XLSpacehsip Guide

[V 1.0.0]

Simon Atta

Table of Contents

Introduction

Purpose:

This document was created to help understand architecture, design, implementation and setup of XLSpaceship game.

Requirments:

1. XLSpaceship

- XLSpaceship is simulation for spaceship which will be container for fighter spaceships.
- XLSpaceship application can N of XLSpaceships which will be running on different addresses. Every
- XLSpacehsip will managed by single user also can be engaged in multiple games.
- XLSpaceship has spaceships board which will contains all spaceships.

2. Spaceship

- Spaceship is subset of XLSpaceship.
- Spaceship has shape, name and size.
- Spaceship placement will be done by player.
- Spaceship will be unique per board and can’t be overlapped.

3. Game

- Game consist of two XLSpaceships fighting together till one of them get die.
- Game has many status created, started and finished.
- Game will be in progress until last salvo shot will kill any board of the players.

Technologies:

1. Java 8
2. Spring boot full stack.
3. Swagger
4. Mysql
5. Thymeleaf
6. Bootstrap, JQuery
7. Maven

Platforms:

1. Docker
2. Azure Cloud

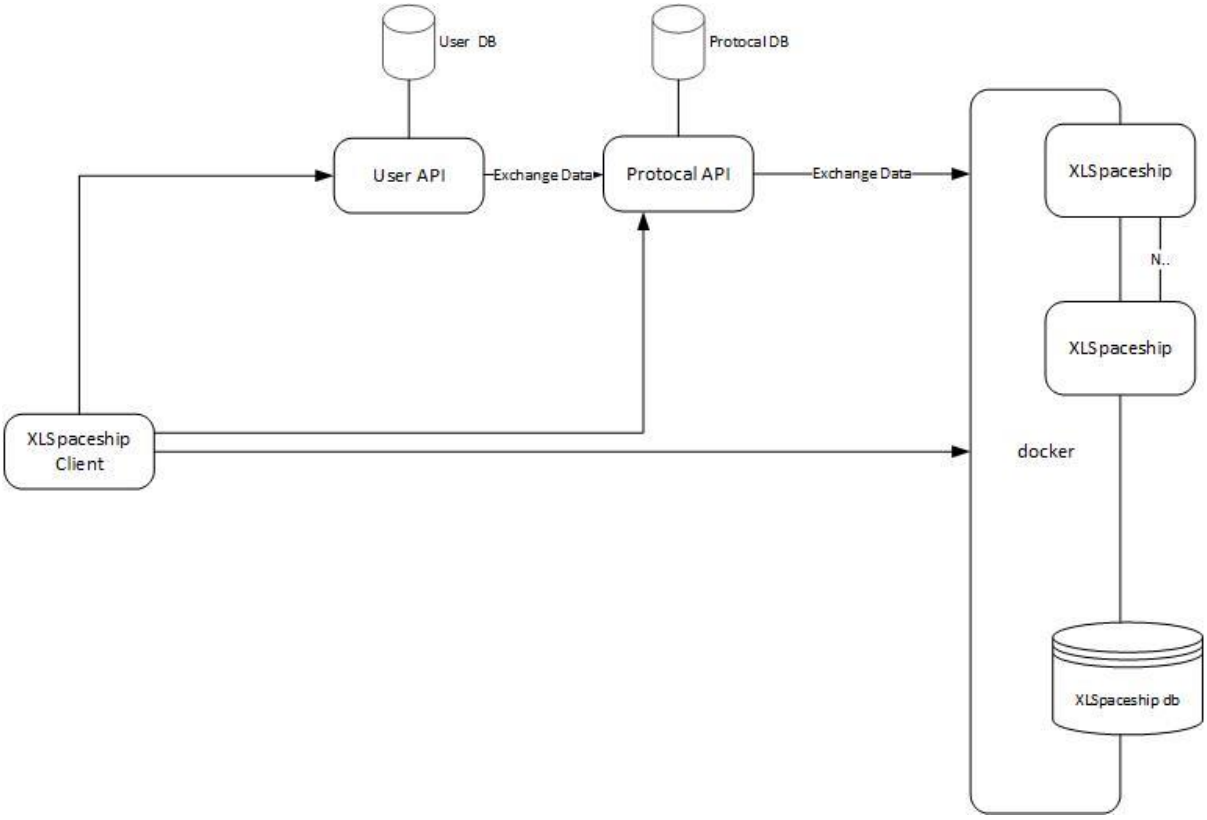
Architecture:

XLSpaceship design is based on micro services architecture. This decision have been taken based on requirement mentioned in the document.

