

Quarkus, Kogito, and ChatGPT

Automating Business Processes the Intelligent Way!

Markus Herhoffer

Dipl.-Inform.

CTO exentra

Lecturer at Technische Hochschule Ingolstadt

<https://www.linkedin.com/in/markus-herhoffer/>

https://mastodon.social/@d135_1r43

<https://bento.me/herhoffer>

markus.herhoffer@exentra.de



Complaint Management for Mario's Pizza Place

Mario runs a pizza place, yet his customers are not very satisfied.

He needs a web app to manage all those terrible complaints. He also wants AI!!!!!!11

Traditional Approach

REST Controllers with a bunch of Java classes and a hardcoded implicit business process



Process Driven Approach

Well designed BPMN business process calling minimalistic and isolated Java classes



Documents Associated With Business Process Model And Notation™ (BPMN™) Version 2.0

Release Date: January 2011

Normative Documents

OMG document number	Explanation	Format	URL
formal/2011-01-03	v2.0	PDF	https://www.omg.org/spec/BPMN/2.0/PDF
formal/2011-01-03	v2.0	PS	https://www.omg.org/spec/BPMN/2.0/PS

Normative Machine Consumable Files

OMG document number	Explanation	Format	URL
dtc/2010-05-04	XSD schema, XMI schema, and XSLT transformations	cmof	https://www.omg.org/spec/BPMN/20100501/BPMN20.cmo
dtc/2010-05-04	XSD schema, XMI schema, and XSLT transformations	cmof	https://www.omg.org/spec/BPMN/20100501/BPMNDI.cmo
dtc/2010-05-04	XSD schema, XMI schema, and XSLT transformations	cmof	https://www.omg.org/spec/BPMN/20100501/DC.cmo
dtc/2010-05-04	XSD schema, XMI schema, and XSLT transformations	cmof	https://www.omg.org/spec/BPMN/20100501/DI.cmo
dtc/2010-05-04	XSD schema, XMI schema, and XSLT transformations	xsd	https://www.omg.org/spec/BPMN/20100501/BPMN20.xsd
dtc/2010-05-04	XSD schema, XMI schema, and XSLT transformations	xsd	https://www.omg.org/spec/BPMN/20100501/BPMNDI.xsd
dtc/2010-05-04	XSD schema, XMI schema, and XSLT transformations	xsd	https://www.omg.org/spec/BPMN/20100501/DC.xsd
dtc/2010-05-04	XSD schema, XMI schema, and XSLT transformations	xsd	https://www.omg.org/spec/BPMN/20100501/DI.xsd
dtc/2010-05-04	XSD schema, XMI schema, and XSLT transformations	xsd	https://www.omg.org/spec/BPMN/20100501/Semantic.xsd
dtc/2010-05-04	XSD schema, XMI schema, and XSLT transformations	xsd	https://www.omg.org/spec/BPMN/20100501/BPMN20-

Was ist Kogito?

Cogito ergo sum

BPMN Engine

Kogito runs BPMN processes by compiling them on build time to Java classes.

Quarkus

Kogito is a bunch of extensions that are optimised for Quarkus.
They also support Spring Boot.

REST, CloudEvents, Kafka

Kogito uses established standards and technologies to implement the process flows.

Demo

The good...

Best Practices

Separation of Domains

Separate into several processes inside several microservices.

Separation of Integration

Do not mix up business processes and integration in the same service.

No bookkeeping

Delegate bookkeeping and auditing to the process engine.

Know your BPMN

Use compensation and error events.

Type safety

Be as type safe as possible, define the contracts well.

DNRY

Reuse processes.

The bad and the ugly...

The Downsides

Devops Overhead

Kafka, Infinispan, DataIndex and tons of services.

BPMN incomplete

Some parts of the BPMN standard are not yet implemented.

Bad Debugging

Debugging is difficult due to complexity and missing/unstable features.

Extremely strict

In comparison to Camunda and Flowable extremely strict typing.

Full stack knowledge

You need to rock Kafka, Infinispan, BPMN, Java, Quarkus and MP Messaging.

VS Code only

No BPMN editor for IntelliJ IDEA or Eclipse IDE.

Where to continue

<https://github.com/d135-1r43/restaurant-complaints>

The screenshot shows a GitHub repository page for 'Kogito/Quarkus PDA Example: Restaurant Complaints'. The main content is the README.md file, which describes a demo application created using Quarkus and Kogito. It lists three services: Complaints Service, Sentiment Analysis Service, and Archive Service. Below the README, there is a 'Prior Knowledge' section with a list of prerequisites and a 'Walkthrough' section with a preview of a BPMN diagram. On the right side, there is a sidebar with the user profile 'sioex', a 'Languages' section showing Java at 98.2% and HTML at 1.8%, and 'Suggested Workflows' such as 'Actions Importer', 'Java with Ant', and 'Publish Java Package with Gradle'.

