Тестовое задание для JAVA EE Developer

Необходимо реализовать серверное приложение осуществляющее операцию аутентификации пользователя.

Взаимодействие клиент/сервер происходит по протоколу webSocket(http://ru.wikipedia.org/wiki/WebSocket). Запросы к серверу и ответы от него передаются в формате JSON(http://ru.wikipedia.org/wiki/JSON).

Приложение должно обеспечивать следующие процессы:

— аутентификацию пользователя

— приложение должно принимать запрос LOGIN\_CUSTOMER проверять наличие данного пользователя в базе с указанным паролем и возвращать либо сообщение об успешно аутентификации с токеном и датой его истечения, либо сообщение об ошибке

— хранение токенов пользователей

— приложение должно вести базу выданных пользователям токенов.

— приложение должно вести историю токенов пользователей.

— приложение должно сбрасывать токен пользователя если он запрашивает повторную аутентификацию.

Описание протокола

**Описание структуры исходящего сообщения:**

{

“type”: “TYPE\_OF\_MESSAGE” , // строка, тип сообщения

“sequence\_id”: “09caaa73-b2b1-187e-2b24-683550a49b23”, // строка, идентификатор связанности сообщений

“data” : {} // объект, содержит данные запроса

}

**Описание структуры входящего сообщения:**

{

“type”: “TYPE\_OF\_MESSAGE” , // строка, тип сообщения

“sequence\_id”: “09caaa73-b2b1-187e-2b24-683550a49b23”, // строка, идентификатор связанности сообщений

“data” : {} // объект, содержит данные ответа

}

**В случае возврата сообщения об ошибке, объект data должен содержать следующие поля:**

"error\_description":"Customer not found”, // поле с описанием ошибки

"error\_code":"customer.notFound” // поле с кодом ошибки

**Запрос для успешной операции аутентификации:**

{

"type":"LOGIN\_CUSTOMER”,

"sequence\_id":"a29e4fd0-581d-e06b-c837-4f5f4be7dd18”,

"data”:{

"email":"fpi@bk.ru”,

"password":”123123"

}

}

**Ответ для успешной операции аутентификации:**

{

"type":"CUSTOMER\_API\_TOKEN”,

"sequence\_id":"cbf187c9-8679-0359-eb3d-c3211ee51a15”,

"data”:{

"api\_token":"afdd312c-3d2a-45ee-aa61-468aba3397f3”,

"api\_token\_expiration\_date":"2015-07-15T11:14:30Z”

}

}

**Запрос для не успешной операции аутентификации:**

{

"type":"LOGIN\_CUSTOMER",

"sequence\_id":"715c13b3-881a-9c97-b853-10be585a9747”,

"data”:{

"email":"123@gmail.com”,

"password":”newPassword"

}

}

**Ответ для не успешной операции аутентификации:**

{

"type":"CUSTOMER\_ERROR”,

"sequence\_id":"715c13b3-881a-9c97-b853-10be585a9747”,

"data”:{

"error\_description":"Customer not found”,

"error\_code":"customer.notFound”

}

}

Разрешается использовать любые фреймворки, а так же maven и gradle.

Application Server на выбор, предпочтение отдается wildfly.

Проект необходимо предоставить целиком.

**Необходимо самостоятельно организовать:**

— поддержку связанности сообщений, в описании протокола средства для этого заложены

— обработку ошибок;

— работу с БД;

— по необходимости работу с конфигурационными файлами;

— по необходимости работу с WebSocket сессиями

— приложение должно быть готово к работе большого числа пользователей через разные WebSocket сессии.