# **Mock ABIS**

## **About**

* A module to mock ABIS functionality for testing non-production MOSIP deployments.
* This module also provides swagger API to test only INSERT/IDENTIFY functionality.

## **Requirements**

Partner certificate (in case of encryption is enabled) cbeff.p12 we put into resource folder.

**## Setup**

In case of partner based encryption. Upload the certificate, by using swagger upload certificate request

**### Server deployment (for sandbox deployment)**

Docker based

Steps:

\* Go to REPO\_ROOT/mock-abis.

\* Build the code: `mvn clean install -Dmaven.test.skip=true -Dgpg.skip=true`.

\* Create a docker image: docker build . --file Dockerfile --tag mock-abis.

\* Push the docket image to docker registry. You can directly use these images for running mock ABIS.

**\*\*Please check Dockerfile for passing env properties\*\***

Swagger url: `https://<hostname>/v1/mock-abis-service/swagger-ui.html#/`

**### Local dev against Server (for testing against server)**

This section is for the developers to run mock-abis locally against MOSIP server

Steps:

\* Go to REPO\_ROOT/mock-abis.

\* Setting ABIS queue conf (here queue details will be for server's queue):<br>

1) Create registration-processor-abis.json in resources.<br>

2) Copy the contents of registration-processor-abis-sample.json to registration-processor-abis.json.<br>

3) Update registration-processor-abis.json with the correct queue details.<br>

4) Download the latest kernel-auth-adapter from `https://mvnrepository.com/artifact/io.mosip.kernel/kernel-auth-adapter` and save into lib folder .<br>

\* Build the code: `mvn clean install -Dmaven.test.skip=true -Dgpg.skip=true`.

\* Run the jar: `java -Dloader.path=lib/kernel-auth-adapter-1.3.0-SNAPSHOT.jar -Dlocal.development=true -Dabis.bio.encryption=true -Dspring.profiles.active=local -Dmosip\_host=https://<server hostname> -jar target/mock-abis-1.3.0-SNAPSHOT.jar`.

Flags:

\* local.development (true: whenever running locally, this will take the registration-processor-abis.json from resources)

\* abis.bio.encryption (true: in case if partner based encryption)

\* mosip\_host (hostname of the MOSIP server)

Swagger url: `http://localhost:8081/v1/mock-abis-service/swagger-ui.html#/`

**### Fully local (for active development)**

This section is for developers to run mock-abis locally against a queue, messages can be inserted directly to the queue and respective mock-abis responses can be analysed.

Requirements:

\* Local ActiveMQ server

Running ActiveMQ locally:

\* Go to REPO\_ROOT/mock-abis/activemq

\* use the docker compose file to create a local activemq server: `docker-compose up`

\* Open activemq web console: `http://localhost:8161/`

Steps:

\* Go to REPO\_ROOT/mock-abis.

\* Setting ABIS queue conf (here queue details will be for local queue):<br>

1) Create registration-processor-abis.json in resources.<br>

2) Copy the contents of registration-processor-abis-sample.json to registration-processor-abis.json.<br>

3) Update registration-processor-abis.json with the correct queue details.<br>

{

"abis": [

{

"name": "ABIS",

"host": "",

"port": "",

"brokerUrl": "tcp://{env}.mosip.net: {port}",

"inboundQueueName": "ctk-to-abis ",

"outboundQueueName": "abis-to-ctk ",

"pingInboundQueueName": "ctk-to-abis ",

"pingOutboundQueueName": "abis-to-ctk ",

"userName": "artemis",

"password": "{password}",

"typeOfQueue": "ACTIVEMQ",

"inboundMessageTTL": 2700

}

]

}

4) \* for local database we use following

javax.persistence.jdbc.url=jdbc:h2:mem:testdb;DB\_CLOSE\_DELAY=-1

javax.persistence.jdbc.driver=org.h2.Driver

javax.persistence.jdbc.user=sa

javax.persistence.jdbc.password=sa

hibernate.ddl-auto=update

hibernate.temp.use\_jdbc\_metadata\_defaults = false

hibernate.dialect = org.hibernate.dialect.H2Dialect

5) Download the latest kernel-auth-adapter from `https://mvnrepository.com/artifact/io.mosip.kernel/kernel-auth-adapter` and save into lib folder .<br>

\* Build the code: `mvn clean install -Dmaven.test.skip=true -Dgpg.skip=true`.

