

Building Intelligent Applications with Kogito

PV207 BPM Labs

Mgr. Marian Macik
Principal Software Quality Engineer

May 2024



KOGITO

<https://kogito.kie.org/>

- Domain-Driven Development
- Generated Domain-specific APIs from your BPMN and DMN models
- Lightweight orchestration microservices
- Event-driven business logic
- Dev-mode hot reload
- Distributed sagas for microservices
- Serverless workflows
- Polyglot programming

More info at

- [Blogs](#)
- [Youtube](#)

PREREQUISITES

- Java 11 or Java 17 with JAVA_HOME configured
- [Visual Studio Code](#)
- [Kogito Bundle Extension](#) for VS Code
- Docker (only for Dev UI)

YOUR FIRST KOGITO APPLICATION

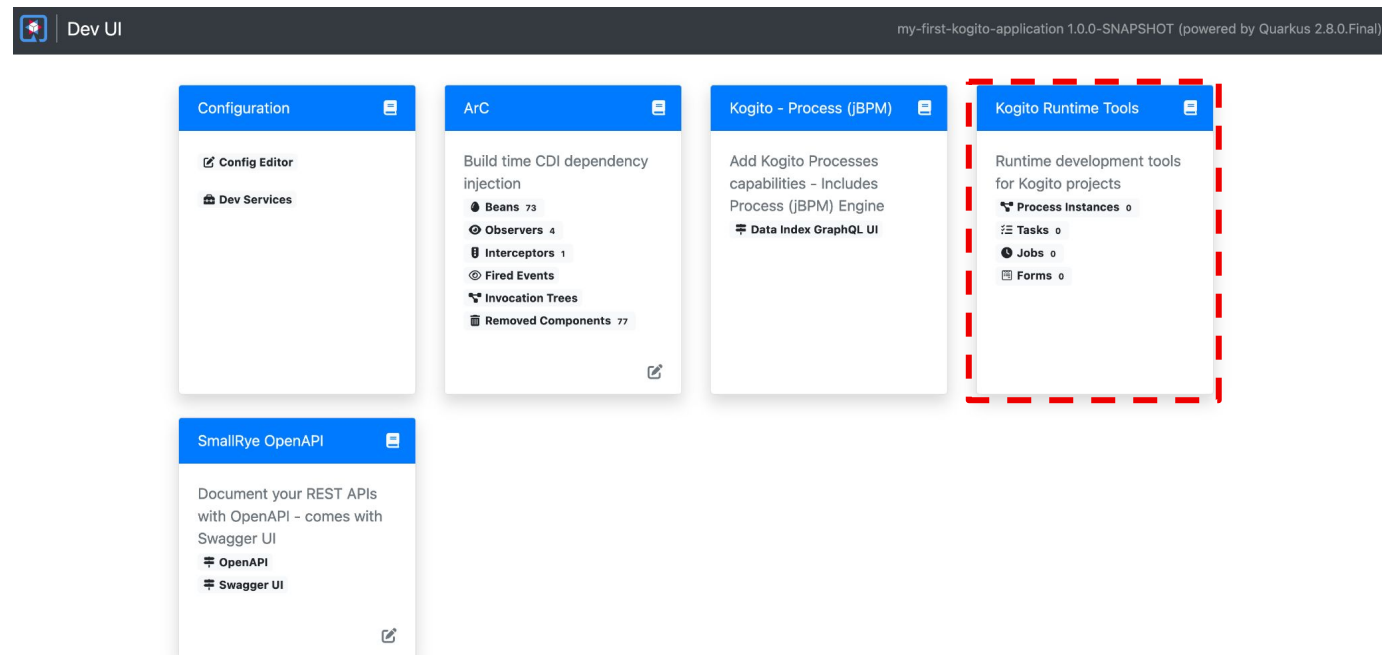
- Generate a preconfigured Quarkus application with Kogito extension at code.quarkus.redhat.com
 - Generate your application -> Download as a zip
 - Version without Dev UI available [here](#)
- Design BPMN/DMN in src/main/resources folder
 - VS Code
 - bpmn.new
 - If using Dev UI, download also SVG file and put it under src/main/resources/META-INF/processSVG
 - If your process definition file is myProcess.bpmn, the svg file must be called myProcess.svg
- Or used the prepared [myProcess](#)
- Start the application in development mode*
 - ./mvnw clean compile quarkus:dev
- Domain-specific API available at <http://localhost:8080/q/swagger-ui/>
 - OpenAPI specification
 - Generated only in development mode by default

* set `<maven.compiler.release>11</maven.compiler.release>` in the pom.xml if you use Java 11 if not already set

YOUR FIRST KOGITO APPLICATION

Dev UI (only available with Docker)

- Start the process using Swagger UI
- Go to <http://localhost:8080/q/dev/> and select Process Instances on the Kogito Runtime Tools card
- Look at the process in detail
- Same for the task, you can complete the task using UI too



