

# Управление приложениями

- Приложения
  - Команда show apps
  - Команда add app
  - Команда update app
  - Команда remove app
- REST-методы приложений
  - Команда showapp-rest-methods
  - Команда add app-rest-method
  - Команда remove app-rest-method
- Примеры
  - Добавление приложения для существующего пользователя bob
  - Изменение приложения, добавление данных для HTTP аутентификации, изменение HTTP адреса
  - Передаем приложение другому пользователю

## Приложения

Приложения хранят REST URL web-сервера, с которым осуществляется интеграция. Кроме того приложения хранят другие настройки и данные:

| Поле          | Описание                                                                                                   |
|---------------|------------------------------------------------------------------------------------------------------------|
| name          | Название приложения.                                                                                       |
| appKey        | Уникальный идентификатор, который будет использоваться при коннекте с этим приложением со стороны клиента. |
| url           | REST URL для интеграции с Web-сервером по WCS RESTful API.                                                 |
| owner         | Пользователь которому принадлежит приложение                                                               |
| mainClass     | Основной класс для прямых вызовов. Для глубокой интеграции и server-side разработчиков.                    |
| callbackClass | Основной класс для обратных вызовов. Для глубокой интеграции и server-side разработчиков.                  |
| restUsername  | Имя пользователя для HTTP аутентификации REST.                                                             |
| restPassword  | Пароль пользователя для HTTP аутентификации REST.                                                          |

Описания приложений хранятся в файле [database.yml](#).

Управлять приложениями можно при помощи следующих команд:

- show apps
- add app
- update app
- remove app

## Команда show apps

Используется для вывода существующих приложений в системе. Пользователю будет показан список, содержащий только принадлежащие ему приложения.

Параметры:

[-d | --detail] включить детализированный вывод

Пример:

```
show apps
show apps -d
```

## Команда add app

Используется для добавления нового приложения в систему. По умолчанию новое приложение будет принадлежать пользователю который его создал.

Параметры:

<app name> имя приложения, обязательный параметр  
<app key> уникальный ключ приложения, обязательный параметр  
<app url> REST URL для интеграции с Web-сервером по WCS RESTful API, обязательный параметр  
[-m | --m-class] основной класс для прямых вызовов, необязательный параметр  
[-c | --c-class] основной класс для обратных вызовов, необязательный параметр  
[-u | --u-rest] имя пользователя для HTTP аутентификации, необязательный параметр  
[-p | --p-rest] пароль пользователя для HTTP аутентификации, необязательный параметр  
[-o | --owner] пользователь которому должно принадлежать приложение, необязательный параметр

Пример:

```
add app MyApp myNewUniqueKey "http://localhost/my_app"  
add app -u rest_http_user -p rest_password MyApp2 myNewKey2 "http://localhost/my_app2"  
add app -o alice MyApp3 myNewKey3 "http://localhost/my_app3"
```

## Команда update app

Используется для изменения существующего приложения в системе. Пользователь может изменить только свои приложения.

Параметры:

<app name> имя приложения, обязательный параметр  
[-k | --key] уникальный ключ приложения  
[-l | --url] REST URL для интеграции с Web-сервером по WCS RESTful API  
[-m | --m-class] основной класс для прямых вызовов  
[-c | --c-class] основной класс для обратных вызовов  
[-u | --u-rest] имя пользователя для HTTP аутентификации  
[-p | --p-rest] пароль пользователя для HTTP аутентификации  
[-o | --owner] пользователь которому должно принадлежать приложение

Пример:

```
update app -k newAppKey -u new_rest_user -p new_rest_password MyApp  
update app -o admin MyApp  
update app -l "http://10.10.10.10/app" MyApp
```

## Команда remove app

Используется для удаления приложения. Приложение может удалить только пользователь которому оно принадлежит. Параметры:  
<app name> имя приложения, обязательный параметр

Пример:

```
remove app MyApp
```

## REST-методы приложений

Управлять REST-методами определенного приложения можно при помощи следующих команд:

- add app-rest-method
- show app-rest-methods
- removeapp-rest-method

## Команда showapp-rest-methods

Используется для вывода существующих REST-методов приложения.

Параметры:

<app key> ключ приложения, обязательный параметр

Пример:

```
show app-rest-methods defaultApp
```

## Команда add app-rest-method

Используется для добавления метода к приложению.

Параметры:

<app key> уникальный ключ приложения, обязательный параметр

<rest method> добавляемый REST метод, обязательный параметр

[-a | --all] добавить все REST методы, необязательный параметр, в этом случае параметр <rest method> не указывается

Пример:

```
add app-rest-method MyAppKey connect
add app-rest-method -a MyAppKey
```

## Команда remove app-rest-method

Используется для удаления REST методов из приложения.

Параметры:

<app key> уникальный ключ приложения, обязательный параметр

<rest method> удаляемый REST метод, обязательный параметр

[-a | --all] удалить все REST методы, необязательный параметр, в этом случае параметр <rest method> не указывается

Пример:

```
remove app-rest-method MyAppKey connect
remove app-rest-method -a MyAppKey
```

## Примеры

### Добавление приложения для существующего пользователя bob

```
add app -o bob BobsApp bobAppKey "http://bobs_web_server.example.com/app"
```

### Изменение приложения, добавление данных для HTTP аутентификации, изменение HTTP адреса

```
update app -u bobs_app -p paSsw0rd BobsApp
update app -l "https://bobs_new_server.example.com/secureApp" BobsApp
```

### Передаем приложение другому пользователю

```
update app -o alice BobsApp
```