

---

# Project ROBIN RPA

*Выпуск*

ROBIN RPA Team

дек. 24, 2020

---

Содержание:

---

<b>1</b>	<b>Rdp Manager Protocol Specification</b>	<b>2</b>
1.1	Оглавление . . . . .	2
1.2	Условные обозначения . . . . .	2
1.3	Описание протокола . . . . .	3



---

## Rdp Manager Protocol Specification

---

### 1.1 Оглавление

- *Handshake*
- *HandshakeAcknowledge*
- *RdpSession*
- *RdpSessionUser*
- *Result*
- *ResultType*
- *RdpSessionStart*
- *RdpSessionStartAcknowledge*
- *RdpSessionStartResult*
- *RdpSessionStop*
- *RdpSessionStopAcknowledge*
- *RdpSessionStopResult*
- *RdpManagerInfo*

### 1.2 Условные обозначения

**RDP менеджер** *Сервис, выполняющий функции создания и закрытия RDP сессий.*

**Сессия** *Соединение с машиной по RDP протоколу.*

**Машина** *Удаленный компьютер, с которым осуществляется соединение.*

**PID (Process identifier)** *Уникальный числовой идентификатор запущенного процесса*

**Сообщение** *Команда, отправленная сервису/сервисом по защищённому протоколу websocket.*

**Ответ на сообщение** *Структурированная информация в виде цельного объекта, возвращаемая сервисом.*

## 1.3 Описание протокола

Инициатором сообщения выступает пользователь, посредством отдачи команд RDP менеджеру.

### 1.3.1 Объекты

**Handshake** - Команда, иницирующая начало общения с оркестратором.

Protobuf-объект:

```
syntax = "proto3";
package Robin.Protocol.Agent.Messages.API.v2;
import "google/protobuf/timestamp.proto";

message Handshake {
  string type = 1;
  string apiVersion = 2;
  int32 pid = 3; // идентификатор процесса агента
  string mac = 4; // мак адрес сервера, на котором запущен агент
  string host = 5; // имя хоста
  google.protobuf.Timestamp startDate = 6; // Дата и время запуска агента в формате ISO-8601,
  ↪ например 2007-12-03T10:15:30+01:00 Europe
  string agentVersion = 7; // версия агента
  string protocolVersion = 8; // версия поддерживаемого протокола
  string user = 9; // УЗ, под которой запущен агент
}
```

Аттрибуты:

- type (string) - тип сообщения
- apiVersion (string) - версия API RDP менеджера
- pid (int32) - идентификатор процесса RDP менеджера
- mac (string) - MAC-адрес компьютера, на котором запущен RDP менеджер
- host (string) - IP адрес / DNS имя машины
- startDate (google.protobuf.Timestamp) - Дата и время запуска RDP менеджера в формате ISO-8601, например 2007-12-03T10:15:30+01:00 Europe
- agentVersion (string) - версия агента
- protocolVersion (string) - версия поддерживаемого протокола
- user (string) - имя учетной записи пользователя, под которой запущен RDP менеджер

**HandshakeAcknowledge** - Ответ от оркестратора, означающий начало общения с RDP менеджером.

Protobuf-объект:

```
syntax = "proto3";
package Robin.Protocol.RdpManager.Messages.API.v2;

import "Result.proto";

message HandshakeAcknowledge {
  string type = 1; // HANDSHAKE_ACK
  string apiVersion = 2;
  Result result = 3; // результат
  string agentUuid = 4; // оркестратор возвращает agentUuid
}
```

Аттрибуты:

- `type` (string) - доменное имя, к которому принадлежит машина
- `username` (string) - имя пользователя, под которым должна открыться сессия
- `password` (string) - пароль пользователя
- `hostname` (string) - IP адрес/DNS имя машины.

### RdpSession - RDP сессия.

Protobuf-объект:

```

syntax = "proto3";

import "RdpSessionUser.proto";

message RdpSession {
    string hostname = 1; // имя компьютера
    string domain = 2; // доменная зона, в которую входит этот компьютер
    string port = 3; // порт
    RdpSessionUser user = 4; // аутентификационные данные пользователя
}

```

Аттрибуты:

- `domain` (string) - доменное имя, к которому принадлежит машина
- `username` (string) - имя пользователя, под которым должна открыться сессия
- `password` (string) - пароль пользователя
- `hostname` (string) - IP адрес/DNS имя машины.

### RdpSessionUser - 'Пользователь RDP-сессии'.

Protobuf-объект:

```

syntax = "proto3";

import "RdpSessionUser.proto";

message RdpSession {
    string hostname = 1;
    string domain = 2;
    string port = 3;
    RdpSessionUser user = 4;
}

```

Аттрибуты:

- `hostname` (string) - имя компьютера.
- `domain` (string) - доменная зона, в которую входит этот компьютер.
- `port` (string) - RDP-порт, открытый на компьютере, для установки RDP-сессии.
- `user` (RdpSessionUser) - аутентификационные данные пользователя.