YOUR FIRST KOGITO APPLICATION

- Change the greeting phrase
 - Edit an asset, e.g. the script task
 - Execute a command to invoke the Kogito endpoint again
 - Asset is recompiled and a new version is invoked
- Compile
 - `./mvnw clean install`
 - Append `-Dquarkus.swagger-ui.always-include=true` to include Swagger UI in production
- Run
 - `java -jar target/quarkus-app/quarkus-run.jar`
 - Application uses **whole quarkus-app** directory
- Run on a different port
 - `java -Dquarkus.http.port=8090 -jar target/quarkus-app/quarkus-run.jar`

YOUR FIRST KOGITO APPLICATION

"Raw commands"

- Start a process

```
curl -X POST -H 'Content-Type:application/json' -H 'Accept:application/json' -d '{"personName": "John"}'  
http://localhost:8080/myProcess
```

- Show processes

```
curl -H 'Accept:application/json' http://localhost:8080/myProcess
```

- Show Tasks

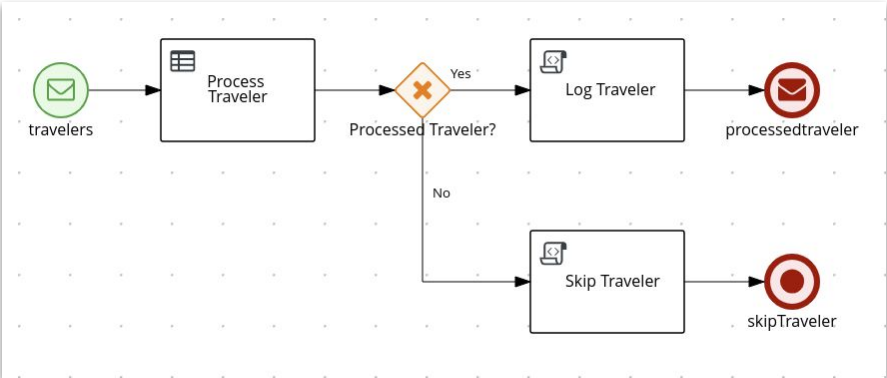
```
curl -H 'Accept:application/json' http://localhost:8080/myProcess/{puuid}/tasks?user=admin
```

- Complete a task

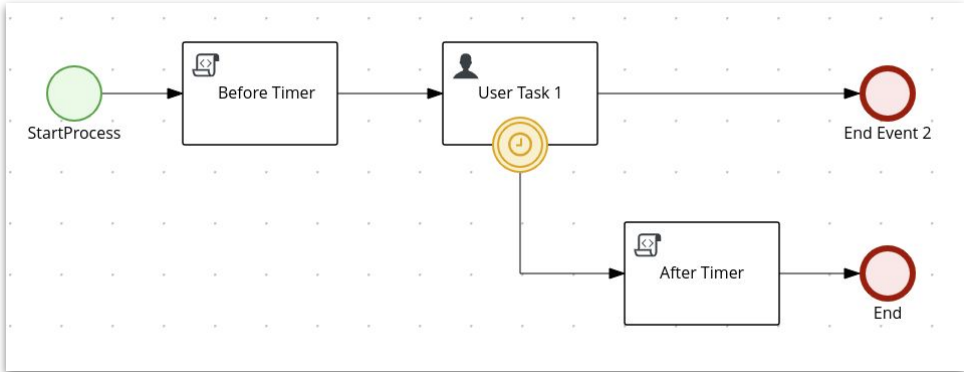
```
curl -X POST -H 'Content-Type:application/json' -H 'Accept:application/json' -d '{"personName": "John Doe"}'  
http://localhost:8080/myProcess/{puuid}/{taskName}/{tuuid}?user=admin
```

KOGITO EXAMPLES

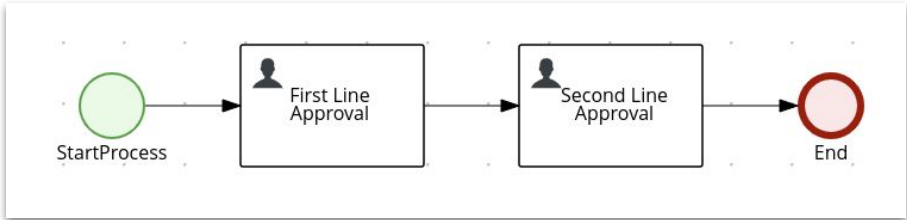
<https://github.com/apache/incubator-kie-kogito-examples>



KAFKA MESSAGE EVENTS
process-kafka-quickstart-quarkus



TIMERS
process-timer-quarkus



4-EYE PRINCIPLE
process-usertasks-quarkus

● ● ● and more

Thank you,
questions?