

Документация по серверу OpenTelega 1.0

Содержание

Краткий обзор сервера OpenTelega	2
Команды сервера	3
Команды администратора	3
Первоначальная настройка сервера	3
Получение списка опций	4
Установка значения опции	5
Смена пароля администратора	7
Регистрация пользователя	8
Сброс пароля пользователя	9
Удаление пользователя	10
Получение списка пользователей	11
Получение списка файлов	12
Скачивание файла	13
Удаление файла	15
Очистка БД и удаление ненужных файлов из директории загрузки	16
Получение версии сервера	17
Команды пользователя	18
Загрузка файла на сервер	18
Получение списка файлов	19
Скачивание файла	21
Смена пароля	22
Получение количества файлов для адресата	23
Получение версии сервера	24
Приложения	25
Запуск сервера в тестовом окружении	25
Коды завершения операции на сервере	25
Опции сервера	27
Безопасность использования сервера OpenTelega	27
FAQ	29
Где можно применить данный сервер?	29
Как посмотреть список платформ, под которые можно написать клиент?	29
Как определить, завершилась ли команда с ошибкой или без неё?	29
У меня сервер вернул код <i>HTTP</i> 500. Что это значит?	29

В этом документе рассмотрены HTTP-запросы для работы с сервером OpenTelega, даны примеры cURL-запросов, а также приведены возможные ответы сервера на рассматриваемые HTTP-запросы. Приведены примеры cURL-запросов для использования в клиенте. Даны рекомендации для создания безопасных коммуникационных систем.

Краткий обзор сервера OpenTelega

OpenTelega — это сервер обмена файлами с открытым исходным кодом, написанный на веб-фреймворке *django*. Простая структура сервера обеспечивает лёгкость написания собственных коммуникационных приложений, таких как чаты, мессенджеры, а также ботов к ним.

Взаимодействие с сервером осуществляется через команды, отправляемые на сервер по протоколу *HTTP*. Поддерживается вывод в трёх форматах: *CSV*, *XML*, *JSON* (по умолчанию), что позволяет разрабатывать на основе данного сервера свои приложения без использования стороннего программного обеспечения во всех современных средствах разработки ПО.

Команды сервера

На сервере можно создать одного администратора и сколько угодно пользователей. При первоначальной настройке сервера создаётся администратор со случайно сгенерированным логином и паролем. Логин и пароль у администратора регистрозависимые. После первоначальной настройки администратор может создавать пользователей. Команды отправляются на сервер посредством HTTP(S)-запросов. Все аргументы (параметры) команды передаются через метод *POST*, а название команды передаётся через метод *GET*.

Команды администратора

Функции администратора предназначены для управления сервером. Пара логин-пароль администратора и пользователей находятся в разных таблицах (моделях). Таким образом, логин и пароль администратора нельзя использовать в командах пользователя.

Первоначальная настройка сервера

Команда вызывается для установки значений параметров в БД, а также генерации логина и пароля администратора. В БД сервера помечается то, что сервер был инициализирован, таким образом, второй раз инициализировать сервер нельзя. Команда возвращает список из одного элемента, содержащего логин и пароль администратора.

Таблица 1. Аргументы команды

Метод	Аргумент	Обязательное поле	Примечание
GET	fileserver/initialize_server/	Да	Название команды
POST	outputFormat	Нет	Формат вывода ответа сервера

Таблица 2. Аргументы ответа сервера

Аргумент	Примечание
AdminUsername	Логин администратора
AdminPassword	Пароль администратора

Таблица 3. Возможные ошибки

Ошибка	Примечание
PermissionDenied	Сервер уже был инициализирован
PermissionDenied	Ошибка чтения/записи опций
UnknownError	Неизвестная ошибка

Пример cURL запроса

Запрос

```
curl -X POST http://127.0.0.1:8000/fileserver/initialize_server/
```

Ответ

```
[{"AdminUsername": "64wyygtm", "AdminPassword": "louq0nfigrtwowa4"}]
```

Получение списка опций

Подробнее об опциях см. в главе [Опции](#).

Команда возвращает список с названиями и значениями открытых опций.

Таблица 4. Аргументы команды

Метод	Аргумент	Обязательное поле	Примечание
GET	fileserver/get_options_list/	Да	Название команды
POST	adminUserName	Да	Логин администратора
POST	adminPassword	Да	Пароль администратора
POST	outputFormat	Нет	Формат вывода ответа сервера

Таблица 5. Аргументы ответа сервера

Аргумент	Примечание
Name	Название опции
Value	Значение опции

Таблица 6. Возможные ошибки

Ошибка	Примечание
NoCredentials	Отсутствует логин или пароль администратора
AdminDidNotCreated	Учётная запись администратора отсутствует в системе (скорее всего, сервер не инициализирован)
WrongCredentials	Неправильные логин или пароль
UnknownError	Неизвестная ошибка

Примеры cURL запросов

Запрос (вывод в формате XML)

```
curl -X POST -F "adminUserName=64wyygtn" -F "adminPassword=louq0nfigrtwowa4" -F "outputFormat=xml" http://127.0.0.1:8000/fileserver/get_options_list/
```

Ответ

```
<root>
  <element>
    <Name>FILE_SIZE_LIMIT</Name>
    <Value>1073741824</Value>
  </element>
  <element>
    <Name>LIMIT_MESSAGES_FOR_QUERY</Name>
    <Value>1000</Value>
  </element>
</root>
```

Запрос (заведомо неправильные логин и пароль, вывод в формате CSV)

```
curl -X POST -F "adminUserName=123" -F "adminPassword=456" -F "outputFormat=csv" http://127.0.0.1:8000/fileserver/get_options_list/
```

Ответ

```
"StatusCode"
"WrongCredentials"
```

Установка значения опции

Подробнее об опциях см. в главе [Опции сервера](#).