API's divided based on domain e.g. User, XLSpaceship, Game each of this domains has its own API.

Each API has its own SDK so all other client can call web services.

Each API has its own database.



1. XLSpaceship

- XLSpaceship application which handle XLSpaceship instances.
- XLSpaceship expose many endpoint to handle XLSpaceship, spaceship management.
- XLSpaceship application is containerized by docker, so it will be easing creating many XLSpaceships.
- XLSpacehsip instance of will be running on specific address.
- XLSpaceship will has record in database specify address as lookup table.

Point of enhancement:

- Extract all admin endpoints for XLSpaceship into new application in order to separate concern.
- Create XLSpaceship in programmable by expose docker API to create containers on demand.

2. Protocal

- Protocol application responsible for management games and communication between XLSpaceships.

3. User

- User application responsible for management user.
- User application also will be responsible for handle fire between users and view status of games.

4. XLSpaceship Client

- XLSpaceship client application is web application to manage the following:
    - Users:
        - Create users.
        - Login into XLSpaceship game platform.
    - Spaceships:
        - Assign XLSpaceship to user.
        - Manage spaceships into XLSpaceship board.
    - Game:
        - Create new game
        - View game status and play games.

Design:

All applications have same design N Tier layering.

All application communicate between each other's through SDK's.

Implmentation:

All operation against DB's are transactional.

All layers throw business exceptions with proper error code will be handled to controllers.

1. XLSpaceship

- XLSpaceship at bootstrap level its initialize static instance in memory for spaceship board and load all spaceship configuration into memory.
- Spaceships configured at XML with all details e.g. name, dimensions and shape.
- In order to communicate with board instance there is factory which always return this instance and make sure that there is no duplication.

Point of enhancement:

- When XLSpaceship is shut down for any reason should update all decencies for this instance.

2. Web client

Here is all business flows are covered and discovered bugs:

a. Create user

b. Login user

    i. There is only one bug after login redirect to error page will be fixed in next batch.

c. List all users in order to see user can play with.

d. List all XLSpaceship in order to assign XLSpaceship to himself.

e. List all my XLSpaceships in order to view and start place spaceships into XLSpaceship.

f. List all games will get all player game.

g. Player can view status of the game.

    i. Player can enable / disable auto pilot per game.

    ii. Player will be able to fire salvo shots if it’s his turn.

Setup & Code Repo:

- Source code is managed by private repo in this URL:
  https://hendaamvs.visualstudio.com/DefaultCollection/_git/Spaceship

  Username: s.samir@hendaam.com

  Password: Xebialabs64

- XLSpaceship latest version is deployed into cloud VM with the following specification
  - 3 XLSpaceship instances.
  - User API
  - Protocol API
  - Web Client
  - Here is all urls:
    - XLSpaceship no.1 http://spaceship.westeurope.cloudapp.azure.com:8080/swagger-ui.html
    - XLSpaceship no.2 http://spaceship.westeurope.cloudapp.azure.com:8081/swagger-ui.html
    - XLSpaceship no.3 http://spaceship.westeurope.cloudapp.azure.com:8082/swagger-ui.html
    - Protocol API http://spaceship.westeurope.cloudapp.azure.com:8083/swagger-ui.html
    - User API http://spaceship.westeurope.cloudapp.azure.com:8084/swagger-ui.html
    - Web client : http://spaceship.westeurope.cloudapp.azure.com:8087/

Environment Setup:

Requisites:

1- MySQL

2- STS

3- Docker latest version

4- Maven

5- Java 8

Steps:

1- Checkout code from repo

2- Create db schema as specified in application.properties for each application.

3- Before maven install enable these properties:

a. `spring.jpa.generate-ddl=true`
b. `spring.jpa.hibernate.ddl-auto=create`

4- Clean & install projects

5-Build docker image for XLSpaceship application by running the following command

a. mvn package docker:build

6-Then runthe following projects by right click run as spring boot

a. User

b. Protocal

c. Client

7-Then run XLspaceship containers by the following:

a. Open bash

b. Then go to path of this project XL-spaceship

c. Then run the following command

i. docker-compose up this command will boot up mysql instance and 3 XLspaceships.