***OLX Advertisement Application***

**Introduction:**

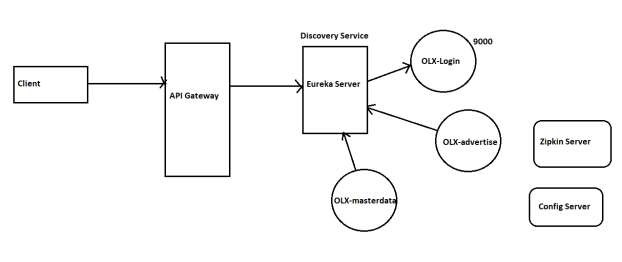
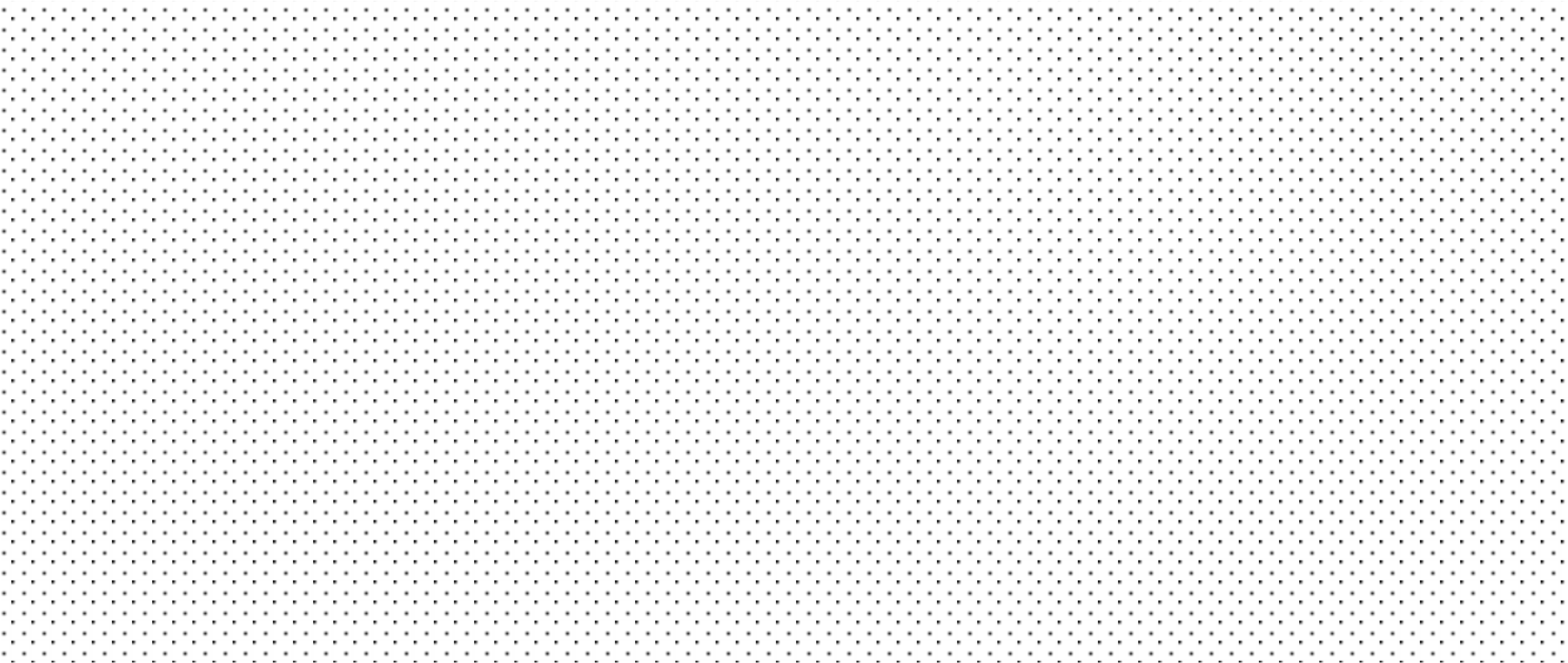
OLX Advertisement application is used for posting online advertisements. This is a web portal useful commodity buyer & seller. The server-side development is done using Spring Boot framework & it follows Microservice Architecture.

