1. Перечислите Spring модули и их назначение

* Spring Boot – запуск приложения с минимальными настройками.
* Spring Data – модуль обеспечивает доступ к данным через реляционные и нереляционные базы данных. Содержит множество проектов, для разных СУБД. Механизм в Spring Data – репозиторий, который используется в JPA Entity для взаимодействия.
* Spring Cloud – позволяет быстро создавать шаблоны в распределенных системах (управление конфигурацией, обнаружение сервисов, интеллектуальная маршрутизация, микропрокси, одноразовые токены)
* Spring Cloud Data Flow – используется потоковая передача и обработка данных.
* Spring Security – базовые функции для безопасности, такие как: аутентификация, авторизация и контроль доступа.
* Spring Integration – позволяет облегчить обмен данными из различных источников.
* Spring Batch – платформа для разработки пакетных приложений, спользуется для ведения логов, трассировка, управление транзакциями, статистика обработки заданий.

1. Расскажите о составе Spring Framework.

Состоит из трех основных контейнеров Data Access/Integration (JDBC, ORM, JMS), Web (WebSocket, Servlet, Web, Portlet) и Core Container (Context, Core, Beans) и таких как TEST (Test NG или Junit)

1. Что такое Spring Boot? В чем его преимущества и для чего он используется?

Позволяет быстро создать и конфигурировать приложение, упаковывать в исполняемый артефакт. Позволяет отказаться от конфигурации на xml и сконфигурировать полностью на аннотациях.

1. Для чего используется аннотация @SpringBootApplication?

Неявно определяет базовый пакет поиска для орпеделенных элементов.

1. Объясните принцип IoC (Inversion of Control)? Какие формы используются в Spring для внедрения?

Inversion of Control – (инверсия управления) принцип, когда создание и управление зависимоостями между, компонентами становятся внешними.

Компонент, который требует зависимостей, часто называют зависимым объектом или, в случае IoC, целевым объектом. Вообще говоря, инверсия управления может быть разделена на два подтипа: внедрение зависимостей (Dependency Injection) и поиск зависимостей (Dependency Lookup)

1. В чем суть понятия lnversion of Control (IoC)?

Inversion of Control – (инверсия управления) принцип, когда создание и управление зависимоостями между, компонентами становятся внешними.

1. В чем различие внедрение зависимостей (Dependency lnjection) и поиска зависимостей (Dependency Lookup)?

Основным отличием является том, кто отвечает за извлечение зависимостей, в Dependency Injection, контейнер ничего не знает о контейнере DI и зависимости автоматически «впрыскиваются» при объявлении конструктура или с помощью анотаций, а в Dependency Lookup нужно указать конкретно, что нужно, то есть указать конкретный Bean и извлечь оттуда параметры.

DL:

ApplicationContext applicationContext = new ClassPathXmlApplicationContext("/application-context.xml");

MyBean bean = applicationContext.getBean("myBean")

DI:

<bean id="myBean" class="org.mypackage.MyBean"/>

@Component

public class MyClass{

@Autowired

MyBean myBean;

// ...

}

1. Что такое JavaBean? Какие есть правила описания и использования?

Классы в Java написанные по определенным правилам.

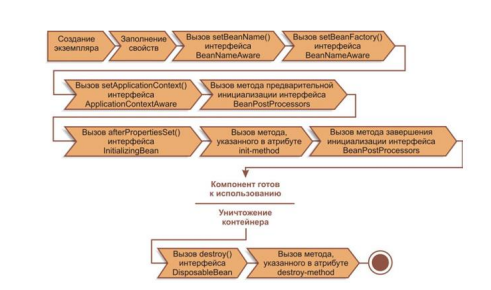
Такой класс должен:

* Иметь конструктор без параметров
* Свойства должны быть доступны через get, set и другие методы, которые должны быть написаны по определенным правилам именования.
* Класс должен быть сериаизуемым.
* Должен иметь переопределенные методы equals(), hashCode() и toString().

1. Перечислите области видимости bean.

* Singleton- (по умолчанию) создает единственный экземпляр бина, используется для бинов без сохранения состояния.
* Prototype – создает любое количество экземпляров бина, создается каждый раз при внедрении зависимостей или через getBean, используется для бинов с сохранением состояяния.
* Request – ограничен одним запросом HTTP
* Session – ограничен в пределах одной сессии
* Global Session – ограничен в пределах одной глобальной сессии
* Application – ограничен в пределах в пределах ServletContext

1. Опишите ЖЦ бина.



1. Поясните значения аннотаций: @Configuration, @Bean, @Component, @Service, @Repository, @Controller

* @Configuration – указывает класс конфигурации Spring Boot
* @Bean – используется для указания того, что метод создает объект, управляемый Spring IoC контейнером.
* @Component – создает автоматически экземпляры и внедрение зависимостей
* @Service – (производный от компонент) классы на уровне бизнес логики
* @Repository – (производный от компонент) аннотируемые классы на уровне сохроняемости, которорый работает с базой данных
* @Controller - – (производный от компонент) повзволяет автоматически определять классы реализации с помощью сканирования пути к классам.

1. Spring Expression Language (SpEL): расскажите об особенности и области использования.

<https://habr.com/ru/company/dsec/blog/433034/>

1. Охарактеризуйте основные Core Container Spring.