README.md

Kogito/Quarkus PDA Example: Restaurant Complaints

This repository contains a basic Process Driven Applications (PDA) demo application created using [Quarkus](#) and the [Kogito](#) process engine. The purpose of this demo is to demonstrate key concepts of Process Driven Applications.

The application includes three services:

1. **Complaints Service** (`de.thi.complaints`): A Quarkus service that handles restaurant complaint submissions.
2. **Sentiment Analysis Service** (`de.thi.sentiment`): A Quarkus service that assesses the sentiment of the complaint. It uses OpenAI (ChatGPT) and a User Task as a fallback.
3. **Archive Service** (`de.thi.archive` , optional): A Quarkus service with a simple REST API to store complaints. This service can be integrated into the Complaints Service.

Prior Knowledge

To understand this walkthrough you should already be able to:

- Open and edit Kogito BPMN in VS Code and run a simple Kogito Quarkus app, as described at [Installing the Kogito VSCode extension bundle without the Kogito Business Modeler Hub](#)
- Understand the basics BPMN and the Process Driven Approach
- Start a Kogito process with the Swagger UI and perform User Tasks in the Dev UIs.

You must

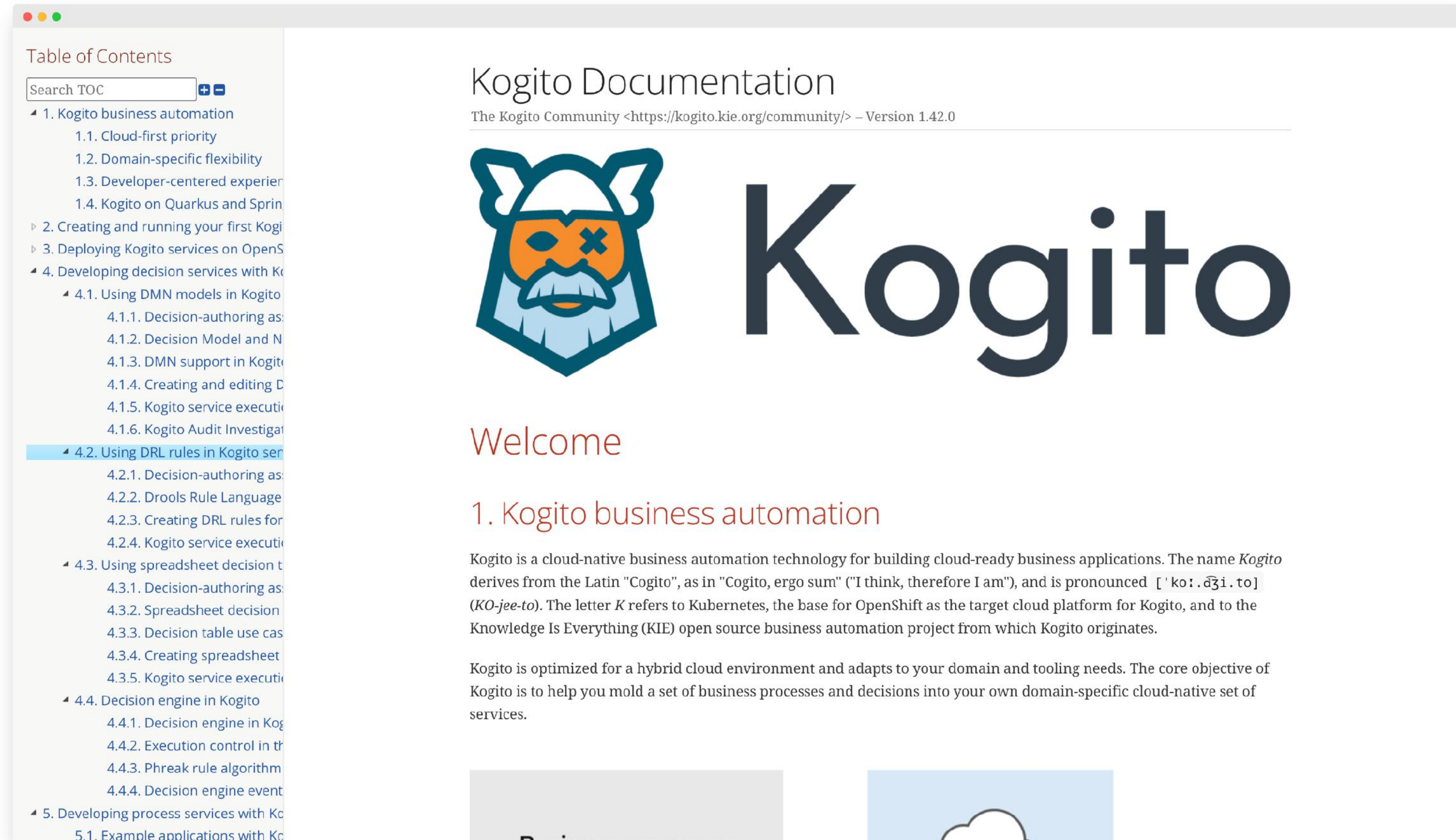
- have Docker running
- have a working developer environment for Java 17 with Maven and an IDE of your choice (VS Code preferred)

