).

Смотрите наши уроки в видео формате

ITVDN.com



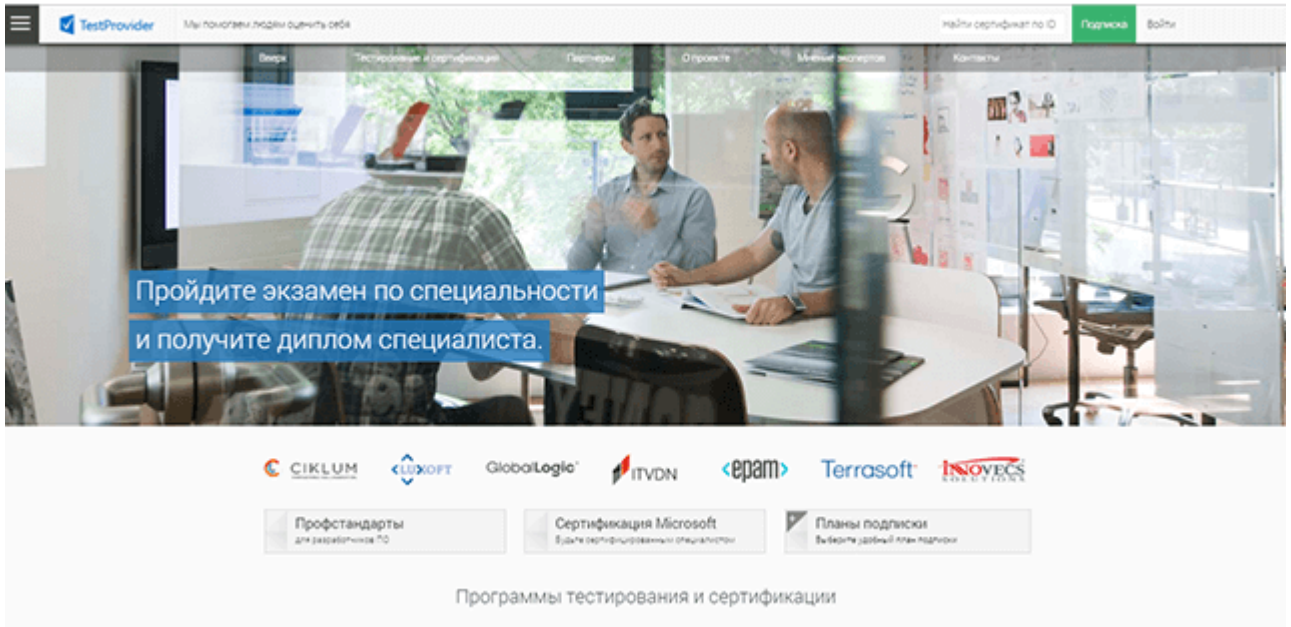
Посмотрите этот урок в видео формате на образовательном портале ITVDN.com для закрепления пройденного материала.

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

ITVDN
IT VIDEO DEVELOPERS NETWORK

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

Успешное прохождение финального тестирования позволит Вам получить соответствующий Сертификат.

TestProvider

Q&A

Информационный видеосервис для разработчиков программного обеспечения

ITVDN

IT VIDEO DEVELOPERS NETWORK

TestProvider PROMETRIC TEST CENTER CyberBionic systematics Microsoft Partner Silver Learning Windows Azure Cloud Partner EBA European Business Association