На сервере изменяется значение уже созданной открытой опции. Возвращает код завершения данной операции (см. [Коды завершения операции на сервере](#)).

Таблица 7. Аргументы команды

Метод	Аргумент	Обязательное поле	Примечание
GET	fileserver/change_option/	Да	Название команды
POST	adminUserName	Да	Логин администратора
POST	adminPassword	Да	Пароль администратора
POST	optionName	Да	Название опции
POST	optionValue	Да	Значение опции

Метод	Аргумент	Обязательное поле	Примечание
POST	outputFormat	Нет	Формат вывода ответа сервера

Таблица 8. Аргументы ответа сервера

Аргумент	Примечание
StatusCode	Код завершения операции

Таблица 9. Возможные ошибки

Ошибка	Примечание
NoCredentials	Отсутствует логин или пароль администратора
AdminDidNotCreated	Учётная запись администратора отсутствует в системе (скорее всего, сервер не инициализирован)
WrongCredentials	Неправильные логин или пароль
NotEnoughArguments	Отсутствует аргумент названия опции или значения опции
CouldNotGetOption	Опция отсутствует или не является открытой
UnknownError	Неизвестная ошибка

Примеры cURL запросов

Запрос (вывод в формате XML)

```
curl -X POST -F "adminUserName=64wyygtm" -F "adminPassword=louq0nfigrtwowa4" -F "optionName=FILE_SIZE_LIMIT" -F "optionValue=10737" -F "outputFormat=xml" http://127.0.0.1:8000/filesserver/change_option/
```

Ответ

```
<root>
  <element>
    <StatusCode>OK</StatusCode>
  </element>
</root>
```

Запрос (вывод в формате XML, опции не существует в БД)

```
curl -X POST -F "adminUserName=64wyygtm" -F "adminPassword=louq0nfigrtwowa4" -F "optionName=BlahBlahBlah" -F "optionValue=12345" -F "outputFormat=xml" http://127.0.0.1:8000/filesserver/change_option/
```


Ответ

```
<root>
  <element>
    <StatusCode>CouldNotGetOption</StatusCode>
  </element>
</root>
```

Смена пароля администратора

Команда меняет пароль администратора. Возвращает код завершения данной операции (см. [Коды завершения операции на сервере](#)).

Таблица 10. Аргументы команды

Метод	Аргумент	Обязательное поле	Примечание
GET	fileserver/change_admin_password/	Да	Название команды
POST	adminUserName	Да	Логин администратора
POST	adminPassword	Да	Пароль администратора
POST	newAdminPassword	Да	Новый пароль
POST	outputFormat	Нет	Формат вывода ответа сервера

Таблица 11. Аргументы ответа сервера

Аргумент	Примечание
StatusCode	Код завершения операции

Таблица 12. Возможные ошибки

Ошибка	Примечание
NoCredentials	Отсутствует логин или пароль администратора
AdminDidNotCreated	Учётная запись администратора отсутствует в системе (скорее всего, сервер не инициализирован)
WrongCredentials	Неправильные логин или пароль
PermissionDenied	Хеш пароля или соль являются открытыми опциями
CouldNotGetNewPassword	Не указан новый пароль
UnknownError	Неизвестная ошибка

Пример cURL запроса

Запрос

```
curl -X POST -F "adminUserName=64wyygtm" -F "adminPassword=louq0nfigrtwowa4" -F "newAdminPassword=1" http://127.0.0.1:8000/fileserver/change_admin_password/
```

Ответ

```
[{"StatusCode": "OK"}]
```

Регистрация пользователя

На сервере регистрируется пользователь с заданным логином. Логины пользователей — регистронезависимые. Сервер возвращает список из одного элемента со случайно-сгенерированным паролем пользователя. Допустимый набор символов, используемый в логине регистрируемого пользователя: "abcdefghijklmnopqrstuvwxyz0123456789-_"

Таблица 13. Аргументы команды

Метод	Аргумент	Обязательное поле	Примечание
GET	fileserver/register_user/	Да	Название команды
POST	adminUserName	Да	Логин администратора
POST	adminPassword	Да	Пароль администратора
POST	usernameForRegistration	Да	Логин пользователя
POST	outputFormat	Нет	Формат вывода ответа сервера

Таблица 14. Аргументы ответа сервера

Аргумент	Примечание
UserPassword	Пароль пользователя

Таблица 15. Возможные ошибки

Ошибка	Примечание
NoCredentials	Отсутствует логин или пароль администратора
AdminDidNotCreated	Учётная запись администратора отсутствует в системе (скорее всего, сервер не инициализирован)
WrongCredentials	Неправильные логин или пароль
NotEnoughArguments	Не указан логин пользователя
InvalidParameters	Логин содержит недопустимые символы

