**Типы сообщений клиента**

* При регистрации клиент посылает JSON следующего формата:

{ "MessageType": 0, "UserId": -1, "Login": "newUser", "Password": "pass" },

где "MessageType": 0 означает, что серверу необходимо выполнить регистрацию пользователя, поле UserId равно -1, так как пользователь ещё не зарегистрирован. Совпадение пароля с его подтверждением проверяется на стороне клиентского приложения.

* При авторизации клиент посылает JSON следующего формата:

 { "MessageType": 1, "UserId": -1, "Login": "Petrov PP", "Password": "pass1" },

где "MessageType": 1 означает, что серверу необходимо выполнить авторизацию пользователя, поле UserId равно -1, так как пользователь ещё не авторизован.

* При отправке данных о местоположении клиент посылает JSON следующего формата:

 { "MessageType": 2, "UserId": 1, "Time": "184107", "Status": "A", "Latitude": 1262.1500000000001, "NorthOrSouth": "N", "Longitude": 3507.3499999999999, "WestOrEast": "E", "Speed": 7.4199999999999999, "Direction": 37.109999999999999, "Date": "110720", "Declination": 3.71, "DecWestOrEast": "W", "ModeInd": "" },

где "MessageType": 2 означает, что серверу необходимо принять данные пользователя.

* При выходе из системы клиент посылает JSON следующего формата:

 { "MessageType": 3, "UserId": 1, "Login": "Petrov PP", "Password": "pass1" },

где "MessageType": 3 означает, что пользователь выходит из системы.

**Типы сообщений сервера**

* В случае, если пользователь успешно зарегистрирован или авторизован, сервер посылает JSON следующего формата:

{ "MessageType": 4, "UserId": 6, "Login": "newUser", "Password": "pass" },

где "MessageType": 4 означает, что операция прошла успешно. В сообщении клиенту отправляется уникальный id пользователя, который затем используется для проверки правильности клиента при получении от него координат.

* В случае, если сервер успешно получил данные клиента, он посылает JSON следующего формата:

{ "MessageType": 4 },

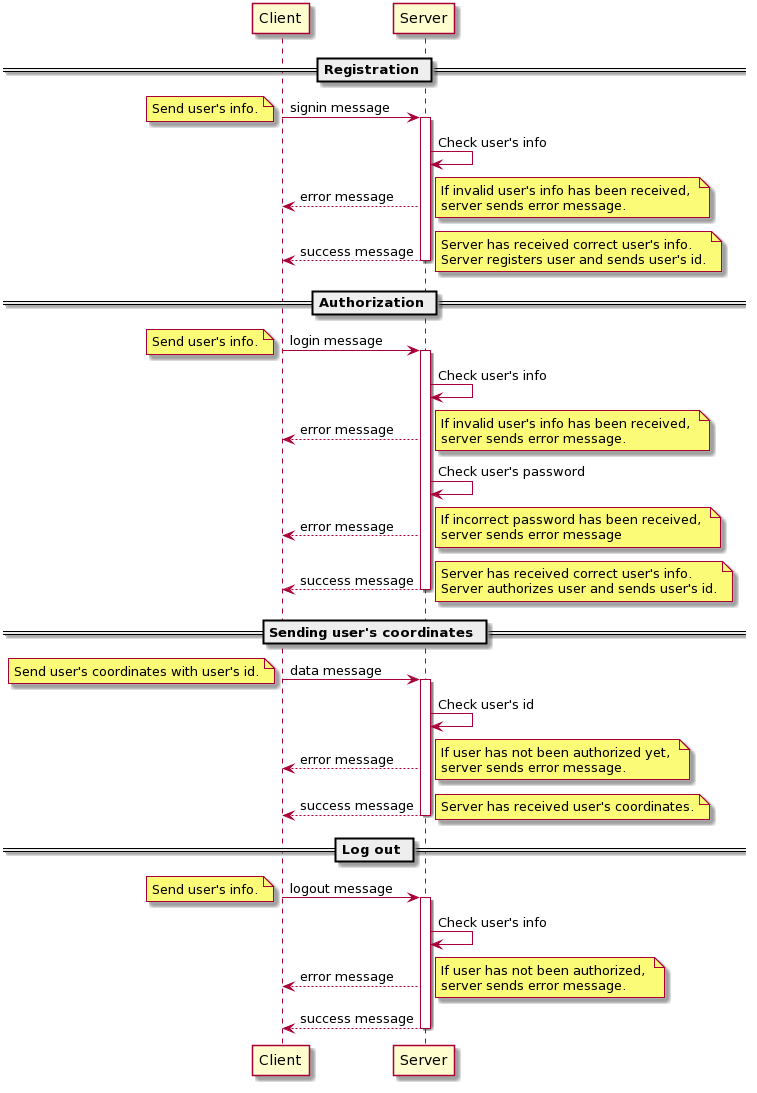
где "MessageType": 4 означает, что операция прошла успешно.

* В случае, если какая-либо операция закончилась ошибкой, сервер посылает JSON следующего формата:

{ "MessageType": 5, "Error": "Incorrect password" },

где "MessageType": 5 означает, что произошла ошибка. В поле Error содержится описание возникшей ошибки.

**Последовательность обмена сообщениями**



*Регистрация/авторизация*

1. Клиент формирует JSON с информацией о пользователе и отправляет на сервер.
2. Сервер получает сообщение клиента и проверяет данные. В случае, если данные пришли в неверном формате, или какие-то поля оказались незаполненными, или пользователь с таким именем уже существует (в случае регистрации), или пользователя с таким именем не существует (в случае авторизации), или пароль не совпадает с тем, который хранится в базе данных (в случае авторизации), то сервер посылает соответствующее сообщение об ошибке. В случае успешной регистрации/авторизации сервер посылает сообщение об успешном завершении операции, в котором передает уникальный id пользователя.
3. Клиент получает сообщение от сервера. В случае, если это сообщение об ошибке, то выводится информация, содержащаяся в поле Error, и клиент может исправить ошибку и повторить операцию заново.

Если регистрация/авторизация прошла успешно, то на клиентском приложении сохраняется уникальный id пользователя. Приложение начинает генерировать NMEA строки и, если включена отправка данных на сервер, отправлять координаты пользователя.

*Отправка координат пользователя на сервер*

1. Клиент формирует JSON с координатами пользователя и его уникальным id и отправляет на сервер.
2. Сервер получает сообщение клиента и проверяет данные. В случае, если полученный id пользователя не совпадает с id авторизованного клиента или переданные данные имеют неверный формат, то сервер посылает соответствующее сообщение об ошибке. В противном случае он отправляет сообщение об успешном завершении операции.
3. Клиент получает сообщение от сервера. В случае, если это сообщение об ошибке, то он повторно отправляет данные. Максимальное число повторов равно 3.

*Выход из системы*

1. Клиент формирует JSON с информацией о пользователе и отправляет на сервер.
2. Сервер получает сообщение клиента и проверяет данные. В случае, если полученный id пользователя не совпадает с id авторизованного клиента или переданные данные имеют неверный формат, то сервер посылает соответствующее сообщение об ошибке. В противном случае он помечает, что пользователь с полученным id больше не авторизован.
3. Клиент получает сообщение от сервера. В случае, если это сообщение об ошибке, то выводится информация, содержащаяся в поле Error.