### Result - Объект, возвращающий результат сообщения, его статус и код.

Protobuf-объект:

```

message Result {
    string code = 1; // код результата
    ResultType type = 2;
    string description = 3; // описание
};

```

Аттрибуты:

- `code` (string) - код результата
- `type` (enum) - статус сообщения
- `description` (string) - описание сообщения

### 1.3.2 Перечисления

**ResultType** - Статус сообщения, возвращаемого объектом *Result*.

Protobuf-объект:

```
/*универсальный объект с результатом*/
syntax = "proto3";
package Robin.Protocol.RDP.v1;

enum ResultType { // тип результата - перечисление
    SUCCESS = 0;
    WARNING = 1;
    ERROR = 2;
}
```

Возможные значения:

- RDP\_SESSION\_CLOSED (0) - сессия закрыта.
- RDP\_SESSION\_OPENED (1) - сессия открыта.
- RDP\_SESSION\_ERROR (2) - возникла ошибка сессии.

### 1.3.3 Сообщения

**RdpSessionStart** - открыть сессию.

Protobuf-объект:

```
syntax = "proto3";
package Robin.Protocol.RdpManager.Messages.API.v2;

import "RdpSession.proto";

message RdpSessionStart {
    string type = 1; // RDP_SESSION_START
    repeated RdpSession sessions = 2;
}
```

Атрибуты:

- `type` (string) - тип сообщения (RDP\_SESSION\_START).
- `sessions` (RdpSession) - список объектов сессии, передаваемый в сообщении.

**RdpSessionStartAcknowledge** - ответ на сообщение об открытии сессии.

Protobuf-объект:

```
syntax = "proto3";
package Robin.Protocol.RdpManager.Messages.API.v2;

import "Result.proto";

message RdpSessionStartAcknowledge {
    string type = 1; // RDP_SESSION_START_ACKNOWLEDGE
    Result result = 2;
}
```

Атрибуты:

- `type` (string) - тип сообщения (RDP\_SERVICE\_OPEN\_SESSION\_ACKNOWLEDGE).
- `sessionPid` (string) - PID (идентификатор процесса) запущенной RDP сессии. Нужен для ↵  
↪ дальнейшего закрытия сессии.
- `status` (RdpServiceStatus) - статус выполнения команды.

**RdpSessionStartResult** - результат на запуск сессии.

Protobuf-объект:

```
syntax = "proto3";
package Robin.Protocol.RdpManager.Messages.API.v2;

import "Result.proto";

message RdpSessionStartResult {
    string type = 1; // RDP_SESSION_STOP_RESULT
    Result result = 2;
}
```

Атрибуты:

- `type` (string) - тип сообщения (RDP\_SESSION\_START\_RESULT).
- `result` (Result) - объект с результатом выполнения команды.

**RdpSessionStop** - закрыть сессию.

Protobuf-объект:

```
syntax = "proto3";
package Robin.Protocol.RdpManager.Messages.API.v2;

import "RdpSession.proto";

message RdpSessionStop {
    string type = 1; // RDP_SESSION_STOP
    repeated RdpSession sessions = 2;
}
```

Атрибуты:

- `type` (string) - тип сообщения (RDP\_SERVICE\_CLOSE\_SESSION).
- `managerUuid` (string) - идентификатор RDP менеджера.
- `username` (string) - имя пользователя
- `hostname` (string) - адрес машины
- `sessionPid` (string) - PID запущенной RDP сессии.

*Примечание:*

В этом сообщении можно использовать или сочетание атрибутов `username` и `hostname` (в этом случае `sessionPid` указывать не требуется), или только `sessionPid`. **RdpSessionStopAcknowledge** - ответ на сообщение о закрытии сессии.

Protobuf-объект:

```
syntax = "proto3";
package Robin.Protocol.RdpManager.Messages.API.v2;

message RdpSessionStopAcknowledge {
    string type = 1; // RDP_SESSION_STOP_ACKNOWLEDGE
}
```

Атрибуты:

- `type` - тип сообщения (RDP\_SERVICE\_CLOSE\_SESSION\_ACKNOWLEDGE)

**RdpSessionStopResult** - результат на запуск сессии.

Protobuf-объект:

```
syntax = "proto3";
package Robin.Protocol.RdpManager.Messages.API.v2;

import "Result.proto";

message RdpSessionStopResult {
    string type = 1; // RDP_SESSION_STOP_RESULT
    Result result = 2;
}
```

Аттрибуты:

- `type` (string) - тип сообщения (RDP\_SESSION\_STOP\_RESULT).  
- `result` (Result) - объект с результатом выполнения команды.

**RdpManagerInfo** - Информация о состоянии сессий RDP-менеджера.

Protobuf-объект:

```
syntax = "proto3";
import "RdpSession.proto";

message RdpManagerInfo {
    string type = 1; // RDP_MANAGER_INFO
    repeated RdpSession runningSessions = 2;
}
```

Аттрибуты:

- `type` - тип сообщения (RDP\_SERVICE\_CLOSE\_SESSION\_ACKNOWLEDGE)  
- `runningSessions` - список открытых сессий RDP-менеджера.