

A Conceptual Study on JMIX-Framework

Ms. Prachi Vedak¹ and Mrs. Gujan Behl²

Student, MCA Final Year¹

Assistant Professor²

Bharati Vidyapeeth' Institute of Management and Information Technology, University of Mumbai, India

Abstract: *We tailor a complex workflow according to customer needs, and Jmix's built-in features such as the BPM module allow us to adapt to each customer's unique work processes. Testing your vision and getting a quick user response is essential for a high quality digital product. Jmix brings the highest productivity development in the first phase of the project, helping to deliver MVPs on days instead of weeks. Jmix is licensed by each developer and does not lock you in proprietary infrastructure. You never get the cost of measuring the value of your customers. An open source framework ensures that your product is completely under your control.*

Keywords: JMIX, CUBA Platform

I. INTRODUCTION

Jmix is an advanced framework for business web applications. Comes with advanced tools and a rich set of functional modules. Jmix is well suited for creating data-based applications with a complex data model and a rich user interface. It is a stack-filled framework in the sense that it provides support in creating both the backend and UI of your application. Jmix is based on Spring Boot, which is a de-facto standard for creating Java web applications for businesses. It means you can use multiple third-party libraries and frameworks with minimal configuration, in addition to the functionality offered by Jmix. Jmix itself contains a number of well-designed modules that you install on your project using launchers, as you would in any Spring Boot app. All modules other than the main one are optional, so you can have only the dependencies you really need. Jmix Studio is an IntelliJ IDEA plugin that assists in all stages of application development: project creation and configuration, defining a data model, generating database transcripts, improving UI screens in the visual editor. Provides advanced navigation, coding and Jmix project-specific testing.

II. BACKGROUND

The background function is designed to perform tasks evenly without restricting user interaction. To use background functions, do the following:

1. Define a function as a subset of BackgroundTask abstract class. Move the reference to the task-linked screen controller and the closing time of the task to the task builder.
2. Locking the screen will interfere with the activities associated with it. Additionally, the function will be interrupted automatically after the expiration of the specified time.
3. Run the function in BackgroundTask.run ().
4. Create an BackgroundTaskHandler class object that controls the task by passing the task pattern in the handle () back of the BackgroundWorker bean. You can get a reference in BackgroundWorker by injecting the screen controller or through ApplicationContext.
5. Start the task by asking for the execute () method of BackgroundTaskHandler

II. LITERATURE REVIEW

2.1 Add-ons and Composite Projects

The concept of add-ons plays an important role when developing with the Jmix framework. In this article, we'll explore what add-ons are and how JmixStudio can help you develop add-ons and modular applications.

Jmix add-ons are fancy words for libraries that are a collection of precompiled code and other resources that your application can use. Unlike the general Java library, which uses the specific term "add-on", this library follows certain rules and leverages some core Jmix features to automatically bring the provided functionality to the host application.

Emphasizes integration into.

Most importantly, the add-on is full stack. This can include entities, database schemas, and UI screens that seamlessly integrate with your application's data model and UI. Simply add the add-on as a build.gradle dependency and you can add a whole new subsystem to your project using the data stored in the database and the user interface integrated into the main menu.

Add-ons do not have to be complex full stack projects. Many add-ons provide only UI features or alternative implementations of framework interfaces like File Storage. In this case, the add-on simply uses a common infrastructure for market publishing and easy installation on the project.

-Konstantin Krivopustov

2.2 A Definitive Guide to Application Modernization

Jmix is a dedicated tool for building business applications. Reduce friction and increase speed by focusing on the area in question.

Jmix covers the basics of efficient software development with add-ons with a full stack framework integrated on top of the Java ecosystem, support for developer tools, and solutions for common tasks. increase.

The modular and open architecture makes the framework extremely flexible. Core functionality can be expanded or modified almost freely. Jmix has a growing community and active development in its core framework and add-on ecosystem

Efficient software development accelerates the feedback cycle, increases user interaction, and provides a better solution for the enterprise. And that's what we all want, right?

If this sounds promising to you and you want to know more, jump to the quickstart or try it yourself. There is also an open community waiting for you, so feel free to ask questions.

