SPRING

1. What is a framework?

A framework is a tool that provides ready components or solutions that are customized in order to speed up the development.

1. What is Spring?

Spring is a powerful open-source, application framework created to reduce the complexity of enterprise application development.

It is light-weighted and loosely coupled.

It has layered architecture, which allows you to select the components to use, while also providing a cohesive framework for J2EE application development.

Spring framework is also called the framework of frameworks as it provides support to various other frameworks such as Struts, Hibernate, Tapestry, EJB, JSF, etc.

1. Name the different modules of the Spring framework.

Some of the important Spring Framework modules are:

* Spring Context – for dependency injection.
* Spring AOP – for aspect-oriented programming.
* Spring DAO – for database operations using DAO pattern
* Spring JDBC – for JDBC and DataSource support.
* Spring ORM – for ORM tools support such as Hibernate
* Spring Web Module – for creating web applications.
* Spring MVC – Model-View-Controller implementation for creating web applications, web services, etc.

1. List some of the important annotations in annotation-based Spring configuration.

The important annotations are:

@Controller – for controller classes in the Spring MVC project.

@RequestMapping – for configuring URI mapping in controller handler methods. This is a very important annotation, so you should go through Spring MVC RequestMapping Annotation Examples

@ResponseBody – for sending Object as a response, usually for sending XML or JSON data as a response.

@PathVariable – for mapping dynamic values from the URI to handler method arguments.

@Autowired – for auto wiring dependencies in spring beans.

@Qualifier – with @Autowired annotation to avoid confusion when multiple instances of bean type are present.

@Service – for service classes.

@Scope – for configuring the scope of the spring bean.

@Configuration, @ComponentScan, and @Bean – for java based configurations.

AspectJ annotations for configuring aspects and advice, @Aspect, @Before, @After, @Around, @Pointcut, etc.

1. Explain Bean in Spring.

Beans are objects that form the backbone of a Spring application. They are managed by the Spring IoC container. In other words, a bean is an object that is instantiated, assembled, and managed by a Spring IoC container.

1. Different Scopes of Spring beans.

There are five Scopes defined in Spring beans:

* Singleton: Only one instance of the bean will be created for each container. This is the default scope for the spring beans. While using this scope, make sure spring bean doesn’t have shared instance variables otherwise it might lead to data inconsistency issues because it’s not thread-safe.
* Prototype: A new instance will be created every time the bean is requested.
* Request: This is the same as the prototype scope, however, it’s meant to be used for web applications. A new instance of the bean will be created for each HTTP request.
* Session: A new bean will be created for each HTTP session by the container.
* Global-session: This is used to create global session beans for Portlet applications.

1. DispatcherServlet

DispatcherServlet is the front controller in the Spring MVC application as it loads the Spring bean configuration file and initializes all the beans that have been configured. If annotations are enabled, it also scans the packages to configure any bean annotated with @Component, @Controller, @Repository, or @Service annotations.

1. ContextLoaderListener

ContextLoaderListener, on the other hand, is the listener to start up and shut down the WebApplicationContext in Spring root. Some of its important functions include tying up the lifecycle of Application Context to the lifecycle of the ServletContext and automating the creation of ApplicationContext.

1. Constructor injection

* No Partial Injection
* Doesn’t override the setter property
* Creates a new instance if any modification occurs
* Better for too many properties

1. Setter injection

* Partial Injection
* Overrides the constructor property if both are defined.
* Doesn’t create a new instance if you change the property value
* Better for a few properties.

1. What is auto wiring in Spring? What are the auto wiring modes?

Autowiring enables the programmer to inject the bean automatically. We don’t need to write explicit injection logic.

Autowiring modes:

* No - this is the default mode, it means auto wiring is not enabled.
* byName - Injects the bean based on the property name. It uses the setter method.
* byType - Injects the bean based on the property type. It uses the setter method.
* Constructor - It injects the bean using the constructor

1. What are the limitations with auto wiring?

Following are some of the limitations you might face with auto wiring:

* Overriding possibility: You can always specify dependencies using <constructor-arg> and <property> settings which will override auto wiring.
* Primitive data type: Simple properties such as primitives, Strings, and Classes can’t be autowired.
* Confusing nature: Always prefer using explicit wiring because auto wiring is less precise.

1. How to handle exceptions in Spring MVC Framework?

Spring MVC Framework provides the following ways to help us achieve robust exception handling.