\* Run the jar: `java -Dloader.path=lib/kernel-auth-adapter-1.3.0-SNAPSHOT.jar -Dlocal.development=true -Dabis.bio.encryption=true -Dspring.profiles.active=local -Dmosip\_host=https://<server hostname> --add-opens java.xml/jdk.xml.internal=ALL-UNNAMED --add-opens java.base/java.lang.reflect=ALL-UNNAMED --add-opens java.base/java.lang.stream=ALL-UNNAMED --add-opens java.base/java.time=ALL-UNNAMED --add-opens java.base/java.time.LocalDate=ALL-UNNAMED --add-opens java.base/java.time.LocalDateTime=ALL-UNNAMED --add-opens java.base/java.io.Reader=ALL-UNNAMED --add-opens java.base/java.util.Optional=ALL-UNNAMED --add-opens java.base/java.time.LocalDateTime.date=ALL-UNNAMED -jar target/mock-abis-1.3.0-SNAPSHOT.jar`.

\* Add message directly to queue and view responses from mock ABIS

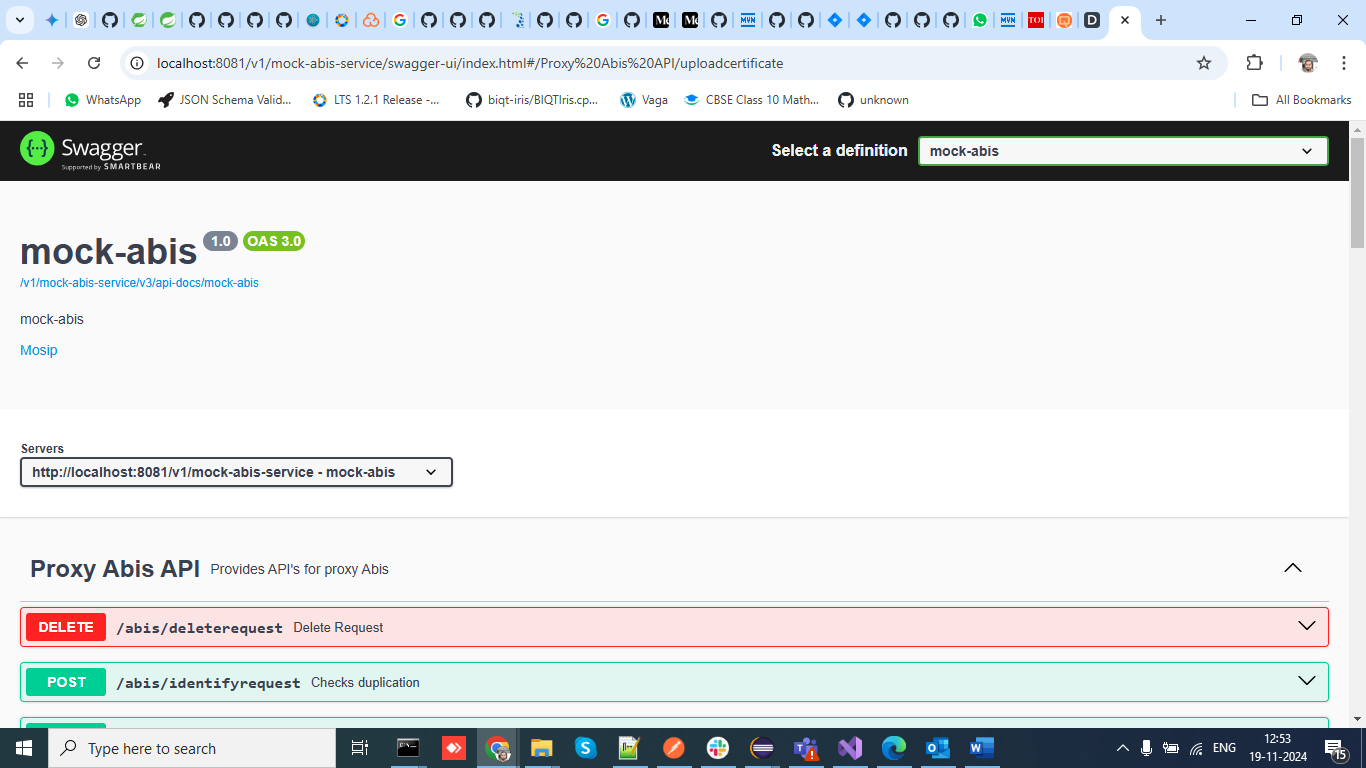
Flags:

\* local.development (true: whenever running locally)

\* abis.bio.encryption (true: in case if partner based encryption)

\* mosip\_host (hostname of the MOSIP server)

Swagger url: `http://localhost:8081/v1/mock-abis-service/swagger-ui/index.html#/`



**## APIs for configuration and expectation setting**

[Sample expectations](./docs/sampleExpectations.md)

**### Update configuration**

Url: http://{host}/v1/mock-abis-service/config/configure

Method: POST

Request:

```json

{

"findDuplicate": "false"

}

```

Response:

```text

Successfully updated the configuration

```

**### Get configuration**

Url: http://{host}/v1/mock-abis-service/config/configure

Method: GET

Response:

```json

{

"findDuplicate": false

}

```

**### Set Expectation**

Url: http://{host}/v1/mock-abis-service/config/expectation

Method: POST

Request:

```json

{

"id": "<Hash of the biometric>",

"version": "xxxxx",

"requesttime": "2021-05-05T05:44:58.525Z",

"actionToInterfere": "Identify/ Insert",

"forcedResponse": "Error",

"errorCode": "",

"delayInExecution": "",

"gallery": {

"referenceIds": [

{

"referenceId": "<Hash of the duplicate biometric>"

}

]

}

}

```

Response:

```text

Successfully inserted expectation $expectation\_id

```

**### Get Expectations**

Url: http://{host}/v1/mock-abis-service/config/expectation

Method: GET

Response:

```json

{

"abshd": {

"id": "abshd",

"version": "xxxxx",

"requesttime": "2021-05-05T05:44:58.525Z",

"actionToInterfere": "Identify/ Insert",

"errorCode": "",

"delayInExecution": "",

"forcedResponse": "Error/Success",

"gallery": {

"referenceIds": [

{

"referenceId": "xxxxxx"

},

{

"referenceId": "xxxxxx"

}

]

}

},

"dffefe": {

"id": "dffefe",

"version": "xxxxx",

"requesttime": "2021-05-05T05:44:58.525Z",

"actionToInterfere": "Identify/ Insert",

"forcedResponse": "Error/Success",

"errorCode": "",

"delayInExecution": "",

"gallery": {

"referenceIds": [

{

"referenceId": "xxxx"

},

{

"referenceId": "xxxxxx"

}

]

}

}

}

```

**### Delete Expectation**

Url: http://{host}/v1/mock-abis-service/config/expectation/{id}

Method: DELETE

Response:

```text

Successfully deleted expectation $expectation\_id

```

**## Tips & tricks**

1) While setting the expectation the hash of iso image should be taken, directly taking bdb hash will not work.

```text

formula for hash: SHA256\_hash(base64\_decode(bdb))

```

2) Use get cached biometics to check whether the hashes are proper.

**## Developer (tips and trick)**

This section is for the developers, for developing this modules fast & efficiently

1) Use local profile: `-Dspring.profiles.active=local`. Pass this as VM options

2) Pass: `mosip\_host=https://<mosip host>` as env variable.

3) Setting ABIS queue conf:

\* Create registration-processor-abis.json in resources

\* Copy the contents of registration-processor-abis-sample.json to registration-processor-abis.json

\* Update registration-processor-abis.json with the correct queue details

By performing the above steps, you are ready to run mock-ABIS in local machine

**## APIs**

API documentation is available [here](https://mosip.github.io/documentation/).

**## License**

This project is licensed under the terms of [Mozilla Public License 2.0](LICENSE).

D:\Project\Mosip\test>git clone https://github.com/mosip/mosip-mock-services.git

Cloning into 'mosip-mock-services'...

remote: Enumerating objects: 10666, done.

remote: Counting objects: 100% (3214/3214), done.

remote: Compressing objects: 100% (1190/1190), done.

remote: Total 10666 (delta 1663), reused 2835 (delta 1488), pack-reused 7452 (from 1)Receiving objects: 100% (10666/10666), 149.56 MiB | 6.67 MiB/s

Receiving objects: 100% (10666/10666), 150.17 MiB | 6.73 MiB/s, done.

Resolving deltas: 100% (3218/3218), done.

Updating files: 100% (444/444), done.

D:\Project\Mosip\test>cd mosip-mock-services

D:\Project\Mosip\test\mosip-mock-services>cd mock-abis

D:\Project\Mosip\test\mosip-mock-services\mock-abis>mvn clean install -Dgpg.skip=true