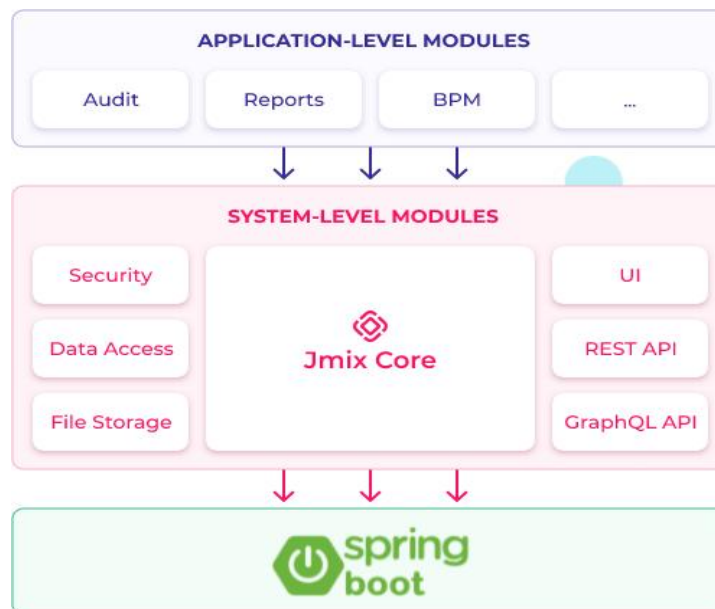
Andrey Glaschenko

III. ADD - ONS

Jmix doesn't just offer a platform to build your apps on. It also contains an ecosystem of additives that enrich essential functions. Many of the additions are provided by the Jmix team, and some are donations from the community.

Jmix supplements are published in the Jmix market place. You can browse and upload them directly to the Studio as described below. To publish your open source app in the marketplace, see Publishing Addition

IV. ARCHITECTURE



V. DATA MODEL

The data model is an integral part of any business plan. At Jmix, application data is represented by businesses. The business can be stored in an external storage location or only in memory. In both cases, the business is associated with a data store, which is more than just a data source in Jmix applications.

5.1 Entities

JPA entities are Java objects stored on the site using the Java Persistence API. DTO entities are simple Java items that cannot be tied to any particular persistent technology. Key- Value Entity is a dynamic business with an incorrect number of attributes. Businesses are identified by their attributes. JPA and DTO businesses are defined in Java classes and have specific annotations specific to Jmix.

5.2 Data Sources

The data store represents the website and any other data source in your application. Jmix provides JpaDataStore that stores JPA businesses on a related website. This is a great way to persevere in Jmix applications, and when we mention a data store in this manual, we always mean JpaDataStore if it is not explicitly stated otherwise.

5.3 Data Types

Each non-reference business attribute is associated with the use of the Datatype interface. This visual interface describes ways to convert attribute values to or from a character unit (formatting and splitting) when displaying businesses in the Backoffice UI and generated in Generic REST.

The framework provides a Datatype application set that is compatible with the most common types of business attributes.

5.4 Enumerations

Calculating a set of constants that can be used as values for a business attribute.

In Jmix, counts to Java enum using the EnumClass interface and has an Integer or String type id field. This field is stored on the site instead of the ordinal () or enum name (), which has the following benefits:

You can rename and rearrange enum constants, and as long as you keep the identifiers, your data model will work with the content on the website.

Even if the site contains a value that does not match any fixed enum (occurs if older and newer versions of the application work with the same site), the business can still be loaded with the enum attribute containing null. With the standard JPA method, loading such a model is not possible at all.

VI. BACK OFFICE - UI

Jmix Backoffice UI allows you to develop a rich web UI using Java and XML. Based on the Vaadin web framework.

In the Backoffice UI, your UI components work on the same JVM as your backend, which makes it much easier to work with data and an attractive business mindset. Also, you do not need to be familiar with the modern JavaScript / HTML / CSS stack.

Backoffice UI technology has a limited scope: the back side of the UI is configured and uses a number of resources in parallel with the number of user intervals at a time. The Backoffice UI is therefore ideal if you have a predicted number and not too high (up to ten thousand) users at one time.

With a web-based project and the complete freedom to customize the UI, consider creating a pre-existing application using traditional advanced technologies. It can work with Jmix backend with REST API.

VII. SECURITY

Jmix Sub-program security provides an easily configurable access control system for your application. It is based on Spring Security and adds the following features:

Integration with your data model.