* Controller-Based - We can define exception handler methods in our controller classes. All we need is to annotate these methods with @ExceptionHandler annotation.
* Global Exception Handler - Exception Handling is a cross-cutting concern and Spring provides @ControllerAdvice annotation that we can use with any class to define our global exception handler.
* HandlerExceptionResolver implementation - For generic exceptions, most of the time we serve static pages. Spring Framework provides a HandlerExceptionResolver interface that we can implement to create a global exception handler. The reason behind this additional way to define the global exception handler is that the Spring framework also provides default implementation classes that we can define in our spring bean configuration file to get spring framework exception handling benefits.

1. How to integrate Spring and Hibernate Frameworks?

We can use the Spring ORM module to integrate Spring and Hibernate frameworks if you are using Hibernate 3+ where SessionFactory provides the current session, then you should avoid using HibernateTemplate or HibernateDaoSupport classes, and better to use the DAO pattern with dependency injection for the integration.

Also, Spring ORM provides support for using Spring declarative transaction management, so you should utilize that rather than going for hibernating boiler-plate code for transaction management.

1. Name the types of transaction management that Spring supports.

Two types of transaction management are supported by Spring. They are:

* Programmatic transaction management: In this, the transaction is managed with the help of programming. It provides you extreme flexibility, but it is very difficult to maintain.
* Declarative transaction management: In this, transaction management is separated from the business code. Only annotations or XML-based configurations are used to manage the transactions.

1. What is a Spring configuration file?

The Spring configuration file is an XML file. This file mainly contains the class information. It describes how those classes are configured as well as introduced to each other. The XML configuration files, however, are verbose and cleaner. If it’s not planned and written correctly, it becomes very difficult to manage big projects.

1. What are the different components of a Spring application?

A Spring application generally consists of the following components:

* Interface: It defines the functions.
* Bean class: It contains properties, its setter and getter methods, functions, etc.
* Spring Aspect-Oriented Programming (AOP): Provides the functionality of cross-cutting concerns.
* Bean Configuration File: Contains the information on classes and how to configure them.
* User program: It uses the function.

1. What is an IoC container?

At the core of the Spring Framework, lies the Spring container. The container creates the object, wires them together, configures them, and manages their complete life cycle. The Spring container makes use of Dependency Injection to manage the components that make up an application. The container receives instructions for which objects to instantiate, configure, and assemble by reading the configuration metadata provided. This metadata can be provided either by XML, Java annotations, or Java code.

1. What do you mean by Dependency Injection?

In Dependency Injection, you do not have to create your objects but have to describe how they should be created. You don’t connect your components and services in the code directly, but describe which services are needed by which components in the configuration file. The IoC container will wire them up together.

1. List some of the benefits of IoC.

Some of the benefits of IoC are:

* It will minimize the amount of code in your application.
* It will make your application easy to test because it doesn’t require any singletons or JNDI lookup mechanisms in your unit test cases.
* It promotes loose coupling with minimal effort and the least intrusive mechanism.
* It supports eager instantiation and lazy loading of the services.

1. In how many ways can Dependency Injection be done?

In general, dependency injection can be done in three ways, namely:

* Constructor Injection
* Setter Injection
* Interface Injection

In Spring Framework, only constructor and setter injections are used.

1. How many types of IOC containers are there in spring?

* BeanFactory: BeanFactory is like a factory class that contains a collection of beans. It instantiates the bean whenever asked for by clients.
* ApplicationContext: The ApplicationContext interface is built on top of the BeanFactory interface. It provides some extra functionality on top BeanFactory.

1. Explain Spring Beans?

They are the objects that form the backbone of the user’s application.

Beans are managed by the Spring IoC container.

They are instantiated, configured, wired, and managed by a Spring IoC container

Beans are created with the configuration metadata that the users supply to the container.

1. How is configuration metadata provided to the Spring container?

Configuration metadata can be provided to the Spring container in the following ways:

* XML-Based configuration: In Spring Framework, the dependencies and the services needed by beans are specified in configuration files which are in XML format. These configuration files usually contain a lot of bean definitions and application-specific configuration options. They generally start with a bean bag.
* Annotation-Based configuration: Instead of using XML to describe a bean wiring, you can configure the bean into the component class itself by using annotations on the relevant class, method, or field declaration. By default, annotation wiring is not turned on in the Spring container. So, you need to enable it in your Spring configuration file before using it.
* Java-based configuration: The key features in Spring Framework’s new Java-configuration support are @Configuration annotated classes and @Bean annotated methods.