Ошибка	Примечание
AlreadyRegisteredUser	Пользователь с таким логином уже существует
UnknownError	Неизвестная ошибка

Пример cURL запроса

Запрос

```
curl -X POST -F "adminUserName=64wyygtm" -F "adminPassword=1" -F "usernameForRegistration=user" http://127.0.0.1:8000/fileserver/register_user/
```

Ответ

```
[{"UserPassword": "c44pitgq3ucysibf"}]
```

Сброс пароля пользователя

На сервере заново регенерируется пароль уже созданного пользователя. Сервер возвращает список из одного элемента со случайно-сгенерированным паролем пользователя.

Таблица 16. Аргументы команды

Метод	Аргумент	Обязательное поле	Примечание
GET	fileserver/reset_user_password/	Да	Название команды
POST	adminUserName	Да	Логин администратора
POST	adminPassword	Да	Пароль администратора
POST	usernameForResetPassword	Да	Логин пользователя
POST	outputFormat	Нет	Формат вывода ответа сервера

Таблица 17. Аргументы ответа сервера

Аргумент	Примечание
UserPassword	Пароль пользователя

Таблица 18. Возможные ошибки

Ошибка	Примечание
NoCredentials	Отсутствует логин или пароль администратора

Ошибка	Примечание
AdminDidNotCreated	Учётная запись администратора отсутствует в системе (скорее всего, сервер не инициализирован)
WrongCredentials	Неправильные логин или пароль
NotEnoughArguments	Не указан логин пользователя
UserNotFound	Пользователь с таким логином не найден
UnknownError	Неизвестная ошибка

Пример cURL запроса

Запрос (вывод в формате CSV)

```
curl -X POST -F "adminUserName=64wyygtm" -F "adminPassword=1" -F
"usernameForResetPassword=user" -F "outputFormat=csv"
http://127.0.0.1:8000/fileserver/reset_user_password/
```

Ответ

```
"UserPassword"
"080y/iikl1hsvsks"
```

Удаление пользователя

На сервере удаляется пользователь. Команда возвращает код завершения данной операции (см. [Коды завершения операции на сервере](#)).

Таблица 19. Аргументы команды

Метод	Аргумент	Обязательное поле	Примечание
GET	fileserver/delete_user/	Да	Название команды
POST	adminUserName	Да	Логин администратора
POST	adminPassword	Да	Пароль администратора
POST	usernameForDelete	Да	Логин пользователя
POST	outputFormat	Нет	Формат вывода ответа сервера

Таблица 20. Аргументы ответа сервера

Аргумент	Примечание
StatusCode	Код завершения операции

Таблица 21. Возможные ошибки

Ошибка	Примечание
NoCredentials	Отсутствует логин или пароль администратора
AdminDidNotCreated	Учётная запись администратора отсутствует в системе (скорее всего, сервер не инициализирован)
WrongCredentials	Неправильные логин или пароль
NotEnoughArguments	Не указан логин пользователя
UserNotFound	Пользователь с таким логином не найден
UnknownError	Неизвестная ошибка

Пример cURL запроса

Запрос

```
curl -X POST -F "adminUserName=64wyygtm" -F "adminPassword=1" -F "usernameForDelete=user" http://127.0.0.1:8000/fileserver/delete_user/
```

Ответ

```
[{"StatusCode": "OK"}]
```

Получение списка пользователей

Возвращает список пользователей.

Таблица 22. Аргументы команды

Метод	Аргумент	Обязательное поле	Примечание
GET	fileserver/list_all_users/	Да	Название команды
POST	adminUserName	Да	Логин администратора
POST	adminPassword	Да	Пароль администратора
POST	outputFormat	Нет	Формат вывода ответа сервера

Таблица 23. Аргументы ответа сервера

Аргумент	Примечание
Username	Логин пользователя

Таблица 24. Возможные ошибки

Ошибка	Примечание
NoCredentials	Отсутствует логин или пароль администратора
AdminDidNotCreated	Учётная запись администратора отсутствует в системе (скорее всего, сервер не инициализирован)
WrongCredentials	Неправильные логин или пароль
UnknownError	Неизвестная ошибка

Пример cURL запроса

Запрос

```
curl -X POST -F "adminUserName=64wyygtm" -F "adminPassword=1"
http://127.0.0.1:8000/fileserver/list_all_users/
```

Ответ

```
[{"Username": "user"}, {"Username": "user2"}, {"Username": "user3"}]
```

Получение списка файлов

Возвращает список всех файлов.