Permissions to apply CRUD performance to businesses and to view / modify specific business attributes. For example, a user can view documents, but cannot create, update, or delete them, and can view all the attributes of a document without

value. Controlling line-level access limits access to individual business events. For example, users can view only the documents created in their department. Integration with the UI.

Permissions to unlock Backoffice UI screens and see main menu items.

When a visual field such as a text field is tied to a business attribute, it automatically becomes read-only or hidden depending on the current user rights of the attribute. Table actions are closed when compatible CRUD tasks are denied by the user. Disclosure of roles and permissions using Java interface.

Ability to define roles and permissions during operation and maintain them on the website.

Backoffice UI for viewing roles and permissions, assigning roles to users, and creating downtime roles. Jmix doesn't just offer a platform to build your apps on. It also contains an ecosystem of add-ons that enrich essential functions. Many of the additions are provided by the Jmix team, and some are donations from the community. Jmix supplements are published in the Jmix marketplace. You can browse and upload them directly to the Studio as described below. To publish your open source app in the marketplace, see Publishing Addition

VIII. INTRODUCTION - CUBA PLATFORM

CUBA started its way back in 2008. Since then it has gone through a few very important stages. Originally, it was an unencrypted internal framework with a small API. It was a company-wide initiative that allowed Haulmont to develop Business Applications quickly.

The CUBA forum is an ideal tool for development teams working on business line applications, often with a comprehensive data model, hundreds of screens and a complex business mindset.

Based on a common technology stack, the CUBA platform delivers unparalleled production through a rich set of data-sensitive components, a wide scaffolding, a visual connector designer and a warm-up application.

Open structures allow the developer to customize any part of the framework, providing high levels of control and flexibility. Developers are free to use the popular Java IDEs and have full access to the source code.

CUBA applications easily fit into the IT business environment, supporting large websites and application servers, as well as the popular aPaaS cloud. Integrated deployment ensures durability and failure, while the standard REST API enables easy integration with other systems.

IX. NEW VERSION OBJECTIVES

Bring the developer's imagination closer to the most popular frames. The CUBA Platform uses the Spring, but these days the Spring Boot has almost conquered the world. New models are emerging - Micronaut and Quarkus. They all have the same principles at their core: easy configuration using .properties or .yaml files, extensive use of annotations, and easy linking and configuration. We therefore want CUBA to provide the same feel for the developers.

Do not reset the tire. Since 2008 new libraries and tools have been established. They are now mature and can be used for business grade applications. So we wanted to install CUBA custom modules with war-proven libraries. For example - a database transit system.

Make CUBA applications smaller. When you create apps with CUBA, you do not always need all the features like research. But it has always been part of the backbone framework, contaminating database with unnecessary tables (in your particular case) and starting additional services on your application server. Therefore, it would be good to have the power to remove some CUBA features and install them only if necessary.

And most importantly - maintain a good experience and speed of application development

X. CONCLUSION

Jmix is the next big step in CUBA Platform evolution. Now you can enjoy almost every Spring Boot launcher and apply the techniques that work on the world-famous Java framework. At the same time, you still have all the useful and standard APIs and functionality provided by Jmix - CUBA interest and excellent development experience for the new Studio. You can check out Jmix preview release on jmix.io. Discuss new releases in the special forum section.

ACKNOWLEDGMENT

The heading of the Acknowledgment section and the References section must not be numbered. Causal Productions wishes to acknowledge Michael Shell and other contributors for developing and maintaining the IEEE LaTeX style files which have been used in the preparation of this template.

REFERENCES

- [1]. <https://docs.jmix.io/jmix/intro.html>
- [2]. <https://www.jmix.io/blog/jmix-the-future-of-cuba-platform/>
- [3]. <https://www.jmix.io/cuba-platform/learn/quickstart/studio/>
- [4]. <https://dzone.com/articles/jmix-the-future-of-cuba-platform>
- [5]. <https://doc.cuba-platform.com/manual-6.5/overview.html> (2002) The IEEE website. [Online]. Available: <http://www.ieee.org/>
- [6]. <https://docs.jmix.io/jmix/reports/>
- [7]. <https://stackoverflow.com/questions/66691678/create-jmix-project-on-eclipse>
- [8]. <https://www.jmix.io/>
- [9]. <https://github.com/cuba-platform/cuba>
- [10]. <https://forum.cuba-platform.com/>
- [11]. <https://www.jmix.io/blog/add-ons-and-composite-projects/>