* Core – cсодержит базовые классы включая IoC и DI
* Bean – объекты bean содержит объекты bean
* Application Context – контейнер является подмножеством Bean
* SPeL ^^^

1. Как в Spring происходит разрешение зависимостей?

1) Создается и инициализируется ApplicationContext с метаданными

конфигурации, которые описывают все бины. Эти метаданные могут

быть описаны через XML, Java-код или аннотации

2) Для каждого бина и его зависимостей вычисляются свойства,

аргументы конструктора или аргументы статического фабричного

метода, либо обычного(без аргументов) конструктора. Эти

зависимости предоставляются бину, когда он(бин) уже создан. Сами

зависимости инициализируются рекурсивно, в зависимости от

вложенности в себе других бинов. Например, при инициализации

бина А, который имеет зависимость В, а В зависит от С, сначала

инициализируется бин С, потом В, а уже потом А

3) Каждому свойству или аргументу конструктора устанавливается

значение или ссылка на другой бин в контейнере

4) Для каждого свойства или аргумента конструктора подставляемое

значение конвертируется в тот формат, который указан для свойства

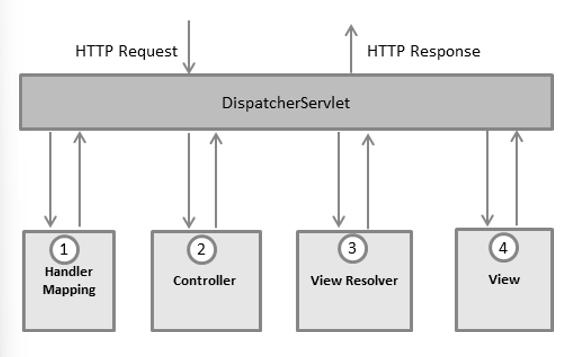
или аргумента. По умолчанию Spring может конвертировать значения

из строкового формата во все встроенные типы, такие

как int, long, String, boolean и др.

1. .Поясните как работает DispatcherServlet, HandlerMapping, ViewResolver? Как происходит обработка запроса?

* После получения HTTP-запроса DispatcherServlet обращается к интерфейсу **HandlerMapping**, который определяет, какой Контроллер должен быть вызван, после чего, отправляет запрос в нужный Контроллер.
* Контроллер принимает запрос и вызывает соответствующий служебный метод, основанный на GET или POST. Вызванный метод определяет данные Модели, основанные на определённой бизнес-логике и возвращает в DispatcherServlet имя Вида (View).
* При помощи интерфейса **ViewResolver** DispatcherServlet определяет, какой Вид нужно использовать на основании полученного имени.
* После того, как Вид (View) создан, DispatcherServlet отправляет данные Модели в виде атрибутов в Вид, который в конечном итоге отображается в браузере.



1. Как используется паттерн «Front Controller» в Spring?

Spring MVC-Front Controller

DispatcherServlet - это реализация шаблона проектирования фронтального контроллера. Он обеспечивает централизованную точку доступа для Spring Web MVC и отвечает за распределение обязанностей.

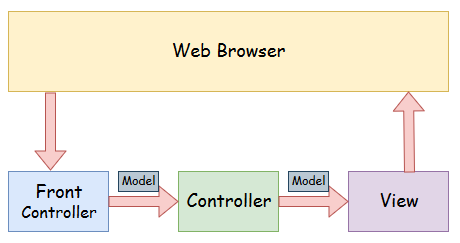
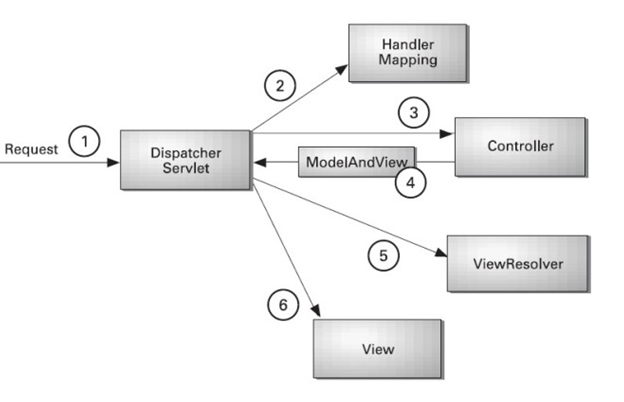
1. Как происходит адресация в контроллере?
2. Расскажите про Spring MVC архитектуру.

Front Controller – в Spring MVC реализованный на шаблоне фронтального контроллера представляет собой класс DispatcherServlet отвечает за перехват всех запросов и переадресация нужному контроллеру.

Controller – содержит бизнес-логику приложения

View – предостовляемая информация для отображении страницы

Model – данные приложения такие, как объекты или коллекция объектов

19. За что отвечает WebApplicationContext

В определении WebApplicationContext сказано, что это интерфейс, который предоставляет конфигурацию для веб приложения. Он является read-only (т.е. его нельзя перезаписать) во время выполнения приложения, но может быть перезагружен, если это позволяет реализация.

Здесь видно, что WebApplicationContext содержит в себе Controllers, HandlerMapping, ViewResolver. Всё это интерфейсы, которые отвечают за контроллеры, поиск представлений и обработку запросов.