**Software Installation:**

Please make sure you have installed below softwares:

1. JDK 1.8 or above
2. Spring Tool Suite(STS) 4.x
3. Tomcat 8.x or above
4. MongoDB 4.x or above
5. Mongo Compass - UI tool (optional)
6. MySQL (preferably 8.x)
7. Postman (REST client)
8. Maven 3.x or above
9. GIT 2.x
10. Jenkins
11. SonarQube
12. RabbitMQ Server
13. ERLANG
14. Docker Desktop

**Architecture diagram:**



**REST API specifications:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Sr. No.** | **URL +**  **METHOD** | **Functionality** | **Header** | **Request body** | **Response** | **Query string** | **Comments** |
|  |  |  |  |  |  |  |  |
| 1 | /user/authenticate POST | Logins a user |  | {"userName": "anand", "password": "anand123"} | auth-token |  |  |
| 2 | /user/logout DELETE | Logs out a user | auth-token |  | true/false |  |  |
| 3 | /user POST | Registers an user |  | { "firstName": "Anand", "lastName": "Kulkarni", "userName": "anand", "password": "anand123", "email": "anand@gmail.com", "phone": 12345  } | {"id": 1, "firstName": "Anand", "lastName": "Kulkarni", "userName": "anand", "password": "anand123", "email": "anand@gmail.com", "phone": 12345  } |  |  |
| 4 | /user GET | Returns user information | auth-token |  | {"id": 1, "firstName": "Anand", "lastName": "Kulkarni", "userName": "anand", "password": "anand123", "email": "anand@gmail.com", "phone": 12345  } |  |  |
| 5 | /token/validate GET | Validates the client’s token | auth-token |  | True/false |  | This service is called by every REST endpoint that acceots auth-toekn in Header |
| 6 | /advertise/category GET | Returns all advertisement categories |  |  | {"categories": [{id:1, category: "Furniture"}, {id:2, category: "Cars", {id:3, category: "Mobiles"}, {id:4, category: "Real Estate"}, {id:5, category:"Sports"}]} |  | Advertisement categories means 'Real Estate', 'Furniture', 'Electronic goods', 'Vehicles' etc. |
| 7 | /advertise/status GET | Returns all possible advertise status |  |  | {"statusList": [{"id": 1, "status": "OPEN"}, {"id": 2, "status": "CLOSED"}]} |  |  |
| 8 | /advertise POST | Posts new advertise | auth-token | {"title": "laptop sale", "price": 54000, "categoryId": 1, "description": "intel core 3 Sony Vaio"} | {"id": 1, "title": "laptop sale", "price": 54000, "category": "Electronic goods", "description": "intel core 3 Sony Vaio", "username": "anand", "createdDate": xxx, "modifiedDate": xxx, "status": "OPEN"} |  |  |
| 9 | /advertise/{id} PUT | Updates existing advertise | auth-token | {"title": "Sofa available for Sale", "price": 30000, "categoryId": 3, "statusId": 2, "description": "Sofa 5 years old available for Sale in Pune"} | {"id": 2, "title": "Sofa available for sale", "price": 30000, "category": "Furniture", "description": "Sofa 5 years old available for Sale in Pune", "username": "anand", "createdDate": xxx, "modifiedDate": xxx, "status": "OPEN"} |  |  |
| 10 | /user/advertise GET | Reads all advertisements posted by logged in user | auth-token | NA | {"advertises": [ {"id": 1, "title": "laptop sale", "price": 54000, "category": "Electronic goods", "description": "intel core 3 Sony Vaio", "createdDate": xxx, "modifiedDate": xxx, "status": "OPEN"}, {"id": 2, "title": "Sofa available for sale", "price": 30000, "category": "Furniture", "description": "Sofa 5 years old available for Sale in Pune", "createdDate": xxx, "modifiedDate": xxx, "status": "OPEN"} ]} |  |  |
| 11 | /user/advertise/{advertiseId} GET | Reads specific advertisement posed by logged in user | auth-token | NA | {"id": 1, "title": "laptop sale", "price": 54000, "category": "Electronic goods", "description": "intel core 3 Sony Vaio", "createdDate": xxx, "modifiedDate": xxx, "status": "OPEN"} |  |  |
| 12 | /user/advertise/{advertiseId} DELETE | Deletes specific advertisement posted by logged in user | auth-token |  | true/false |  |  |
| 13 | /advertise/search/filtercriteria GET | Search advertisements based upon given filter criteria |  | NA | {"advertises": [ {"id": 1, "title": "laptop sale", "price": 54000, "category": "Electronic goods", "description": "intel core 3 Sony Vaio", "createdDate": xxx, "modifiedDate": xxx, "status": "OPEN"}, {"id": 2, "title": "Sofa available for sale", "price": 30000, "category": "Furniture", "description": "Sofa 5 years old available for Sale in Pune", "createdDate": xxx, "modifiedDate": xxx, "status": "OPEN"} ]} | searchText,category, postedBy, dateCondition, onDate, fromDate, toDate, sortBy, startIndex, records | Note: this advertise search does not require authenticated user. |
| 14 | /advertise/search GET | Matches advertisements using the peovided 'searchText' within all fields of an advertise. |  | NA | {"advertises": [ {"id": 1, "title": "laptop sale", "price": 54000, "category": "Electronic goods", "description": "intel core 3 Sony Vaio", "createdDate": xxx, "modifiedDate": xxx, "status": "OPEN"}, {"id": 2, "title": "Sofa available for sale", "price": 30000, "category": "Furniture", "description": "Sofa 5 years old available for Sale in Pune", "createdDate": xxx, "modifiedDate": xxx, "status": "OPEN"} ]} | searchText | Note: this advertise search does not require authenticated user. |
| 15 | /advertise/{advertiseId} GET | Return advertise details | auth-token | NA | {"id": 1, "title": "laptop sale", "price": 54000, "category": "Electronic goods", "description": "intel core 3 Sony Vaio", "createdDate": xxx, "modifiedDate": xxx, "status": "OPEN", "postedBy": "Anand Kulkarni"} |  |  |

