Audit Chain (AMOS SS 2022)

Here you can find the build and deploy guide for our project ☺
For further information, check the Documentation Folder on our GitHub Repo ☺

1. Programming Languages:

- **Java (Backend):** Java is an object-oriented programming language that consists of a development tool for creating code and a runtime environment to run code.
- **Python:** It is a computer programming language often used to build scripts, automate tasks, and conduct data analysis. Its purpose here is to create, in combination with PySide2, a GUI.

2. Prerequisites

- Git
- Java SDK 16
- Docker or Ubuntu Linux

3. Build, Deployment Documentation, and Testing

- **Clone the repository from GitHub:**

  - Clone via HTTPS:
    - git clone https://github.com/amosproj/amos2022ss02-audit-chain.git

  - Clone via SSH:

    - git clone git@github.com:amosproj/amos2022ss02-audit-chain.git

  - *Alternative* downloads the latest release and unzip the package from git.

    - https://github.com/amosproj/amos2022ss02-audit-chain

Setup Rabbit

Docker Version

- Credentials used for rabbitmq in this project:
  - username: admin
  - password: admin
- To change the credential, update RABBITMQ_DEFAULT_USER, RABBITMQ_DEFAULT_PASS in docker-compose.yml file
- Create a network by the name rabbitmq-cluster:
  - docker network creates rabbitmq-cluster
- Get the middleware, i.e., rabbitmq-cluster running in daemon mode:
  - docker-compose up –d

Virtual Machine Version

- Since the VM Version is more complex, click here to see the document, which also holds the script you must execute.

Build Project

- The building tool is Maven, and the following command is used.
  - mvn package
    - execute JUnit Tests
    - Create Jar Files for ProducerDummy, ConsumerDummy and BlockchainDummy, and Producer_Consumer_Dummy.

Run Project

- Prerequisite

- Our software architecture is component-based therefore to successfully run the Project, you have two options to set the components:
    1. Adjust the **config.properties** File according to your needs. You can find the respective config.properties file in the following location: /src/main/resources/module_name. By default, the modules check if this Path/File exists and will use this if found. As a Fallback, if it does not find the config.properties there, it checks the current Path for a config.properties file.
    2. If you just want to check how it is working, there is minimal cmd support:
       h, --host <arg> (host IP of docker (127.0.0.1))
       p, --port <arg> (port (5672))
       pw, --password <arg> (password of RabbitMQ)
       u, --username <arg> (username of RabbitMQ)

Run the Project via Maven

- mvn clean compile exec:java@ProducerDummy
    - This Command starts the ProducerDummy.

- mvn clean compile exec:java@ConsumerDummy
    - This Command starts the ConsumerDummy.

- mvn clean compile exec:java@Blockchain
    - This Command starts the BlockchainConsumerDummy.

Run the Project via Jar

- java -jar AuditChain-ConsumerDummy.jar
- java -jar AuditChain-ProducerDummy.jar
- java -jar AuditChain-Producer_ConsumerDummy.jar
- java -jar AuditChain-Blockchain.jar

Note: *Remember to use the username and password of rabbitmq as arguments or edit the [config. properties]*

- **Testing**

 The applications of our project contain unit tests that can be used to test the individual components. To run them locally, just execute the following command:
    - mvn clean compile test