Walkthrough

The walkthrough section shows a preview of a BPMN diagram in VS Code. The diagram illustrates a process flow for handling restaurant complaints, starting with a start event, followed by a task to 'Receive Complaint', then a decision event to 'Check Sentiment', and finally a task to 'Archive Complaint'.

Where to continue

Kogito Documentation and Examples Repository



The screenshot shows a web browser window displaying the Kogito Documentation page. On the left, there is a 'Table of Contents' sidebar with a search box and a list of topics. The main content area features the Kogito logo (a stylized blue and orange character) and the title 'Kogito Documentation'. Below the logo, the text 'Welcome' is followed by the section '1. Kogito business automation'. The text describes Kogito as a cloud-native business automation technology and provides pronunciation information. It also mentions that Kogito is optimized for a hybrid cloud environment.


Table of Contents

Search TOC

- 1. Kogito business automation
 - 1.1. Cloud-first priority
 - 1.2. Domain-specific flexibility
 - 1.3. Developer-centered experier
 - 1.4. Kogito on Quarkus and Sprin
- 2. Creating and running your first Kogi
- 3. Deploying Kogito services on OpenS
- 4. Developing decision services with Kogito
 - 4.1. Using DMN models in Kogito
 - 4.1.1. Decision-authoring as
 - 4.1.2. Decision Model and N
 - 4.1.3. DMN support in Kogito
 - 4.1.4. Creating and editing D
 - 4.1.5. Kogito service executi
 - 4.1.6. Kogito Audit Investigat
 - 4.2. Using DRL rules in Kogito ser
 - 4.2.1. Decision-authoring as
 - 4.2.2. Drools Rule Language
 - 4.2.3. Creating DRL rules for
 - 4.2.4. Kogito service executi
 - 4.3. Using spreadsheet decision t
 - 4.3.1. Decision-authoring as
 - 4.3.2. Spreadsheet decision
 - 4.3.3. Decision table use cas
 - 4.3.4. Creating spreadsheet
 - 4.3.5. Kogito service executi
 - 4.4. Decision engine in Kogito
 - 4.4.1. Decision engine in Kog
 - 4.4.2. Execution control in th
 - 4.4.3. Phreak rule algorithm
 - 4.4.4. Decision engine event
- 5. Developing process services with Kogito
 - 5.1. Example applications with Kog

Kogito Documentation

The Kogito Community <<https://kogito.kie.org/community/>> – Version 1.42.0



Kogito

Welcome

1. Kogito business automation

Kogito is a cloud-native business automation technology for building cloud-ready business applications. The name *Kogito* derives from the Latin "Cogito", as in "Cogito, ergo sum" ("I think, therefore I am"), and is pronounced [' ko : . ə ʒ i . to] (*KO-jee-to*). The letter *K* refers to Kubernetes, the base for OpenShift as the target cloud platform for Kogito, and to the Knowledge Is Everything (KIE) open source business automation project from which Kogito originates.

Kogito is optimized for a hybrid cloud environment and adapts to your domain and tooling needs. The core objective of Kogito is to help you mold a set of business processes and decisions into your own domain-specific cloud-native set of services.

Business processes

Where to continue

Books



Where to continue

Podcast



German Podcast
with Lucas and Markus



Thank you!

For excellent Java Developers
who want to work with Quarkus:
karriere@exentra.de

How to reach us

exentra GmbH

Löwenstraße 11
85276 Pfaffenhofen
+49 8441 47247 0

web: exentra.de

Email: kontakt@exentra.de

