Staff Engineering - Level 6 | AstroPay – SOLUTION

Description

The following graphical documentation details the proposed solution to manage the activities performed by users within Astropay's transaction system. The primary objective is to provide a robust solution capable of supporting millions of transactions and scalable over time.

Components Software

The following software components that were used for the solution:

- Zuul Api Gateway
- Eureka (Service Discovery)
- Oauth2
- Springboot
- H2 (memory DB)
- JPA

Components Diagram

The solution was designed in order to allow the data corresponding to different Payment, Deposit, P2P Transfers events to be stored in a JMS queue to avoid penalizing transaction time. Simultaneously, the JMS queue is consumed by a component that inserts the data into the activity microservice database with the following record design:

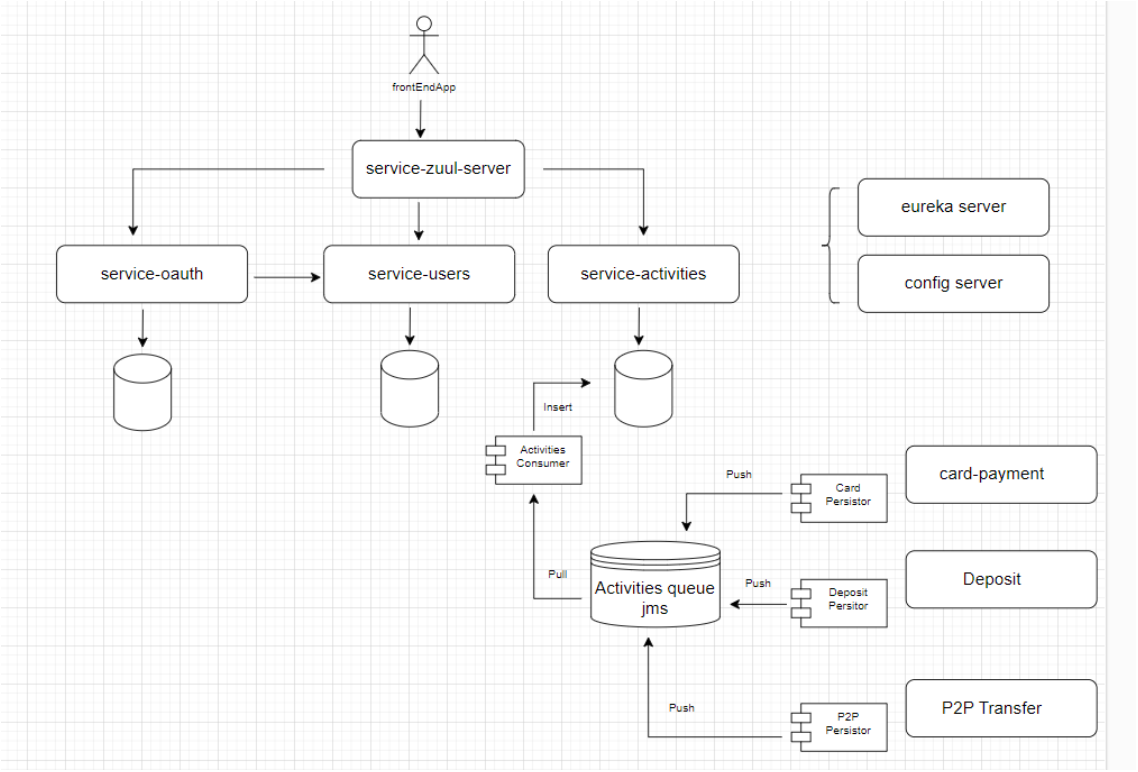
Field	Type
user_id	INTEGER
activity_id	STRING
activity_type	STRING
data_json	JSON
create_at	DATE

The architecture was conceived with an API Gateway serving as the entry point and load balancer. The OAuth2 component is responsible for generating and validating tokens, which will be use to call resources of the 'springboot-service-activities' microservice.

