RaspberryWallet / **AuthorizationServer**

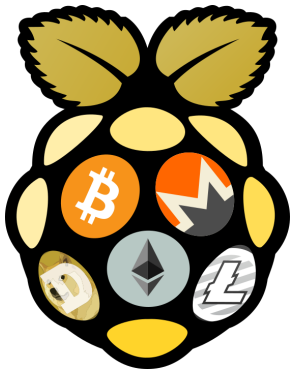
Authorization Server that will be used to authorize user's transactions and optionally other security modules.

Manage topics

24 commits | 2 branches | 0 releases | 2 contributors

README.md

# AuthorizationServer Wallet Module



This module is simple REST API application, that is used by Raspberry Wallet as Module, which is trusted third party side.

## Requirements

- Maven 3+
- Java 8

## Installation

```
# get repo
git clone https://github.com/RaspberryWallet/Backend.git
cd Backend

# build and install jars, then build docker image with Fabric8 plugin
mvn clean install docker:build

# after we made docker images, we can just turn on the system
docker-compose up -d
```

## Table of contents



## Details

This module uses Redis as data storage and Jedis library for connection. REST endpoints are provided by Spring Boot.

After a new version release, you have to change jar file name manually, since it's hardcoded into Dockerfile and there is no Dockerfile generator.

## Endpoints

Endpoints documentation is not actual and will be not updated, since the goal is to introduce Spring REST Docs.

Login is done by custom token system implementation.

#### Register

```
POST /authorization/register
Content-Type: application/json; charset=UTF-8
{
        "walletUUID":"abcd",
        "password":"1234"
}
```

#### Login

```
POST /authorization/login
Content-Type: application/json; charset=UTF-8
{
        "walletUUID":"abcd",
        "password":"1234"
}

# returns token which is UUID converted to String
```

#### Logout

```
POST /authorization/logout
Content-Type: application/json; charset=UTF-8
{
        "walletUUID":"abcd",
        "password":"1234"
}
```

#### Set secret

```
POST /authorization/secret/set
Content-Type: application/json; charset=UTF-8
{
        "walletUUID":"abcd",
        "token":"01badf5d-fb41-4d2a-a029-0bce54bea501",
        "secret":"data"
}
```

#### Get secret

```
POST /authorization/secret/get
Content-Type: application/json; charset=UTF-8
{
        "walletUUID":"abcd",
        "token":"01badf5d-fb41-4d2a-a029-0bce54bea501"

}

# returns secret as String
```

#### Overwrites existing secret

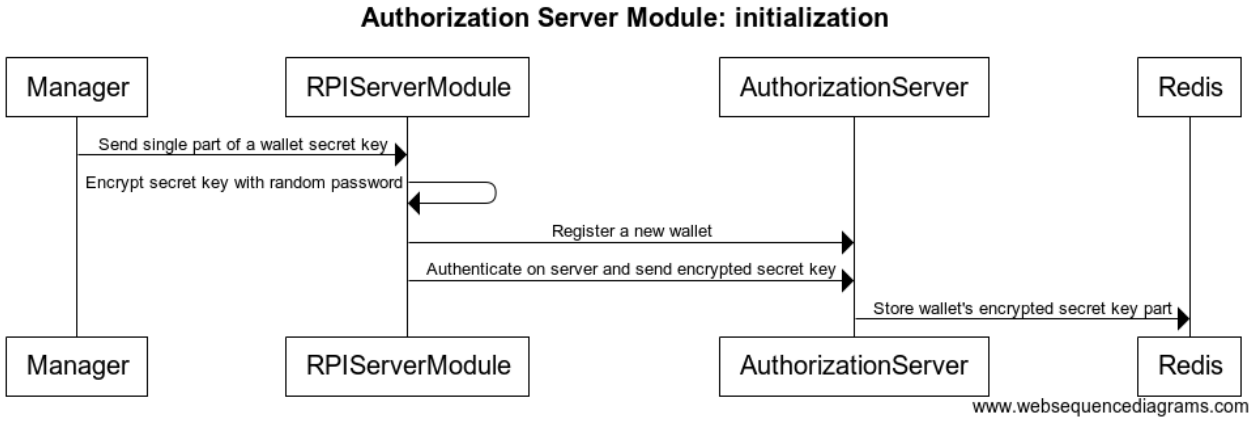
```
POST /authorization/secret/overwrite
Content-Type: application/json; charset=UTF-8
{
        "walletUUID":"abcd",
```

```
        "token":"01badf5d-fb41-4d2a-a029-0bce54bea501",
        "secret":"duplicate"
}
```

## Old sequence diagrams

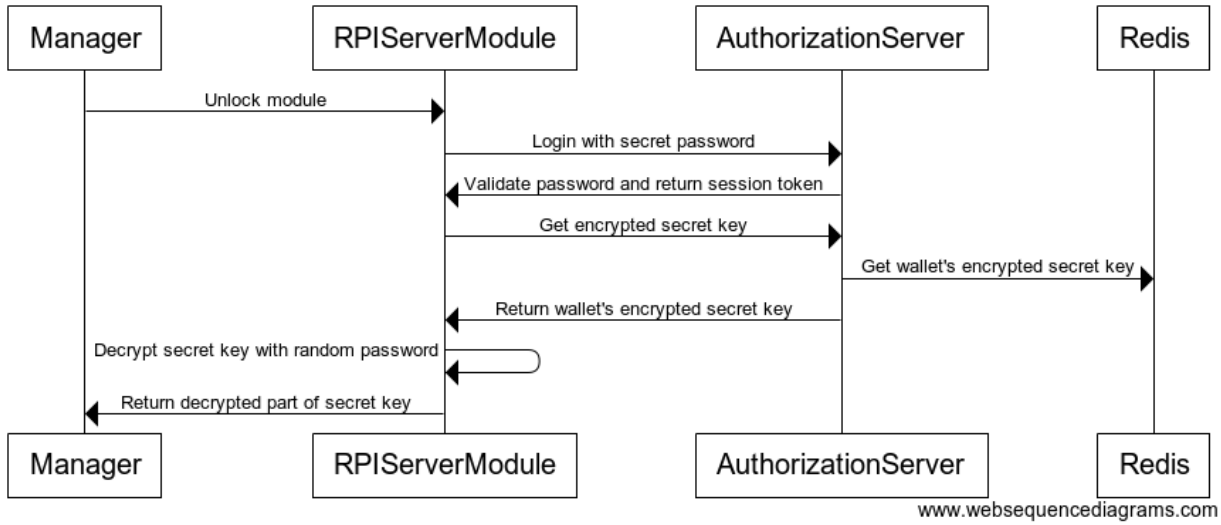
They are not actual and we left them here, because they will be refreshed soon.

**Initialization of module sequence diagram:**




▶ Sequence diagram code

**Unlock module sequence diagram:**




▶ Sequence diagram code

## Authors

| Name | email |
| --- | --- |
| Patryk Milewski | patryk.milewski@gmail.com |

## Changelog

| Version | Is backward-compatible | Changes | Commit ID |
| --- | --- | --- | --- |
| 0.3 | Yes | Working version | 3b73c7c2cd815a07972e5aee06e7a3d9f45d9dc7 |