Таблица 25. Аргументы команды

Метод	Аргумент	Обязательное поле	Примечание
GET	fileserver/list_all_files/	Да	Название команды
POST	adminUserName	Да	Логин администратора
POST	adminPassword	Да	Пароль администратора
POST	outputFormat	Нет	Формат вывода ответа сервера

Таблица 26. Аргументы ответа сервера

Аргумент	Примечание
Id	Идентификатор файла
Date	Дата загрузки
Sender	Логин отправителя
Recipients	Логины получателей, разделённые запятой
Hashsum	Хеш-сумма
Algorithm	Алгоритм подсчёта хеш-суммы
Size	Размер файла в байтах

Таблица 27. Возможные ошибки

Ошибка	Примечание
NoCredentials	Отсутствует логин или пароль администратора
AdminDidNotCreated	Учётная запись администратора отсутствует в системе (скорее всего, сервер не инициализирован)
WrongCredentials	Неправильные логин или пароль
GetFileListError	Ошибка получения списка файлов
UnknownError	Неизвестная ошибка

Пример cURL запроса

Запрос

```
curl -X POST -F "adminUserName=64wyygtm" -F "adminPassword=1"
http://127.0.0.1:8000/fileserver/list_all_files/
```

Ответ

```
[{"Id": "1", "Date": "2017-08-16 04:51:28.525526+00:00", "Sender": "user",
"Recipients": "user2", "Hashsum":
"355cdfac40614cf818dec74edba11a1210270aad292fbf929b8db0a8802639585ef88ccda809c19ff6b6
c436989063058217bd07680faa677f7200541d307dd", "Algorithm": "sha512", "Size":
"155648"}, {"Id": "2", "Date": "2017-08-16 04:52:52.646161+00:00", "Sender": "user",
"Recipients": "user3", "Hashsum":
"fb94b9e7017f73dfac4bba199f88a425ea71dcc5f2c6a4184ef76cc7645da53d6fef2060fa4867842c6e2
ba9ca539253ad86e7aeb98d23995d645bf55cc4164f", "Algorithm": "sha512", "Size":
"155648"}, {"Id": "3", "Date": "2017-08-16 04:53:48.250649+00:00", "Sender": "user",
"Recipients": "user2, user3", "Hashsum":
"ed927b682f20e37c51c72157623d81f4d10a0d9c2ff022aa5887cc9cc523074f59370c1edacf95e64e8f2
d06d983942ffdfbb129c0625a0eb983c14a6a413a20", "Algorithm": "sha512", "Size":
"155648"}, {"Id": "4", "Date": "2017-08-16 04:54:11.454020+00:00", "Sender": "user",
"Recipients": "user3", "Hashsum":
"ad2d0e3b233f0f0873dd6ced399f9083d0cc0fdda1bbb30dc1650408a21d0f3aae3d9a2f6dd8bbdce2d67
1921c4fdd56230503ed0ec75804096d98d2f9e2029b", "Algorithm": "sha512", "Size":
"155648"}]
```

Скачивание файла

Скачивает файл при помощи логина и пароля администратора.

Таблица 28. Аргументы команды

Метод	Аргумент	Обязательное поле	Примечание
GET	fileserver/admin_down load_file/	Да	Название команды

Метод	Аргумент	Обязательное поле	Примечание
POST	adminUserName	Да	Логин администратора
POST	adminPassword	Да	Пароль администратора
POST	targetId	Да	Идентификатор файла
POST	outputFormat	Нет	Формат вывода ответа сервера

Таблица 29. Возможные ошибки

Ошибка	Примечание
NoCredentials	Отсутствует логин или пароль администратора
AdminDidNotCreated	Учётная запись администратора отсутствует в системе (скорее всего, сервер не инициализирован)
WrongCredentials	Неправильные логин или пароль
NotEnoughArguments	Не указан идентификатор файла
FileNotFound	Файл с указанным идентификатором не найден в БД
UnknownError	Неизвестная ошибка

Пример cURL запроса

Запрос

```
curl -D headers.txt -X POST -F "adminUserName=64wyygtm" -F "adminPassword=1" -F "targetId=2" http://127.0.0.1:8000/fileserver/admin_download_file/ -o file.bin
```

Команда запишет файл (или код ошибки) в **file.bin**, а в **headers.txt** можно посмотреть код HTTP-ответа:

HTTP-заголовки ответа

```
$ cat headers.txt
HTTP/1.0 200 OK
Date: Wed, 16 Aug 2017 15:53:54 GMT
Server: WSGIServer/0.2 CPython/3.6.2
Content-Type: text/html; charset=utf-8
Content-Disposition: attachment; filename=2
X-Frame-Options: SAMEORIGIN
Content-Length: 155648
```

Видим, что код ответа HTTP - **200**, значит файл скачан в **file.bin**.

Рассмотрим другую ситуацию, когда по какой-либо причине операция завершилась с ошибкой.

HTTP-заголовки ответа

```
$ cat headers.txt
HTTP/1.0 400 Bad Request
Date: Wed, 16 Aug 2017 15:57:44 GMT
Server: WSGIServer/0.2 CPython/3.6.2
Content-Type: text/html; charset=utf-8
X-Frame-Options: SAMEORIGIN
Content-Length: 32
```

Здесь код ответа равен **400**, тогда в файл скачался код ошибки. Открываем файл и видим ошибку, возникшую на сервере.

Просмотр кода ошибки

```
$ cat file.bin
[{"StatusCode": "FileNotFound"}]
```

Удаление файла

Удаляет файл с сервера. Команда возвращает код завершения данной операции (см. [Коды завершения операции на сервере](#)).

