МЕТОДОЛОГИЯ СОЗДАНИЯ ПРИЛОЖЕНИЙ ИСПОЛЬЗУЯ SPRING FRAMEWORK

METODOLOGIA ELABORĂRII APLICAȚIILOR FOLOSIND SPRING FRAMEWORK

Денис НЕГУРА, ORCID: 0000-0003-0618-5360
Татьяна ШЕСТАКОВА, ORCID: 0000-0003-1283-8524
Вадим ГОНЦА
Ольга ЧЕРБУ, ORCID: 0000-0002-6278-7115
Технический университет Молдовы,
Молдавский государственный университет

CZU: 004.438

e-mail: den.negura@gmail.com
e-mail: tatiana.sestacova@sde.utm.md
e-mail: vadimgonta1968@gmail.com
e-mail: olga.cerbu@gmail.com

Состав Spring Framework

Ядро каркаса Spring основано на *инверсии управления* (Inversion of control – IoC), IoC подразумевает делегирование программистом компонентов программы фреймворку, давая ему возможность вызывать код программиста при необходимости.[1] Инверсия управления в Spring часто замещается термином *внедрение зависимостей*, это связанно с тем, что Spring действует подобно контейнеру, предоставляя экземпляры классов со всеми необходимыми зависимостями.

Позволяя Spring внедрять зависимости, программист достигает гибкости в определении конфигурации зависимостей в приложениях такими средствами как: XML-файлы, конфигурационные классы Java, аннотации в коде. Внедрение зависимостей нацелено на ослабление связей между компонентами программы, что упрощает в дальнейшем поддержку модулей.

Beans – это компоненты java управляемые контейнером, размещённые в контексте приложения [2, 3]. Бины можно конфигурировать XML разметкой, по средством аннотаций.

Доступ к данным и работа с данными – основной функционал любой программы. Spring легко интегрируется с любым выбранным инструментом. Он так же поддерживает простой интерфейс JDBC (Java Database Connectivity). Кроме того, Spring поддерживает библиотеки на основе спецификаций JPA (технология ORM – объектно реляционное отображение) и JDO, самую популярную реализа-

цию Hibernate [4, 5]. Spring позволяет сочетать технологии доступа к базам данных. К примеру, использовать специфичные средства БД через JDBC, а основную логику строить на Hibernate.

Web MVC – проектный шаблон для создания веб-приложений включает три части:

1. Модель – данные и объекты для доступа к данным.

2. Представление – отображение данных в удобном для пользователя формате, стилизация, интернализация, и возможность взаимодействия на стороне клиента.

3. Контроллер – обработчик запросов пользователей. Взаимодействует с моделью, обновляет, сохраняет, читает данные и направляет пользователей к представлениям на основании действий пользователей.

Работа Spring MVC построена вокруг DispatcherServlet – центрального сервлета, получающего запросы и направляющего их на контроллеры.

Разработка приложения:

Для того, чтобы иметь возможность работать со Spring Framework и его модулями, необходимо подключить зависимости. С этим помогут справиться две известные системы Maven и Gradle, они самостоятельно загрузят и подключат указанные модули.

После генерации и загрузки проект можно открыть в любом удобном IDE направленным на разработку под Java (IntelliJ IDEA, Eclipse, Netbeans).

Файл pom.xml содержит информацию о проекте и подробности конфигурации используемые Maven для сборки проекта.

Напишем небольшое web приложение используя базу данных H2. Для работы с данными применим технологию Hibernate. Следующим шагом в каталоге resources создадим два файла: schema.sql и data.sql. Spring сам найдёт их, создаст базу данных и запишет данные в schema.sql. Следубщим шагом будет реализация паттерна MVC.

Далее нам потребуется инструмент для взаимодействия с базой данных, но у Spring уже есть решение данного вопроса, это интерфейс CrudRepository, он позволяет сохранять, читать, удалять и обновлять данные. Необходимо написать контроллер, его целью будет принятие запроса на данные и передача их представлению. Осталось написать представления и сохранить в папку resource/templates.

Вывод. Spring Framework использует лучшие практики и объединяет множество готовых инструментов освобождая разработчика от написания рутинного кода. Spring экономит время разработчика, при этом предлагает продуманные и гибкие решения возникающих вопросов. На примере, мы убедились, что при создании web приложения под Spring разработчик уделяет большее количество времени на написания логики, за всем остальным следит framework.

Литература:

1. Spring IoC container: https://docs.spring.io/spring-framework/docs/3.2.x/spring-framework-reference/html/beans.html
2. Spring beans @scope: https://docs.spring.io/spring-framework/docs/3.0.0.M3/reference/html/ch04s04.html
3. Spring beans @component: https://docs.spring.io/spring-framework/docs/current/javadoc-api/org/springframework/stereotype/Component.html
4. Spring data: https://spring.io/projects/spring-data
5. Spring MVC: https://docs.spring.io/spring-framework/docs/3.2.x/spring-framework-reference/html/mvc.html