1. Define Bean Wiring.

When beans are combined within the Spring container, it’s called wiring or bean wiring. The Spring container needs to know what beans are needed and how the container should use dependency injection to tie the beans together, while wiring beans.

1. What do you mean by Annotation-based container configuration?

Instead of using XML to describe a bean wiring, the developer moves the configuration into the component class itself by using annotations on the relevant class, method, or field declaration. It acts as an alternative to XML setups.

1. What are the ways by which Hibernate can be accessed using Spring?

There are two ways by which we can access Hibernate using Spring:

* Inversion of Control with a Hibernate Template and Callback
* Extending HibernateDAOSupport and Applying an AOP Interceptor node

1. Describe Spring DAO support?

The Data Access Object (DAO) support in Spring makes it easy to work with data access technologies like JDBC, Hibernate, or JDO consistently. This allows one to switch between the persistence technologies easily. It also allows you to code without worrying about catching exceptions that are specific to each of these technologies.

1. Describe AOP.

Aspect-oriented programming or AOP is a programming technique that allows programmers to modularize crosscutting concerns or behavior that cuts across the typical divisions of responsibility. Examples of cross-cutting concerns can be logging and transaction management. The core of AOP is an aspect. It encapsulates behaviors that can affect multiple classes into reusable modules.

1. What do you mean by Proxy in Spring Framework?

An object which is created after applying advice to a target object is known as a Proxy. In the case of client objects, the target object and the proxy object are the same.

1. What do you mean by the Spring MVC framework?

The Spring web MVC framework provides model-view-controller architecture and ready-to-use components that are used to develop flexible and loosely coupled web applications. The MVC pattern helps in separating the different aspects of the application like input logic, business logic, and UI logic while providing a loose coupling between all these elements.

1. Describe DispatcherServlet.

The DispatcherServlet is the core of the Spring Web MVC framework. It handles all the HTTP requests and responses. The DispatcherServlet receives the entry of handler mapping from the configuration file and forwards the request to the controller. The controller then returns an object of Model And View. The DispatcherServlet checks the entry of the view resolver in the configuration file and calls the specified view component.

1. In the Spring MVC framework, what is a controller?

Controllers provide access to the application behavior. These behaviors are generally defined through a service interface. Controllers interpret the user input and transform it into a model which is represented to the user by the view. In Spring, the controller is implemented in a very abstract way. It also enables you to create a wide variety of controllers.

ПРОЛЕТ

1. Какво представлява рамката?

Рамка е инструмент, който предоставя готови компоненти или решения, които са персонализирани с цел ускоряване на развитието.

2. Какво представлява Пролетта?

Пролетта е мощна рамка с отворен код, рамка за кандидатстване, създадена за намаляване на сложността на развитието на корпоративното приложение.

Тя е леко претеглена и хлабаво съчетана.

Разполага със слоеста архитектура, която ви позволява да изберете компонентите, които да използвате, като същевременно осигурява и сплотена рамка за разработване на приложения на J2EE.

Пролетната рамка също се нарича рамка на рамки, тъй като предоставя подкрепа за различни други рамки като Подпори, Хибернат, Гоблен, EJB, JSF и др.

3. Назовете различните модули на рамката Пролет.

Някои от важните пролетни рамкови модули са:

        Пролетен контекст – за зависимост инжектиране.

        Spring AOP – за аспектно ориентирано програмиране.

        Пролет DAO – за операции на база данни с помощта на DAO модел

        Пролет JDBC – за JDBC и DataSource поддръжка.

        Пролетна ORM – за ORM инструменти поддръжка като Хибернат

        Пролетен уеб модул – за създаване на уеб приложения.

        Spring MVC – Model-View-Controller внедряване за създаване на уеб приложения, уеб услуги и др.

4. Избройте някои от важните анотации в конфигурацията Spring, базирана на анотация.

Важните анотации са:

@Controller – за класове контролери в проекта Spring MVC.

@RequestMapping – за конфигуриране на съпоставяне на URI в методите на манипулатора на контролери. Това е много важна анотация, така че трябва да преминете през Пролет MVC RequestMapping Анотация Примери

@ResponseBody – за изпращане на Обект като отговор, обикновено за изпращане на XML или JSON данни като отговор.

@PathVariable – за съпоставяне на динамични стойности от URI на манипулатор метод аргументи.

@Autowired – за зависимости от автоматично окабеляване в пролетния боб.

@Qualifier – с @Autowired анотация, за да се избегне объркване, когато са налице множество екземпляри от тип боб.

@Service – за сервизни класове.