Таблица 30. Аргументы команды

Метод	Аргумент	Обязательное поле	Примечание
GET	fileserver/delete_file/	Да	Название команды
POST	adminUserName	Да	Логин администратора
POST	adminPassword	Да	Пароль администратора
POST	targetId	Да	Идентификатор удаляемого файла
POST	outputFormat	Нет	Формат вывода ответа сервера

Таблица 31. Аргументы ответа сервера

Аргумент	Примечание
StatusCode	Код завершения операции

Таблица 32. Возможные ошибки

Ошибка	Примечание
NoCredentials	Отсутствует логин или пароль администратора

Ошибка	Примечание
AdminDidNotCreated	Учётная запись администратора отсутствует в системе (скорее всего, сервер не инициализирован)
WrongCredentials	Неправильные логин или пароль
NotEnoughArguments	Не указан идентификатор удаляемого файла
FileNotFound	Файл с таким идентификатором не найден
UnknownError	Неизвестная ошибка

Пример cURL запроса

Запрос

```
curl -X POST -F "adminUserName=64wyygtm" -F "adminPassword=1" -F "targetId=3"
http://127.0.0.1:8000/fileserver/delete_file/
```

Ответ

```
[{"StatusCode": "OK"}]
```

Очистка БД и удаление ненужных файлов из директории загрузки

Удаляет записи о файлах из БД, если соответствующих файлов нет в ФС и удаляет файлы в ФС из директории загрузки файлов, если нет соответствующих записей в БД. При большой загрузке сервера загружаемые пользователями файлы могут сохраняться в ФС, но не сохраняться в БД. По разным причинам может случаться и обратная ситуация, когда файл может сохраняться в БД, но не сохраняться в ФС. В обоих случаях, скачать такие файлы методами сервера *OpenTelega* невозможно, поэтому написана такая функция, дабы не приходилось лазить в media-директорию, через ssh-консоль для обслуживания сервера. Возвращает список из одного элемента, содержащий идентификаторы файлов: успешно удалённых из БД; с возникшей ошибкой во время удаления из БД; успешно удалённых из ФС; с возникшей ошибкой во время удаления из ФС.

Таблица 33. Аргументы команды

Метод	Аргумент	Обязательное поле	Примечание
GET	fileserver/clean_media_directory/	Да	Название команды
POST	adminUserName	Да	Логин администратора
POST	adminPassword	Да	Пароль администратора
POST	outputFormat	Нет	Формат вывода ответа сервера

Таблица 34. Аргументы ответа сервера

Аргумент	Примечание
SuccessfullDeletedFilesFromDb	Идентификаторы файлов, разделённые точкой с запятой, успешно удалённые из БД
DeleteFilesFromDbError	Идентификаторы файлов, разделённые точкой с запятой, с возникшей ошибкой во время удаления из БД
SuccessfullDeletedFilesFromFs	Имена файлов, разделённые точкой с запятой, успешно удалённые из ФС
DeleteFilesFromFsError	Имена файлов, разделённые точкой с запятой, с возникшей ошибкой во время удаления из ФС

Таблица 35. Возможные ошибки

Ошибка	Примечание
NoCredentials	Отсутствует логин или пароль администратора
AdminDidNotCreated	Учётная запись администратора отсутствует в системе (скорее всего, сервер не инициализирован)
WrongCredentials	Неправильные логин или пароль
UnknownError	Неизвестная ошибка

Пример cURL запроса

Запрос

```
curl -X POST -F "adminUserName=64wyygtm" -F "adminPassword=1"
http://127.0.0.1:8000/fileserver/clean_media_directory/
```

Ответ

```
[{"SuccessfullDeletedFilesFromDb": "1; 4", "DeleteFilesFromDbError": "",
"SuccessfullDeletedFilesFromFs": "123; 456", "DeleteFilesFromFsError": ""}]
```

Получение версии сервера

Возвращает список из одного элемента, содержащего версию сервера (текущая: 1.0).
Аргументы команды

Метод	Аргумент	Обязательное поле	Примечание
GET	fileserver/admin_get_v ersion/	Да	Название команды
POST	adminUserName	Да	Логин администратора

Метод	Аргумент	Обязательное поле	Примечание
POST	adminPassword	Да	Пароль администратора
POST	outputFormat	Нет	Формат вывода ответа сервера

Таблица 36. Аргументы ответа сервера

Аргумент	Примечание
Version	Номер версии

Таблица 37. Возможные ошибки

Ошибка	Примечание
NoCredentials	Отсутствует логин или пароль администратора
AdminDidNotCreated	Учётная запись администратора отсутствует в системе (скорее всего, сервер не инициализирован)
WrongCredentials	Неправильные логин или пароль
UnknownError	Неизвестная ошибка

Пример cURL запроса

Запрос

```
curl -X POST -F "adminUserName=64wyygtm" -F "adminPassword=1"
http://127.0.0.1:8000/fileserver/admin_get_version/
```

Ответ

```
[{"Version": "1.0"}]
```

Команды пользователя

Во всех командах пользователя используются логин и пароль пользователя.

Загрузка файла на сервер

Закачивает файл на сервер. Возвращает идентификатор загруженного файла.

