2nd International Conference on Problems of Logistics, Management and Operation in The East-West Transport Corridor (PLMO 2023) Baku, Azerbaijan
DOI:10.54381/plmo2023.14

Java Spring Framework for web applications

Shahnaz SHAHBAZOVA, Agha AGHAYEV
Azerbaijan State Economic University, UNEC, Department of Digital Technologies and Applied Informatics, 6 Istiqlaliyat str., Baku, Azerbaijan
E-mail: shhabazova@gmail.com, shahnaz_shahbazova@unec.edu.az

Abstract. Spring Framework is a universal open-source framework for the Java platform. It is mainly used in the development of web applications. Spring is a set of tools that helps to assemble the structure of the program at the highest level. In this paper, the development of a web application on the Java platform is considered in detail. An overview is given of the functioning of special tools that simplify the work in some aspects. And also the basics of what this set of tools consists of, and what advantages it has.

***Keywords*:** *Spring framework; Spring Security; Spring Data; Spring Beans; Java Persistence API(JPA); MVC; Spring Gradle & Spring Maven; Spring Localhost*

1. INTRODUCTION

Spring is a tool for implementing programs. It simplifies the work of the developer by what it includes, and other frameworks such as Hibernate, Struts, EJB, JSF, and others. Spring simplifies the work by having certain libraries, and it is not necessary to separately assemble different frameworks and other additional libraries, since they are already included in it. Spring itself can process queries to the database, and can also create tables for new records, and the developer does not have to manually create tables. Also, due to Spring, we can configure authentication and authorization, as well as other security features for industrial applications [10]. One of the main components of Spring is the IoC (Inversion of Control) container, which is responsible for managing objects and their dependencies. An IoC container allows you to create objects, manage their lifecycle, and provide access to them through interfaces. Another important component of Spring is Spring MVC, a framework for developing web applications. It includes controllers, views, and data models that are used to process HTTP requests and generate responses.

2. SPRING FRAMEWORK

The Spring Framework consists of several main parts, which can be classified as follows:

- **Core Container:** This is the core part of the Spring Framework, which includes tools for dependency management (DI) and inversion of control (IoC), as well as support for aspect-oriented programming (AOP). Core Container consists of the following modules [1]**:**
  - **Spring Core:** Provides basic DI and IoC tools, including ApplicationContext, BeanFactory, and BeanPostProcessor.
  - **Spring Beans:** Provides tools for creating and managing Spring beans, including annotation support, factory methods, and bean scopes.

- **Spring Context:** Provides advanced tools for working with ApplicationContext, including support for events, resources, and inter-thread synchronization.
- **Spring Expression Language (SpEL):** Provides an expression language for working with Spring configuration files.

- **Data Access/Integration:** This is part of the Spring Framework, which provides tools for working with various data sources, including databases, file systems, and web services. Data Access/Integration consists of the following modules:
  - **Spring JDBC:** Provides support for JDBC and ORM frameworks such as Hibernate and MyBatis.
  - **Spring ORM:** Provides integration with ORM frameworks, including support for JPA and Hibernate.
  - **Spring Transaction:** Provides tools for managing transactions in applications, including support for programmatic and declarative transactions.
  - **Spring Data:** Provides tools for working with various types of data warehouses, including NoSQL databases.
- **Web:** This is part of the Spring Framework, which provides tools for developing web applications, including web services and RESTful services. The Web consists of the following modules:
  - **Spring Web:** Provides tools for developing web applications, including support for controllers and views.
  - **Spring WebFlux:** Provides tools for developing reactive web applications.
  - **Spring WebSocket:** provides tools for working with the WebSocket protocol.
  - **Spring Security:** Provides tools for securing web applications, including out

3. SPRING SECURITY

Spring Security is a web application security framework based on the Spring Framework. It provides mechanisms for authentication, authorization, and resource access control [7].
Spring Security allows you to create secure web applications, and control access to individual resources in the application, such as pages, REST services, or controller methods. It provides security at different levels, including protecting passwords and other sensitive information, protecting against attacks, managing user sessions, etc.

The core principles of Spring Security are the inversion of control, modularity, and reconfigurability. Spring Security allows you to configure application security through XML configuration, annotations, or Java code, which provides flexibility and ease of use.

Spring Security also includes a lot of functionality, such as support for various authentication methods, protection against CSRF attacks, protection against XSS attacks, and other mechanisms that allow protection of the web application from various threats.

All in all, Spring Security is a powerful web application security tool based on the Spring Framework that allows you to build secure and secure applications with a flexible and customizable access control system.

4. SPRING DATA

Spring Data is a Spring Framework module that provides convenient access to databases and other data sources. It provides a unified API for working with various databases, including SQL and NoSQL databases such as MongoDB, Cassandra, Redis, and others. Due to this, we can work on databases such as MySQL, PL/SQL, MongoDB, and so on. It provides many features, such as automatic code generation for working with databases, and automatic creation of SQL queries. Spring Data also provides mechanisms for creating your repositories, which makes it easier to work with the database and increases the portability of code between different databases [4].

In general, Spring Data is a powerful database management tool that allows you to speed up the development process, simplify working with databases and increase the portability of code between different databases.