@Scope – за конфигуриране на обхвата на пролетния боб.

@Configuration, @ComponentScan и @Bean – за конфигурации, базирани на Java.

AspectJ анотации за конфигуриране на аспекти и съвети, @Aspect, @Before, @After, @Around, @Pointcut и др.

5. Обяснете Бийн през пролетта.

Фасулът са обекти, които образуват гръбнака на приложение Spring. Те се управляват от контейнера Spring IoC. С други думи, боб е обект, който е моментиран, сглобен и управляван от spring IoC контейнер.

6. Различни Обхвати на пролетния боб.

Има пет Обхвати, дефинирани в Пролетен боб:

        Singleton: Само един екземпляр на боба ще бъде създаден за всеки контейнер. Това е обхватът по подразбиране за пролетния боб. Докато използвате този обхват, уверете се, че пролетният боб няма споделени екземпляр променливи в противен случай това може да доведе до проблеми с несъответствието на данните, защото не е безопасно за конци.

        Прототип: Ще се създава нов екземпляр всеки път, когато бобът бъде поискан.

        Заявка: Това е същото като обхвата на прототипа, обаче, той е предназначен да се използва за уеб приложения. За всяка HTTP заявка ще бъде създаден нов екземпляр на боба.

        Сесия: Ще бъде създаден нов боб за всяка HTTP сесия от контейнера.

        Глобална-сесия: Това се използва за създаване на глобална сесия боб за Приложения Portlet.

7. ДиспечерСервлет

DispatcherServlet е предният контролер в приложението Spring MVC, тъй като зарежда конфигурационния файл Spring bean и инициализира всички боб, които са конфигурирани. Ако са разрешени анотации, той сканира и пакетите, за да конфигурира всеки боб, анотиран с @Component, @Controller, @Repository или @Service анотации.

8. ContextLoaderЛистенер

ContextLoaderListener, от друга страна, е слушателът да започне нагоре и изключване на WebApplicationContext в пролет корен. Някои от важните му функции включват завързяване на жизнения цикъл на Application Context към жизнения цикъл на ServletContext и автоматизиране на създаването на ApplicationContext.

9. Инжектиране на конструктор

        Без частично инжектиране

        Не премеждва свойството setter

        Създава нов екземпляр, ако възникне някаква модификация

        По-добре за твърде много имоти

10. Инжектиране на сетер

        Частично инжектиране

        Замества свойството на конструктора, ако и двете са дефинирани.

        Не създава нов екземпляр, ако промените стойността на свойството

        По-добре за няколко имота.

11. Какво представлява автоматичното окабеляване през Пролетта? Какви са режимите на автоматично окабеляване?

Автоматичното включване дава възможност на програмиста да инжектира боба автоматично. Няма нужда да пишем изрична логика на инжектиране.

Режими на автоматично окабеляване:

        Не - това е режимът по подразбиране, това означава, че автоматичното окабеляване не е разрешено.

        byName - Инжектира боба въз основа на името на свойството. Той използва метода на сетер.

        byType - Инжектира боба въз основа на типа свойство. Той използва метода на сетер.

        Конструктор - Той инжектира боба с помощта на конструктора

12. Какви са ограниченията при автоматичното окабеляване?

Следват някои от ограниченията, с които може да се сблъскате с автоматичното окабеляване:

        Импераваща възможност: Винаги можете да посочите зависимости, като използвате <конструктор-арг> и <свойство> настройки, които ще заместят автоматичното окабеляване.

        Тип примитивни данни: Прости свойства като примитиви, низове и класове не могат да бъдат автоматично въртетени.

        Объркваща природа: Винаги предпочитайте да използвате изрично окабеляване, защото автоматичното окабеляване е по-малко прецизно.

13. Как да се справяте с изключенията в Spring MVC Framework?

Spring MVC Framework предоставя следните начини, които да ни помогнат да постигнем солидна обработка на изключенията.

        Базирани на контролери - Ние можем да определим изключение манипулатор методи в нашите класове контролер. Трябва ни само да анотираме тези методи с @ExceptionHandler анотация.

        Global Exception Handler - Exception Handling е междусекторна грижа и Spring предоставя @ControllerAdvice анотация, която можем да използваме с всеки клас, за да дефинираме нашия глобален манипулатор на изключения.