Таблица 38. Аргументы команды

Метод	Аргумент	Обязательное поле	Примечание
GET	fileserver/fileserver/upload_file/	Да	Название команды
POST	userName	Да	Логин пользователя

Метод	Аргумент	Обязательное поле	Примечание
POST	password	Да	Пароль пользователя
POST	recipients	Да	Логин адресатов, указанные через точку с запятой
POST	uploadedFile	Да	Файл для загрузки сервером
POST	outputFormat	Нет	Формат вывода ответа сервера

Таблица 39. Аргументы ответа сервера

Аргумент	Примечание
FileId	Идентификатор файла

Таблица 40. Возможные ошибки

Ошибка	Примечание
NoCredentials	Отсутствует логин или пароль пользователя
WrongCredentials	Неправильные логин или пароль
NoRecipients	Отсутствуют адресаты
WrongRecipients	Один или несколько логинов адресатов не найдены в БД
UploadFileError	Ошибка при получении файла
CouldNotGetOption	Ошибка при получении опции максимального размера загружаемого файла
FileSizeExceedsLimit	Размер файла превышает допустимый предел
DbWorkingError	Ошибка сохранения файла
UnknownError	Неизвестная ошибка

Пример cURL запроса

Запрос (вывод в формате XML)

```
curl -X POST -F "uploadedFile=@file.bin" -F "userName=user" -F "password=123" -F "recipients=user2;user;user3" http://127.0.0.1:8000/fileserver/upload_file/
```

Ответ

```
[{"FileId": "12"}]
```

Получение списка файлов

Возвращает список файлов, адресованных пользователю. Максимальное количество

возвращаемых файлов в списке определяется опцией `LIMIT_MESSAGES_FOR_QUERY`. Также можно указать значение идентификатора, начиная с которого будет формироваться список.

Таблица 41. Аргументы команды

Метод	Аргумент	Обязательное поле	Примечание
GET	fileserver/get_file_list/	Да	Название команды
POST	userName	Да	Логин пользователя
POST	password	Да	Пароль пользователя
POST	startFromId	Нет	Вывести файлы, идентификаторы которых больше указанного
POST	outputFormat	Нет	Формат вывода ответа сервера

Таблица 42. Аргументы ответа сервера

Аргумент	Примечание
Id	Идентификатор файла
Date	Дата загрузки
Sender	Логин отправителя
Recipients	Логины получателей, разделённые запятой
Hashsum	Хеш-сумма
Algorithm	Алгоритм подсчёта хеш-суммы
Size	Размер файла в байтах

Таблица 43. Возможные ошибки

Ошибка	Примечание
NoCredentials	Отсутствует логин или пароль администратора
WrongCredentials	Неправильные логин или пароль
InvalidParameters	Ошибка при преобразовании значения параметра startFromId в целое число
GetFileListError	Ошибка получения списка файлов
UnknownError	Неизвестная ошибка

Пример cURL запроса

Запрос

```
curl -X POST -F "userName=user" -F "password=1"  
http://127.0.0.1:8000/fileserver/get_file_list/
```

```
[{"Id": "7", "Date": "2017-08-16 17:49:27.921752+00:00", "Sender": "user",
"Recipients": "user2, user3, user", "Hashsum":
"545cf0e52039a339b006c05bcd99196aad27178da4c9a9ea80cb13cde230efd6e195d3d8c64ab5438ff50
9981a2bcd78865d142d8fde3b1990f54b4914173ed7", "Algorithm": "sha512", "Size":
"155648"}, {"Id": "8", "Date": "2017-08-16 17:49:29.945411+00:00", "Sender": "user",
"Recipients": "user2, user3, user", "Hashsum":
"1ee3780d7bacc4ab867e99ae5875326dc2e826bb156c97f6bfd6edfdadcbbe7c1609d421e8d7b1e69a29a
f1d227b7311937c0887e0be941e8f4ba67f8be38e83", "Algorithm": "sha512", "Size":
"155648"}, {"Id": "9", "Date": "2017-08-16 18:34:24.915857+00:00", "Sender": "user",
"Recipients": "user2, user, user3", "Hashsum":
"99438f681a8b74e6953cdacc151a0874ed0d86d1d7160f8e18372d61d104f0640fdef02df1e7a66b7a1e8
6d2c3dc7d2e26e01e3e53da685f836d0e164fe3265b", "Algorithm": "sha512", "Size": "32"}]
```

Скачивание файла

Функция, аналогичная [функции скачивания файла администратором](#), за исключением того что пользователь может скачать только те файлы, адресатом которых он является.

Таблица 44. Аргументы команды

Метод	Аргумент	Обязательное поле	Примечание
GET	fileserver/download_file_by_id/	Да	Название команды
POST	userName	Да	Логин пользователя
POST	password	Да	Пароль пользователя
POST	id	Да	Идентификатор файла
POST	outputFormat	Нет	Формат вывода ответа сервера

Таблица 45. Возможные ошибки

Ошибка	Примечание
NoCredentials	Отсутствует логин или пароль пользователя
WrongCredentials	Неправильные логин или пароль
NoFileId	Не указан идентификатор файла
FileNotFound	Файл с указанным идентификатором не найден в БД, либо пользователь не является адресатом указанного файла
UnknownError	Неизвестная ошибка

Пример cURL запроса

```
curl -D headers.txt -X POST -F "userName=user" -F "password=12345" -F "id=6"
http://127.0.0.1:8000/fileserver/download_file_by_id/ -o file.bin
```

Подробнее о заголовках *HTTP* и скачивании файла см. в [функции скачивания файла администратором](#).

