# AKTIMETRIX

## *A Business Activity Monitoring Framework*

POWERED BY N|Solid

build passing

Aktimetrix is a generic business process monitoring cloud native analytical streaming framework intended for any domain.

## Features

Aktimetrix provides below high level features.

- Allows designing 'multiple business process' for any domain
- Allows defining the multiple Plans for single process
- Monitors business activity
- Generate the notifications
- Generates Metric and KPIs

## Architecture

Aktimetrix framework is based on micro service architecture as event component is driven by events (event driven) from the messaging broker of your choice.

Aktimetrix primarily exposes below domain language. *Reference Data*

- Process - Collection of Step
- Step - A individual milestone in the process
- Metadata - A domain specific information
- Measurement - A definition of calculation to be performed
- Metric - Aggregation of measurement
- Business Entity - External business object injected into the aktimetrix.
- Process Instance - instance of process for the belong to business entity
- Step Instance - instance of step
- Measurement Instance - instance of measurement
- Metric Instance - instance of metric

The following are the primary APIs which are included as part of aktimetrix.

- ***Reference data API*** - This REST style API will allow to define the reference data required for aktimetrix such as process definitions, step definitions, measurement definitions and metric definitions.(more about these in coming sections)
- ***Processor API*** - Processor API allows to create the process instance and step instance etc for each business entity.
- ***Meter API*** - Meter API will capture measurements from the incoming process instance event and step instance events.

- ***Planner API*** - Planner API will allow to create the plan based on the process instance, step instance and measurement instances.

Aktimetrix requires Java v8+ and spring boot v2.6.6 to run.

## Tech

Aktimetrics uses a number of open source projects to work properly:

- Apache Kafka - Apache Kafka is a distributed event store and stream-processing platform.!
- Mongodb - MongoDB is a source-available cross-platform document-oriented database program.
- Spring Cloud Steam - Markdown parser done right. Fast and easy to extend.
- Java 8+ - Java Language
- Maven - Apache Maven is a software project management and comprehension tool.
- Docker - Docker is a set of platform as a service products that use OS-level virtualization to deliver software in packages called containers. ..and more

And of course Aktimetrix itself is open source with a public repository on GitHub. Install the dependencies and devDependencies and start the server.

## Development

Developers can bootstrap the development of aktimetrix for by following below steps. Create a base spring boot application using spring initializer. Add aktimetrix core dependency to the project

```
<dependency>
    <groupId>com.aktimetrix</groupId>
    <artifactId>aktimetrix-core</artifactId>
    <version>0.0.1-SNAPSHOT</version>
</dependency>
```

### Event Handler

Event handler are responsible for processing the external business events. For example in cargo domain to create a handler for processing 'BKD' event create class which extend the AbstractEventHandler class from aktimetrix core and override the handle method.

You should annotate the class with @EventHandler annotation

```
@EventHandler(eventType = EventType.BKD)
public class BKDEventHandler extends AbstractEventHandler {
    ...
    @Override
    public void handle(Event<?, ?> event) {
        super.handle(event);
    }

    @Override
     public abstract String entityId(Event<?, ?> event){
         Cargo cargo = (Cargo)event.getEntity();
```

```
          String entityId = cargo.getDocumentInfo().getAwbInfo().getDocumentPrefix() +
"-" + cargo.getDocumentInfo().getAwbInfo().getDocumentNumber(); // domain specific
entity id
         return entityId;
      }

    @Override
    public abstract String entityType(Event<?, ?> event){
        return "ciq.cargo.awb" // domain specific namespace
      }
    ..
}
```

This will search for the process definitions whos start event is 'BKD' and execute the processor configured for that process.

### Processor

Processor will create the process instance and step instances associated witht the process. Processor will read the definitions from the reference data.

For performing any business/domain specific infromation can be stored as metadata. A metadata can be associated with *process instance* or *step instance* .

For example in IATA Cargo IQ - CDMP-C process cargo information can be associated with process instance of air waybill process.

```
@Component
@ProcessHandler(processType = ProcessType.CDMP_C)
public class CiQA2AProcessor extends AbstractProcessor {
     @Override
    protected Map<String, Object> getStepMetadata(ProcessContext context)
    {
        return new HashMap<>();
    }
    @Override
    protected Map<String, Object> getProcessMetadata(ProcessContext context)
    {
        return new HashMap<>();
    }
}
```

### Meters

### Planner

Want to contribute? Great!

Aktimetric uses Spring + Spring boot for fast developing. Make a change in your file and instantaneously see your updates!

## License

MIT

**Free Software, Hell Yeah!**