Спецификации протокола mSIM

Мусатов М.Д. Солкин И.В.

28 октября 2012 г.

Глава 1

В целом

1.1 О протоколе mSIM

Что такое mSIM? Если в двух словах, то это лёгкий расширяемый протокол для обмена текстовыми сообщениями. mSIM был создан с прицелом на нетребовательность к ресурсам, масштабируемость и простоту реализации.

1.2 Суть протокола

Итак. В прокотоле mSIM имеются два главных действующих лица — клиент и сервер. Что характерно, клиент подключается к серверу через сокет с использованием протокола TCP (рекомендуемый порт — 3215). После этого начинается процесс обмена информацией, которая передаётся в виде пакетов.

Пакет mSIM по своей структуре очень прост. Он состоит из следующих друг за другом трёх строк в кодировке UTF-8 — эти строки называются полями. В начале каждой строки идут два байта, обозначающие её длину (в прямом порядке, например, 8 записывается как `0x00 0x08`), за ними — собственно значащие байты[1]. Названия и назначения полей приведены ниже.

1. `Destination`: Логин получателя (для исходящих пакетов) или отправителя (для входящих). Если пакет адресован самому серверу, здесь должно стоять имя `SERVER`.

[1]Больше всего повезло программистам на Java, потому что они могут прозрачно работать с такими строками через методы DataInputStream.readUTF() и DataOutputStream.writeUTF() для чтения и записи соответственно.

2. `Type`: Тип пакета.

3. `Content`: Содержимое пакета.

Поле `Content` содержит собственные поля, количество и назначение которых зависит от типа пакета. Эти поля отделяются друг от друга символом вертикальной черты: «|». Некоторые типы пакетов предусматривают передачу информации в формате INI.

Пакеты не имеют заголовков, маркеров и никак не отделяются друг от друга. Типы пакетов и информацию, которую они в себе несут, мы подробно рассмотрим далее.

Что-то ещё? Да, сервер разрывает соединение в случае, если клиент в течение 120 секунд не подаёт признаков жизни. Поэтому не забывайте периодически отправлять пинги (см. п. 2.3.1).

1.3 Технологии

Для реализации клиента mSIM точно потребуется поддержка следующих технологий:

- Сокеты
- Строки в UTF-8
- Разбиение строки на поля по заданному разделителю
- Парсер формата INI

Если говорить про реализацию на Java SE, то первые три возможности изначально встроены в язык, а для четвёртой можно использовать библиотеку BinGear, которая используется и в официальном клиенте mSIM.

Глава 2

В деталях

2.1 Соглашение об обозначениях

Обозначения обозначаются обозначениями в соответствии с тем, что они обозначают. Статью потом допилю.

2.2 Базовые типы пакетов

2.2.1 «Успех»

Пакет с типом `success` оповещает об успешном завершении операции. Как правило, в поле `Content` вписано её название.

2.2.2 «Неудача/отказ»

Пакет с типом `fail` оповещает о том, что запрошенную операцию по какой-то причине не удалось выполнить. Как правило, в поле `Content` вписано её название.

2.2.3 «Нет необходимости»

Пакет с типом `wtf` оповещает о том, что операция не была совершена, поскольку в ней нет необходимости. Как правило, в поле `Content` вписано её название.

2.2.4 «Внутренняя ошибка»

Пакет с типом `error-internal` оповещает о том, что операция не была закончена из-за неожиданного исключения. Поле `Content`, как правило,

пустое.

2.2.5 «Неизвестный тип пакета»

Пакет с типом `error-unknown-type` оповещает о том, что другой стороне незнаком этот тип пакета и он (пакет), вероятно, ошибочный. В поле `Content` вписан тип полученного пакета.

2.2.6 «Не поддерживается/ещё не реализовано»

Пакет с типом `error-not-implemented` оповещает о том, что операция не была выполнена из-за того, что другая сторона её не поддерживает. Как правило, в поле `Content` вписано её название.

2.3 Основные возможности протокола

2.3.1 Проверка связи

Используется для (сюрприз!) проверки связи и просто для поддержания соединения в активном состоянии.

Типичный случай использования:

```
<< SERVER: ping: «»
>> SERVER: ping-response: «»
```

2.3.2 Авторизация

Используется для авторизации на сервере.

Процедура в общем случае выглядит так:

```
<< SERVER: auth: «login|password»
>> SERVER: success: «auth»
```

Если пара логин-пароль неверна, сервер отвечает пакетом `fail`. Если клиент уже авторизован, сервер отправляет в ответ пакет `wtf`.

2.3.3 Регистрация

Используется для создания новой учётной записи на сервере.

Процедура в общем случае выглядит так:

```
<< SERVER: register: «login|password»
>> SERVER: success: «register»
```

Если такой аккаунт уже существует или по какой-то другой причине выполнить регистрацию невозможно, сервер отвечает пакетом `fail`.

2.4 Текстовые сообщения

2.4.1 Сообщение

Используется для отправки обычных текстовых сообщений адресату.

```
<< friend: message: «Привет!»
```

Факт доставки никак не подтверждается.

2.5 Работа со списком контактов

2.5.1 Запрос списка

Запрашивает у сервера список контактов. Ответ приходит в формате INI. Заголовкам соответствуют группы контактов, ключам — ники, значениям — адреса.

```
<< SERVER: contacts-list: «»
>> SERVER: contacts-list: «[General]
friend=friend@m1kc.tk
vasya=pupkin@poupkine.com
[Other]
number=one@m1kc.tk
»
```

2.5.2 Добавление контакта

Для добавления контакта используется пакет типа `contacts-add`. В поле `Content` передаются id контакта и назначаемые ему ник и группа. Если заданной группы не существует, она будет создана.

Типичный случай:

```
<< SERVER: contacts-add: «wellie@server.ru|One more friend|General»
>> SERVER: success: «contacts-add»
```

2.5.3 Переименование контакта

Контакты переименовываются с помощью пакета типа `contacts-rename`. В поле `Content` передаются id контакта и его новый ник. Если такого контакта в списке нет, сервер отвечает пакетом `fail`.

```
<< SERVER: contacts-rename: «friend@m1kc.tk|Enemy»
>> SERVER: success: «contacts-rename»
```

2.5.4 Удаление контакта

Для удаления контакта используется пакет типа `contacts-remove`. В поле `Content` передаётся id контакта. Если такого контакта в списке нет, сервер отвечает пакетом `fail`.

```
<< SERVER: contacts-remove: «one@m1kc.tk»
>> SERVER: success: «contacts-remove»
```

2.5.5 Получение списка групп

Пакет типа `contacts-groups-list` используется для получения списка групп.

```
<< SERVER: contacts-groups-list: «one@m1kc.tk»
>> SERVER: contacts-groups-list: «General|Other»
```

Может, убрать его вообще?

2.5.6 Переименование группы

Пакет типа `contacts-groups-rename` используется для переименования группы. В поле `Content` передаются название группы и её новое имя. Если такой группы нет, сервер отвечает пакетом `fail`.

```
<< SERVER: contacts-groups-rename: «General|Main»
>> SERVER: success: «contacts-groups-rename»
```

Группа также удаляется автоматически, когда из неё удалён последний контакт. Напоминаем, что группа автоматически создаётся при добавлении контакта, если группы с заданным названием ещё нет.