

Тестовое задание для iOS Developer

Необходимо реализовать приложение реализующее операцию аутентификации пользователя на сервере.

Приложение должно содержать два экрана:

- экран ввода электронной почты и пароля пользователя (далее – экран входа), должен содержать поля ввода электронной почты (с вариацией корректно введенного адреса), поле ввода пароля(должно маскировать вводимые символы), поле вывода информации об ошибках и кнопку «Login»;
- экран вывода информации об успешном выполнении операции аутентификации (далее – экран приложения), должен содержать поле вывода информации.

В качестве идентификатора аутентифицированного пользователя используется значение поля `api_token` из ответа сервера об успешной операции аутентификации(см. Описание протокола).

При запуске приложения необходимо проверять была ли произведена операция аутентификации ранее и не истек ли срок действия аутентифкации (поле `api_token_expiration_date` в ответе от сервера об успешной аутентификации (см. Описание протокола)).

Если вход был произведен ранее и срок действия `api_token` не истек, необходимо отображать экран приложения с датой истечения срока действия `api_token_expiration_date`. Если операция аутентификации не была выполнена ранее, либо срок действия `api_token` истек, необходимо отображать экран входа.

На экране входа пользователь должен вводить адрес электронной почты и пароль и после нажатия на кнопку «Login» выполнять запрос к серверу на выполнение операции аутентификации.

В случае успешного выполнения операции аутентификации необходимо вывести экран приложения с информацией об успешном входе и дату истечения срока действия(поле `api_token_expiration_date` из тела ответа от сервера) в поле вывода информации, в случае ошибки при выполнении операции аутентификации необходимо вывести информацию об ошибке(`error_description` из тела ответа от сервера) в поле вывода информации экрана входа.

Описание протокола

Взаимодействие с сервером 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>).

`WebSocket URL` сервера: `ws:// 52.29.182.220:8080/customer-gateway/customer`

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
}
```

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",
  "in_event": { "type": "LOGIN_CUSTOMER", "event_id": "c82803ef-157a-410f-bcab-85ea16d3a5b7", "sequence_id": "715c13b3-881a-9c97-b853-10be585a9747", "customer_id": "8ece7825-7ac6-4555-adb7-fa02647a0b70", "email": "123@gmail.com", "password": "newPassword"}, "error_uuid": "77b75372-ccb0-4fa3-94fb-e6577c882fb4"
}
```