Java backend research

19-9-2017

Sjoerd Brauer
Fontys Venlo

Dear FSG1,

In this document 7 Java frameworks will be described. This document will outline the pros and cons of each framework. Hopefully this will be enough for you to make a calculated unbiased choice.

Criteria for our project are:

1. Framework must work with PostgreSql.
2. Front-end should be able to communicate with backend.
3. Framework must be backward compatible. (as our project will highly likely be carried on by a different group. Time might pass and framework might update).
4.

Sincerely,

Sjoerd Brauer
E: s.brauer@student.fontys.nl

Spring MVC
Spring is a full blown MVC java framework.

Pros:

1. Simplified injection of test data through the use of POJOs.
2. Enhanced modularity, resulting in better code readability.
3. Loose coupling between different modules.
4. Dependency Injection (DI) flexible use.
5. you can avoid using SQL scripts but still retrieve data from your database.
6. You can also create a restful interface which can be extended later on by other programmers.
7. You could also use spring security, so that only certain people have access to the rest interface.
8. Easy to set up OATH2.

Cons:

1. Steep learning curve.
2. Consequences of a conflict to an organization.
3. Types of conflict.
4. Identifying in which state a conflict is according to the conflict model somewhere in the sources.

(Sources)

Strut 2
Struts 2 is a pull-MVC framework. i.e. the data that is to be displayed to user has to be pulled from the Action.

Pros:

1. Configurable MVC components, which are stored in struts.xml file. If you want to change anything, you can easily do it in the xml file.
2. POJO based actions. Struts 2 action class is Plain Old Java Object, which prevents developers to implement any interface or inherit any class.
3. Support for Ajax, which is used to make asynchronous request. It only sends needed field data rather than providing unnecessary information, which at the end improves the performance.
4. Whether you want to use JSP, freemarker, velocity or anything else, you can use different kinds of result types in Struts 2.

Cons:

1. Compatibility
2. Limited Documentation.
3. UI driven framework.

Hibernate

Hibernate ORM (Hibernate in short) is an object-relational mapping tool for the Java programming language. It provides a framework for mapping an object-oriented domain model to a relational database. Hibernate handles object-relational impedance mismatch problems by replacing direct, persistent database accesses with high-level object handling functions.

Pros:

1. Caching mechanism to bug database with similar queries.
2. N+1 or Lazy loading support.
3. Inheritance, encapsulation

Cons:

1. Does not permit multiple inserts
2. Supports less queries then JDBC.
3. Not a good choice for small projects.

akka
Welcome to Akka, a set of open-source libraries for designing scalable, resilient systems that span processor cores and networks. Akka allows you to focus on meeting business needs instead of writing low-level code to provide reliable behavior, fault tolerance, and high performance.

Pros:

1. Multi-threaded behavior without the use of low-level concurrency constructs like atomics or locks âĂŤ relieving you from even thinking about memory visibility issues.
2. Transparent remote communication between systems and their components âĂŤ relieving you from writing and maintaining difficult networking code.
3. A clustered, high-availability architecture that is elastic, scales in or out, on demand âĂŤ enabling you to deliver a truly reactive system.

Cons:

1. Does not run on Java JEE server

Vert.x

Eclipse Vert.x is a polyglot event-driven application framework that runs on the Java Virtual Machine.

1. Non-blocking, event driven runtime
2. Simple to use concurrency and scalability
3. Polyglot (multiple languages can use vert.x)
4.

Cons:

1. Does not run on Java JEE server

JDBC
Java Database Connectivity (JDBC) is an application program interface (API) specification for connecting programs written in Java to the data in popular databases.

Pros:

1. You have complete control.
2. Can manipulate large data sets.
3. Great for handling large data transferring.

Cons:

1. Hard to implement MVC concept.
2. Large programming overhead.
3. UI driven framework.

(Sources)
https://www.slideshare.net/rajkrrsingh/proscons-jdbc-hibernate-ejb
https://stackoverflow.com/questions/35955020/hibernate-orm-framework-vs-jdbc-pros-and-cons