In the service 12, understanding on datecondition with possible values in the URI:

datcondition=equals&onDate=2021-10-17

datcondition=greatethan&fromDate=2021-11-08

datcondition=lessthan&fromDate=2021-11-08

datcondition=between&fromDate=2021-10-17&toDate=2021-11-08

**Database design:**

In the Microservice architecture, generally we create separate database for every microservice. In OLX app, we have 3 Microservices configured & hence we will have 3 databases needs to be created. Please find below the tables along with their CREATE script. Note that developer is free to give different database names, table names, column names and add new columns.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sr. No.** | **Database Name** | | **Table/s** | **Create Table Script** |
| 1. | olx-users | USERS | | CREATE TABLE `USERS` (  `id` int NOT NULL,  `username` varchar(20) NOT NULL,  `password` varchar(20) NOT NULL,  `roles` varchar(50) NOT NULL,  `active` enum('true','false') NOT NULL,  `firstname` varchar(45) DEFAULT NULL,  `lastname` varchar(45) DEFAULT NULL,  PRIMARY KEY (`id`)  ) |
| 2. | olx-masterdata | CATEGORIES, ADV\_STATUS | | CREATE TABLE `CATEGORIES` (  `id` int NOT NULL,  `name` varchar(50) NOT NULL,  `description` varchar(100) NOT NULL,  PRIMARY KEY (`id`)  )  CREATE TABLE `ADV\_STATUS` (  `id` int NOT NULL,  `status` varchar(50) NOT NULL,  PRIMARY KEY (`id`)  ) |
| 3. | olx-advertises | ADVERTISES | | CREATE TABLE `ADVERTISES` (  `id` int NOT NULL,  `title` varchar(50) NOT NULL,  `category` int NOT NULL,  `status` int NOT NULL,  `price` double(10,2) DEFAULT NULL,  `description` varchar(100) NOT NULL,  `photo` blob,  `created\_date` date DEFAULT NULL,  `modified\_date` date DEFAULT NULL,  `active` enum('0','1') DEFAULT NULL,  `posted\_by` varchar(45) DEFAULT NULL,  `username` varchar(255) DEFAULT NULL,  PRIMARY KEY (`id`)  ) |

**Micro-services design:**

Note that every microservice is a separate Spring Boot project. Please create Spring Boot projects as per below guidelines-

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr. No.** | **Microservice Name** | **Database Name** | **REST Endpoints from specification** |
| 1. | OLX-Login | olx-users | Nos 1 to 5 |
| 2. | OLX-Masterdata | olx-masterdata | Nos 6 & 7 |
| 3. | OLX-Advertises | olx-advertises | Nos 8 to 15 |

**Code submission**

Participants needs to push the final code into GITHUB repository using personal GIT account & share the GIT URL with us for code review.