5. SPRING BEANS

Spring Beans are the basic building blocks of applications developed using the Spring Framework. Classes that are managed by the Spring container and configured as beans are called Spring Beans. Beans are objects that are managed by the Spring container and configured using configuration files or annotations. The Spring container creates beans, sets their properties, manages their lifecycle, and makes them available to other beans and application components [5]. The main benefits of using Spring Beans include the ability to easily manage application dependencies and configuration, making code more readable, extensible, and easy to test. Also, the use of Spring Beans makes it easy to integrate various components and libraries, which simplifies the development and maintenance of the application.

6. JAVA PERSISTENCE API(JPA)

JPA (Java Persistence API) is a standard Java interface for working with object-relational mappings, which provides developers with a convenient and unified way to interact with databases. JPA was created to facilitate the process of working with databases and simplify the application code associated with databases. JPA provides an abstraction for mapping Java objects to database tables and provides the automatic creation of SQL queries for reading and writing data [2].

Main advantages of JPA:

- Simplify working with databases and reduce the amount of application code associated with databases.
- Reduce dependency on a particular database vendor and the ability to migrate to a different database without changing application code.
- Simplifying the process of mapping Java objects to database tables.
- Improve the performance and reliability of applications through the use of caching mechanisms and query optimization.

All in all, JPA is a unified and standardized way of working with databases in Java applications that eases the development process and increases code flexibility and portability across different database vendors.

7. SPRING MVC

Spring MVC (Model-View-Controller) is a module of the Spring framework that provides a powerful mechanism for developing web applications based on the MVC architectural pattern. It is used to create web applications in which the view (View) and application logic (Controller) are separated from the model (Model), which allows you to develop applications more efficiently and flexibly [6].

Spring MVC provides many functionalities such as:

- A powerful mechanism for processing requests and mapping URLs to the appropriate controllers.
- Support for various view types such as JSP, Thymeleaf, Freemarker, Velocity, etc.
- Support for multiple languages (i18n) and localization.
- Support for data validation and error handling.
- Support various query parameters and data formats such as JSON, XML, etc.
- Support for security mechanisms such as Spring Security.
- Integration with other Spring Framework modules such as Spring Data and Spring Boot.

Spring MVC uses the Inversion of Control (IoC) principle, which allows developers to focus on the application's business logic rather than the infrastructure. It also provides flexibility and extensibility, allowing you to create applications of any complexity and scale [3].

All in all, Spring MVC is a powerful tool for building web applications based on the MVC architectural pattern that provides high performance, flexibility, and extensibility.

8. SPRING GRADLE & SPRING MAVEN

The Spring Framework can be used with various dependency management systems, including **Gradle** and **Maven**.

Gradle and **Maven** are tools for automating the build of projects and managing their dependencies. Both tools support the Spring Framework and allow you to manage dependencies, settings, and project configuration.

Spring Gradle is the integration of the **Spring Framework** with the **Gradle Dependency** management system. **Gradle** provides a more flexible and powerful system for managing dependencies and building a project than Maven. However, **Gradle** has a higher entry threshold and may require additional knowledge [8].

Spring Maven is an integration of the **Spring Framework** with the **Maven Dependency** management system. Maven provides an easier and more convenient way to manage dependencies and build a project. However, Maven may limit the flexibility and power of the system.

In general, the choice between **Spring Gradle** and **Spring Maven** depends on the specific requirements of the project, the developer's preferences, and experience with each of the tools.

9. SPRING LOCALHOST

Localhost in Spring generally refers to the local server running on the developer's machine, which is used to test and debug the application.

The Spring Framework often uses the built-in Tomcat web server to run a local server, which can be configured and run using **Spring Boot**. To do this, you need to add a dependency to the pom.xml

file or **Gradle** build file and configure the necessary settings in the application.properties or application.yml file [9].

10. CONCLUSION

In general, the Spring Framework is a powerful and flexible tool for developing applications in Java, which allows you to speed up the development process, increase modularity and reduce code complexity.

References

1. Herbert Java 2 v5.0 (Tiger). New opportunities / Herbert, Schildt. - M.: St. Petersburg: BHV-Petersburg, 2005. - 208 p.
2. Goodman, D. Java Script and DHTML. Collection of recipes for professionals / D. Goodman. - M.: St. Petersburg: Piter, 2004. - 523 p.
3. Dakonta, M. XML and Java 2 / M. Dakonta, A. Saganich. - M.: St. Petersburg: Piter, 2001. - 384 p.
4. Mughal, Khalid A. Java. Guide to preparing for the certification exam CX-310-035 (+ CD-ROM) / Mugal, Khalid A., Rasmussen, Rolf V .. - M .: KUDITS-Obraz, 2006. - 688 p.
5. Noton Java. Reference guide. Everything You Need for Java Programming / Naughton, Patrick. - M.: Binom, 1996. - 448 p.
6. Knowlton, P. Java 2 in the original / P. Knowlton, G. Schildt. - M.: SPb: BHV, 2001. - 526 p.
7. Java programming. - M.: Multimedia Technologies and Distance Education, 2004. - 416 p.
8. Richter CLR via C#. Programming on the Microsoft .NET Framework 2.0 in C# / Richter, Jeffrey. - M.: Piter, 2007. - 656 p.
9. Richter Programming on the Microsoft platform. NET Framework / Richter, Jeffrey. - M.: Russian Edition, 2002. - 512 p.
10. Richter, J. Programming on the .NET Framework / J. Richter. - M.: Microsoft Press. Russian Edition; Edition 2, rev., 2003. - 512 p.