Смена пароля

Команда меняет пароль пользователя. Возвращает код завершения данной операции (см. [Коды завершения операции на сервере](#)).

Таблица 46. Аргументы команды

Метод	Аргумент	Обязательное поле	Примечание
GET	fileserver/change_user_password/	Да	Название команды
POST	userName	Да	Логин пользователя
POST	password	Да	Пароль пользователя
POST	newPassword	Да	Новый пароль
POST	outputFormat	Нет	Формат вывода ответа сервера

Таблица 47. Аргументы ответа сервера

Аргумент	Примечание
StatusCode	Код завершения операции

Таблица 48. Возможные ошибки

Ошибка	Примечание
NoCredentials	Отсутствует логин или пароль пользователя
WrongCredentials	Неправильные логин или пароль
CouldNotGetNewPassword	Не указан новый пароль
UnknownError	Неизвестная ошибка

Пример cURL запроса

Запрос (вывод в формате XML)

```
curl -X POST -F "userName=user" -F "password=123" -F "newPassword=1" -F
"outputFormat=xml" http://127.0.0.1:8000/fileserver/change_user_password/
```



```
<root>
  <element>
    <StatusCode>OK</StatusCode>
  </element>
</root>
```

Получение количества файлов для адресата

Возвращает список из одного элемента, содержащего количество файлов, адресатом которого является пользователь. Функция может оказаться полезной, если у адресата достаточно много файлов (по умолчанию, в списке файлов выводятся не более **1000** (опция **LIMIT_MESSAGES_FOR_QUERY**)).

Таблица 49. Аргументы команды

Метод	Аргумент	Обязательное поле	Примечание
GET	fileserver/get_files_count/	Да	Название команды
POST	userName	Да	Логин пользователя
POST	password	Да	Пароль пользователя
POST	outputFormat	Нет	Формат вывода ответа сервера

Таблица 50. Аргументы ответа сервера

Аргумент	Примечание
FilesCount	Количество файлов

Таблица 51. Возможные ошибки

Ошибка	Примечание
NoCredentials	Отсутствует логин или пароль пользователя
WrongCredentials	Неправильные логин или пароль
InvalidParameters	Ошибка при получении количества файлов
UnknownError	Неизвестная ошибка

Пример cURL запроса

Запрос (вывод в формате XML)

```
curl -X POST -F "userName=user" -F "password=1"
http://127.0.0.1:8000/fileserver/get_files_count/
```

Ответ

```
[{"FilesCount": "2"}]
```

Получение версии сервера

Возвращает список из одного элемента, содержащего версию сервера (текущая: **1.0**).
Используется пара логин-пароль пользователя.

Таблица 52. Аргументы команды

Метод	Аргумент	Обязательное поле	Примечание
GET	fileserver/user_get_version/	Да	Название команды
POST	userName	Да	Логин пользователя
POST	password	Да	Пароль пользователя
POST	outputFormat	Нет	Формат вывода ответа сервера

Таблица 53. Аргументы ответа сервера

Аргумент	Примечание
Version	Номер версии

Таблица 54. Возможные ошибки

Ошибка	Примечание
NoCredentials	Отсутствует логин или пароль администратора
WrongCredentials	Неправильные логин или пароль
UnknownError	Неизвестная ошибка

Пример cURL запроса

Запрос

```
curl -X POST -F "userName=user" -F "password=1"  
http://127.0.0.1:8000/fileserver/user_get_version/
```

Ответ

```
[{"Version": "1.0"}]
```

Приложения

Запуск сервера в тестовом окружении

1. Клонировем репозиторий сервера:

```
git clone https://github.com/opentelega/OpenTelega.git
```

2. Переходим в папку сервера:

```
cd OpenTelega
```

3. Выполняем миграцию

```
python manage.py makemigrations fileserver  
python manage.py migrate
```

4. Запускаем сервер

```
python manage.py runserver
```

Коды завершения операции на сервере

В случае ошибок выполнения команд пользователя выводится список из одного элемента, в котором указывается код завершения операции. Все коды завершения операции, кроме кода успешного выполнения команды, выводятся с 400-м кодом состояния HTTP. Код успешного выполнения операции выводится с 200-м кодом состояния HTTP.