        HandlerExceptionResolver внедряване - За генерични изключения, през повечето време обслужваме статични страници. Spring Framework предоставя HandlerExceptionResolver интерфейс, който можем да внедрим, за да създадем глобален манипулатор на изключения. Причината зад този допълнителен начин за определяне на глобалния манипулатор на изключения е, че рамката Spring също осигурява класове по изпълнение по подразбиране, които можем да определим в нашия пролетен боб конфигурационен файл, за да получите пролетна рамка изключение обработка ползи.

14. Как да се интегрират пролетни и хибернаторски рамки?

Можем да използваме пролетния ORM модул за интегриране на spring и Hibernate рамки, ако използвате Hibernate 3+ където SessionFactory осигурява текущата сесия, тогава трябва да избягвате да използвате HibernateTemplate или HibernateDaoSupport класове, и по-добре да използвате DAO модела с зависимост инжекция за интеграцията.

Също така, Spring ORM предоставя поддръжка за използване на Spring декларативно управление на транзакции, така че трябва да се използва, че вместо да става за хибернация котел-плоча код за управление на транзакции.

15. Назовете видовете управление на транзакциите, които Spring поддържа.

Два вида управление на транзакциите се поддържат от Spring. Те са:

        Програмно управление на транзакции: В това транзакцията се управлява с помощта на програмиране. Тя ви осигурява изключителна гъвкавост, но е много трудно да се поддържа.

        Управление на декларативни транзакции: В това управлението на транзакциите е отделено от бизнес кода. Само анотации или XML-базирани конфигурации се използват за управление на транзакциите.

16. Какво представлява конфигурационният файл Spring?

Пружинният конфигурационен файл е XML файл. Този файл съдържа главно информацията за класа. Тя описва как тези класове са конфигурирани, както и въведени един към друг. XML конфигурационните файлове обаче са глаголи и по-чисти. Ако не е планирано и написано правилно, става много трудно да се управляват големи проекти.

17. Какви са различните компоненти на приложение Spring?

Приложение Spring като цяло се състои от следните компоненти:

        Интерфейс: Той определя функциите.

        Клас боб: Той съдържа свойства, неговия сетер и getter методи, функции и т.н.

        Пролетно ориентирано към аспекта програмиране (AOP): Осигурява функционалността на междусекторните опасения.

        Файл за конфигуриране на боб: съдържа информацията за класовете и как да ги конфигурирате.

        Потребителска програма: Тя използва функцията.

18. Какво представлява контейнер IoC?

В сърцевината на Пролетната рамка, се намира контейнерът Пролет. Контейнерът създава обекта, жици ги заедно, конфигурира ги и управлява пълния им жизнен цикъл. Контейнерът Spring използва Dependency Injection за управление на компонентите, които съставят приложение. Контейнерът получава инструкции, за които обектите да моментират, конфигурират и сглобяват, като прочетат предоставените конфигурационни метаданни. Тези метаданни могат да бъдат предоставени или по XML, Java анотации, или Код на Java.

19. Какво имате предвид под Инжектиране на зависимости?

В Dependency Injection не трябва да създавате вашите обекти, а трябва да опишете как трябва да бъдат създадени. Не свързвате директно компонентите и услугите си в кода, а описвате кои услуги са необходими, с кои компоненти в конфигурационния файл. Контейнерът на IoC ще ги жицира заедно.

20. Избройте някои от ползите от IoC.

Някои от ползите от IoC са:

        Тя ще сведе до минимум количеството код във вашата молба.

        Тя ще направи вашето приложение лесно да се тества, защото не изисква никакви singletons или JNDI справочна механизми във вашите единица тест случаи.

        Той насърчава хлабав съкачване с минимални усилия и най-малко натрапчив механизъм.

        Поддържа нетърпеливи инстанция и мързеливо зареждане на услугите.

21. По колко начина може да се направи Dependency Injection?

Като цяло, зависимост инжектиране може да се направи по три начина, а именно:

        Конструктор Инжекция

        Сетер Инжектиране

        Интерфейс Инжектиране

В Spring Framework се използват само инжекции на конструктор и сетер.

22. Колко вида контейнери на IOC има през пролетта?

        BeanFactory: BeanFactory е като фабричен клас, който съдържа колекция от боб. Той моментира боба винаги, когато бъде поискан от клиенти.

        ПриложениеКонтекст: Интерфейсът на ApplicationContext е изграден на върха на BeanFactory интерфейс. Той осигурява известна допълнителна функционалност на върха BeanFactory.

23. Обяснете Пролетен боб?

Те са обектите, които образуват гръбнака на приложението на потребителя.

Фасулът се управлява от контейнера Spring IoC.