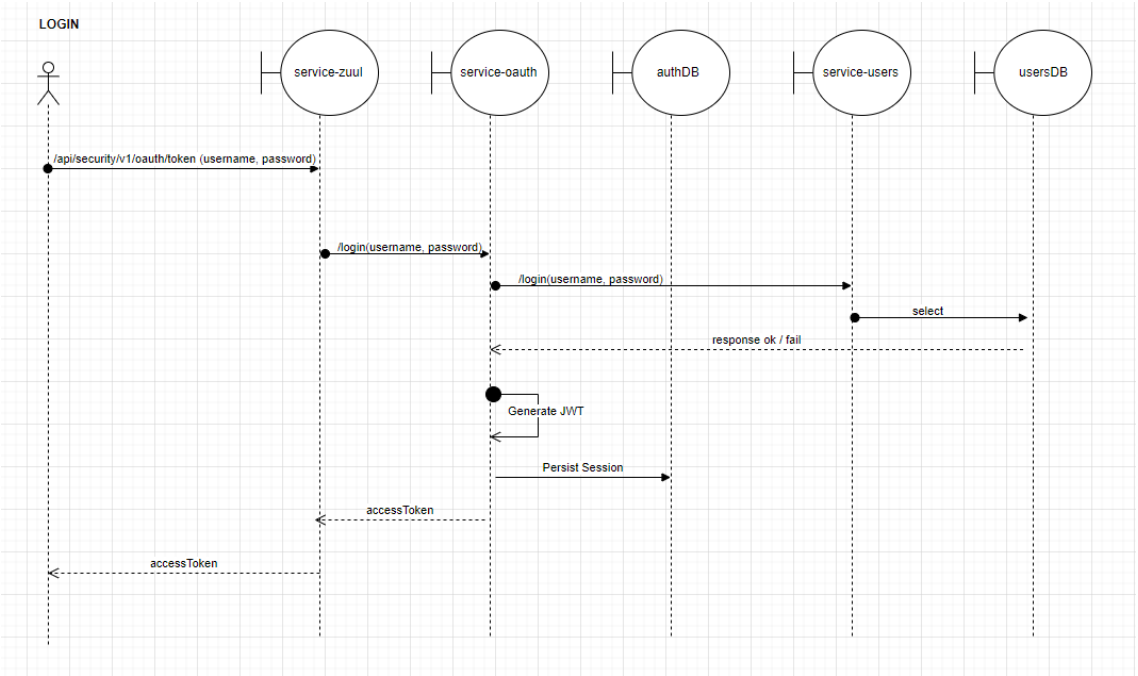
List of components in the solution:

- springboot-service-zuul-server
- springboot-service-eureka-server
- springboot-service-oauth
- springboot-service-users
- springboot-service-config-server

