

Управление приложениями

Приложения

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

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

```
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`
- `remove app-rest-method`

Команда show app-rest-methods

Используется для вывода существующих REST-методов приложения.

Параметры:

- `app key` - уникальный 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
```