Таблица 55. Список кодов завершения операции

Код завершения операции	HTTP-код	Примечание
OK	200	Успешное завершение операции
NoCredentials	400	Отсутствует логин или пароль
WrongCredentials	400	Неправильные логин или пароль
InvalidParameters	400	Параметры команды имеют неверный вид
UploadFileError	400	Ошибка при загрузке файла

Код завершения операции	HTTP-код	Примечание
FileSizeExceedsLimit	400	Размер загружаемого файла превышает допустимое значение
WrongRecipients	400	Не найдены адресаты исходного файла
NoRecipients	400	Не указаны адресаты
FileNotFound	400	Файл не найден
GetFileListError	400	Ошибка при получения списка файлов
NoFileId	400	Не указан идентификатор файла
CouldNotGetNewPassword	400	Не указан новый пароль
AdminDidNotCreated	400	Администратор не инициализирован
CouldNotGetOption	400	Ошибка чтения/записи опции
UserNotFound	400	Пользователь не найден
PermissionDenied	400	Доступ запрещён
NotEnoughArguments	400	Недостаточно аргументов в команде
AlreadyRegisteredUser	400	Пользователь с таким именем уже зарегистрирован
DbWorkingError	400	Ошибка при работе с БД
UnknownError	400	Неизвестная ошибка

Таблица 56. Примеры ответов с кодами завершения операции

Формат	Ответ
CSV	"StatusCode" "WrongCredentials"
XML	<root> <element> <StatusCode>WrongCredentials</StatusCode> </element> </root>
JSON	[{"StatusCode": "NoCredentials"}]

Опции сервера

Опции сервера реализованы для настройки сервера через HTTP-запросы. Опции сервера хранятся в БД и могут быть доступными (открытая опция) или недоступными для настройки посредством HTTP-запросов (закрытая опция). В закрытых опциях хранятся логин и пароль администратора, а также опция, обозначающая произведённую начальную настройку сервера. Названия и значения опций являются строками.

Таблица 57. Сводная таблица опций

Название	Описание	Тип	Значение по умолчанию
FILE_SIZE_LIMIT	Максимальный размер загружаемого файла (в байтах)	Открытая	1073741824
LIMIT_MESSAGES_FOR_QUERY	Максимальное количество сообщений в скачиваемом списке	Открытая	1000
is_initialized	Наличие переменной обозначает факт произведённой первоначальной настройки сервера	Закрытая	Не определено
admin_login	Логин администратора	Закрытая	Не определено
admin_hashed_password	Хешированный пароль администратора	Закрытая	Не определено
admin_salt	Соль, используемая при проверки пароля администратора	Закрытая	Не определено

Безопасность использования сервера OpenTelega

ВНИМАНИЕ

Любая информация, передаваемая через протокол *HTTP*, может быть прочитана или изменена третьей стороной (злоумышленником). Для предотвращения данной уязвимости рекомендуется использовать *HTTPS* протокол с использованием соответствующего сертификата (подписанного УЦ [2: Удостоверяющий центр], или другим доверенным источником). Данная рекомендация не относится к тем случаям, в которых безопасность обмена информацией обеспечивается архитектурой сети (например, *i2p*). В таких случаях можно использовать протокол *HTTP*.

ВНИМАНИЕ

Для снижения эффективности применения злоумышленником уязвимостей веб-серверов (например, *apache* или *nginx*) рекомендуется использовать HTTP-аутентификацию на веб-сервере.

ВНИМАНИЕ

При написании собственных мессенджеров рекомендуется реализовать сквозное шифрование, поскольку даже при использовании безопасного протокола *HTTPS* злоумышленник, в теории, может просматривать содержимое на сервере. Если Вы не специалист в области написания криптографических приложений, рекомендуется использовать уже готовые программы/библиотеки. Автор данного документа предпочитает использовать *GnuPG* или *OpenSSL*. В данных программах используются надёжные и проверенные алгоритмы шифрования, а также безопасные функции генерации ключа нужной длины по паролю, введённому с клавиатуры пользователем.

FAQ

Где можно применить данный сервер?

Спектр применения самый разный: коммуникационные приложения, роботы, аналогичные Telegram-ботам, для личного использования, роботы для устройств "Интернета вещей". Преимущество данного приложения перед аналогами — *OpenTelega* не требует номера телефона при регистрации, не собирает никаких данных, не показывает рекламу, в теории, можно написать клиент данного сервера под любую платформу, начиная с *DOS* и заканчивая современными операционными системами *Windows*, *Linux*, *Mac* и др. Плюсом также может служить то, что данный сервер относительно просто поднять в анонимных сетях, таких как, например, *Tor* или *I2P*.

Как посмотреть список платформ, под которые можно написать клиент?

В данном документе приведены примеры для *curl* в качестве программы, отправляющей HTTP-запросы на сервер. На <https://curl.haxx.se/download.html> находятся ссылки, по которым можно скачать себе *curl*. Также не стоит забывать, что *curl* далеко не единственная программа, позволяющая отправлять HTTP-запросы. Например, под *.NET* существует целых три стандартных библиотеки для работы с *HTTP*:

1. *HttpRequest*
2. *WebClient*
3. *HttpClient*

Автор данного документа предпочитает использовать *curl*, поскольку в данной программе очень просто указывать дополнительные параметры, вроде проху-сервера или сертификатов *SSL*.

Как определить, завершилась ли команда с ошибкой или без неё?

Нужно смотреть в заголовки ответа *HTTP*. Любая команда, которая была успешно выполнена на сервере, возвращает код **200**, с ошибкой — код **400**. Пример *cURL*-запроса с получением кода *HTTP* приведён в [функции скачивания файла администратором](#).

У меня сервер вернул код HTTP 500. Что это значит?

Проверьте адрес команды (параметр *GET*, в конце там должен быть "/" — слеш)