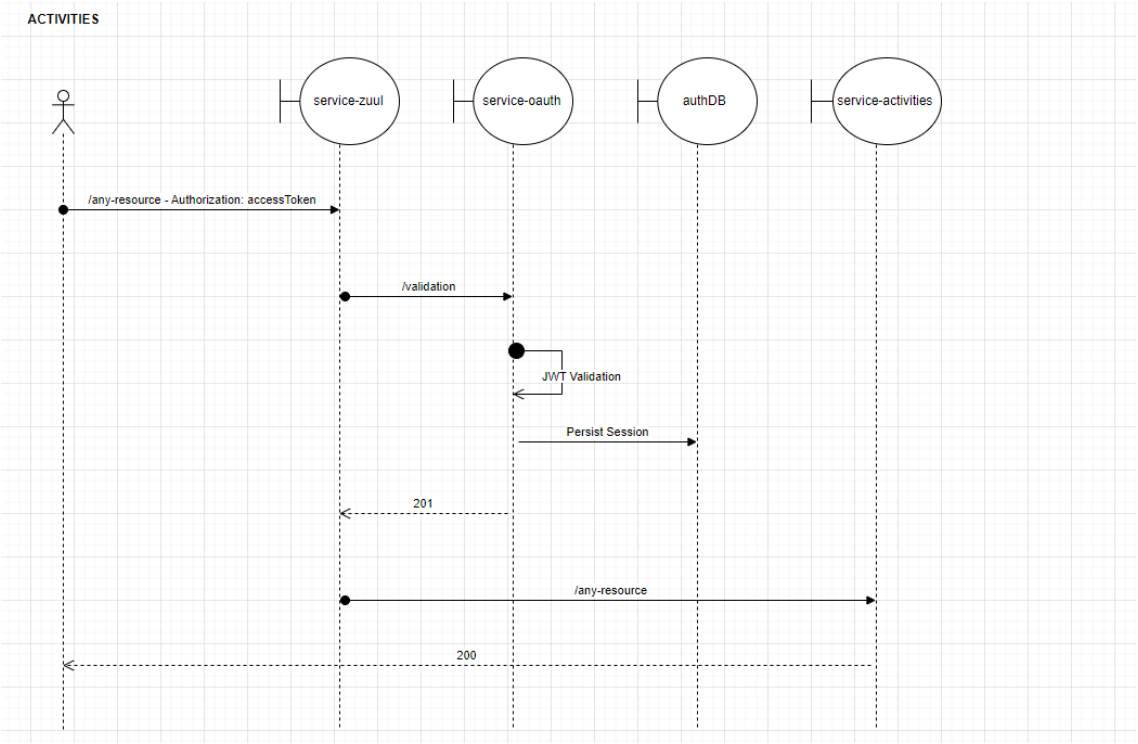
- springboot-service-activities



Sequences Diagram (Login)



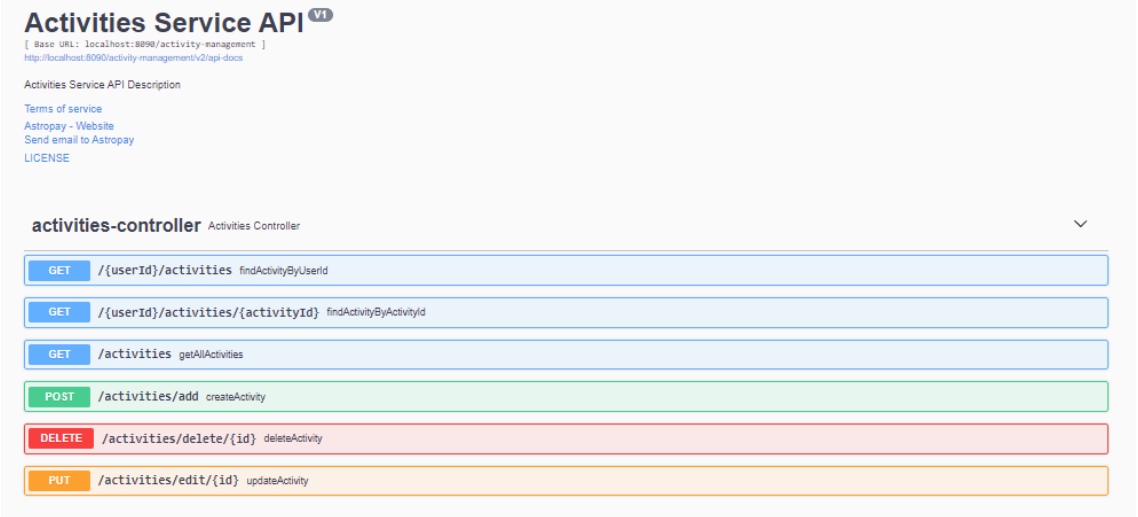
Sequences Diagram (Activities)



APPENDIX

Api Documentation

http://localhost:8090/activity-management/swagger-ui/#/activities-controller



Test Users

USERNAME	PASSWORD	ROLE
frontendapp	12345	
user	12345	ROLE_USER
admin	12345	ROLE_ADMIN

Access Users

METHOD	RESOURCE	ROLE
POST	/api/security/v1/oauth/token	permitAll()
GET	/api/activity-management/v1/activities	ROLE_ADMIN
GET	/api/activity-management/v1/{userId}/activities	ROLE_ADMIN, ROLE_USER
GET	/api/activity-management/v1/{userId}/activities/{activityId}	ROLE_ADMIN, ROLE_USER
POST	/api/activity-management/v1/activities/add	ROLE_ADMIN
DELETE	/api/activity-management/v1/activities/delete/{userId}	ROLE_ADMIN
GET	/api/activity-management/v1/actuator/health	permitAll()

Postman Collection

To test the API, in postman import Project from [folder]\activities-solution\Postman Collection\