Те са моментирани, конфигурирани, кабелна и управлявани от spring IoC контейнер

Фасулът се създава с метаданните на конфигурацията, които потребителите доставят на контейнера.

24. Как се предоставят конфигурационните метаданни на контейнера Spring?

Конфигурационните метаданни могат да бъдат предоставени на контейнера Spring по следните начини:

        XML-Базирана конфигурация: В Пролетна рамка зависимостите и услугите, необходими на боба, са посочени в конфигурационните файлове, които са във формат XML. Тези конфигурационни файлове обикновено съдържат много боб дефиниции и специфични за приложението опции за конфигурация. Те по принцип започват с торбичка с боб.

        Конфигурация, базирана на анотация: Вместо да използвате XML, за да опишете окабеляване на боб, можете да конфигурирате боба в самия клас компоненти, като използвате анотации на съответната декларация за клас, метод или поле. По подразбиране анотационната окабеляване не е включена в контейнера Spring. Така че, трябва да го активирате във вашия конфигурационен файл Spring, преди да го използвате.

        Конфигурация, базирана на Java: Ключовите функции в новата поддръжка на Java-configuration на Spring Framework са @Configuration анотирани класове и @Bean анотирани методи.

25. Дефиниране на Бийн Окабеляване.

Когато бобът се комбинира в рамките на контейнера Spring, той се нарича окабеляване или окабеляване на боб. Контейнерът Spring трябва да знае какви боб са необходими и как контейнерът трябва да използва зависимост инжекция за връзване на боба заедно, докато окабеляване на боб.

26. Какво имате предвид под конфигурация на контейнер, базиран на Annotation?

Вместо да използва XML, за да опише окабеляване на боб, разработчикът премества конфигурацията в самия клас компоненти, като използва анотации на съответната декларация за клас, метод или поле. Той действа като алтернатива на XML настройките.

27. Какви са начините, по които Хибернат може да бъде достъпен с помощта на Spring?

Има два начина, по които можем да осъществим достъп до Hibernate с помощта на Spring:

        Инверсия на контрола с хибернатен шаблон и обратно повикване

        Удължаване на хибернацияDAOSupport и прилагане на AOP прихващач възел

28. Опишете spring DAO поддръжка?

Поддръжката на обекта за достъп до данни (DAO) през Пролетта улеснява последователното работа с технологии за достъп до данни като JDBC, Hibernate или JDO. Това позволява на един да превключвате между технологиите за устойчивост лесно. Той също така ви позволява да код, без да се притеснявате за улов изключения, които са специфични за всяка от тези технологии.

29. Опишете AOP.

Ориентираното към аспекта програмиране или AOP е програмна техника, която позволява на програмистите да модулират опасенията или поведението, свързани с кръстосаните пътища, които разрязват типичните разделения на отговорността. Примери за междусекторни опасения могат да бъдат регистрирането и управлението на сделките. Ядрото на AOP е аспект. Той капсулира поведения, които могат да засегнат няколко класа в модули за многократнаупотребима.

30. Какво имате предвид под Прокси в Пролетна рамка?

Обект, който се създава след прилагане на съвети към целеви обект, е известен като Прокси. В случай на клиентски обекти целевият обект и прокси обектът са еднакви.

31. Какво имате предвид с рамката spring MVC?

Рамката spring web MVC предоставя архитектура на контролера за модел и готови за използване компоненти, които се използват за разработване на гъвкави и хлабаво обвързани уеб приложения. Моделът MVC помага при разделянето на различните аспекти на приложението като входната логика, бизнес логиката и логиката на ПИ, като същевременно осигурява хлабаво съединяване между всички тези елементи.

32. Опишете ДиспечерServlet.

ДиспечерътServlet е ядрото на рамката на Spring Web MVC. Тя обработва всички HTTP заявки и отговори. Диспечерservlet получава записа на манипулатор съпоставяне от конфигурационния файл и препраща искането към контролера. След това контролерът връща обект на "Модел и изглед". DispatcherServlet проверява записа на резолвера за изглед в конфигурационния файл и извиква посочения компонент на изгледа.

33. В рамката spring MVC какво представлява администраторът?

Контролерите предоставят достъп до поведението на приложението. Тези поведения обикновено се определят чрез интерфейс за услуги. Контролерите интерпретират потребителския вход и го трансформират в модел, който е представен на потребителя от изгледа. През Пролетта контролерът се изпълнява по много абстрактен начин. Той също така ви дава възможност да създадете голямо разнообразие от контролери.