Версия 2025.2.5
© 2008-2025 ООО Быстрые отчеты

Руководство программиста МоиОтчеты Облако

Предисловие

Любые фрагменты кода, которые могут быть встречены в этой документации, являются только демонстрационными примерами. Эти примеры служат только одной цели - проиллюстрировать применение продукта МоиОтчеты Облако. Компания "Быстрые отчеты" не гарантирует правильность и полноту данных примеров и не несет ответственности за прямой или косвенный ущерб, который может быть ими принесен.

Документация может содержать гиперссылки на различные ресурсы в сети Интернет. Эти ссылки актуальны на момент написания документации. Компания "Быстрые отчеты" не несет ответственности за их доступность на момент прочтения документации или нанесенный ими ущерб.

Руководства

Этот раздел посвящен руководствам по работе с облаком и интеграцией МоиОтчеты Облако в приложение. На данный момент доступны руководства для:

- REST API
- C#/.NET

Руководства для использования с другими языками программирования в планах.

Напишите нам если вас интересует какой-то конкретный язык программирования или платформа. Написать!

Инструменты разработки

Доступны SDK для следующих языков:

- Haskell
- JavaScript
- C++
- Python
- Java
- Go
- C#/.NET и ASP.NET

Обратите внимание, инструменты разработки выложены исходниками на Github, что бы их использовать, из исходников необходимо будет собрать пакет или библиотеку самостоятельно.

REST API

В этом разделе приведены пошаговые инструкции и руководства для выполнения типовых задач по использованию МоиОтчеты Облако без привязки к конкретному языку программирования.

Приступая к работе

Вам понадобятся следующие инструменты и возможности:

1. Инструмент curl.

   Подойдёт любой другой REST клиент, но примеры будут построены для curl.

2. Дизайнер отчётов FastReport Community Designer.

   Для некоторых руководств нужен будет дизайнер для создания отчёта.

3. Активная подписка МоиОтчеты Облако.

4. Доступ в интернет.

> Обратите внимание! Эти руководства рассчитаны на то, что вы уже знаете, как использовать curl или использовать другой REST клиент.

> Обратите внимание! Пункты выше описывают рекомендуемые инструменты.

Что дальше?

Советуем начать своё ознакомление со статей:

- Статус коды.
- Аутентификация и авторизация.

Статус коды

В этой статье перечислены все статус-коды, которые могут быть возвращены сервисами МоиОтчеты Облако.

Запрос успешно обработан — 2xx

- 200 OK - запрос обработан успешно, используется в большинстве случаев, когда вместе с результатом возвращено тело ответа.
- 201 Created - запрос обработан успешно, в результате его выполнения была создана новая сущность.
- 204 No Content - запрос обработан успешно, возвращено пустое тело ответа.

Ошибка пользователя — 4xx

- 400 Bad Request - запрос содержит в себе ошибки, например:
  - Отсутствует параметр, необходимый для выполнения запроса.
  - Формат `Id` сущности отличается от Hex 24.
  - В запросе отрицательный параметр skip или take.

- 401 Unauthorized - запрос пришёл от неавторизованного пользователя.
- 402 Payment Required - запрашиваемый ресурс ограничен по плану подписки.
- 403 Forbidden - запрашиваемый ресурс найден, но у пользователя нет прав на действие.
- 404 Not Found - запрашиваемый ресурс не найден.
- 413 Request Entity Too Large - тело запроса превышает лимиты в настройках сервиса.

Ошибка сервера — 5xx

- 500 Internal Server Error - запрос прошёл все проверки на валидность, но в процессе его выполнения возникло исключение Exception. В этом случае просим сообщить, написав по адресу - support@fastreport.ru

Другое

Промежуточные сервисы между МоиОтчеты Облако и пользователем могут возвращать другие статус коды.

Аутентификация и авторизация

Процесс проверки подлинности пользовательских данных и выдача пользователю определённых прав в МоиОтчеты Облако осуществляется по одному из двух доступных способов:

1. Через JWT token.

   В этом случае аутентификацию нужно пройти лично, а токен будет действовать только 5 минут, за которые пользователь должен войти в своё приложение. При подключении к серверу браузер перенаправит на сервер аутентификации, после чего сгенерирует токен доступа. С точки зрения безопасности мы ограничиваем возможность получения JWT токена только лично пользователем.

   > Если в течении 5 минут пользователь не зашёл в приложение, то аутентификацию необходимо пройти заново. Если пользователь зашёл в приложение, повторная аутентификация не требуется.

2. Через API key.

   В этом случае процесс получения прав доступа осуществляется для серверных приложений. Для получения ключа доступа (API key) необходимо присутствие пользователя. Однако сам ключ может действовать продолжительное время, например, год.

Получение первого API key

Для получения первого API key воспользуйтесь пользовательской панелью. Если по какой то причине доступа к пользовательской панели нет, можно запросить ключ по описанию ниже.

1. Откройте ссылку в браузере: https://облако.моиотчеты.рф/account/signin?r=https://облако.моиотчеты.рф/api/manage/v1/ApiKeys.

   Переход по этой ссылке направит вас на автоматический процесс аутентификации браузера.

2. Теперь когда аутентификация пройдена, необходимо запросить новый ключ.

   Нажмите `F12` или `Ctrl+Shift+I`, что бы открыть панель разработчика. Сочетание клавиш могут отличаться от стандартных, в этом случае откройте панель разработчика через меню браузера.

3. Скопируйте и выполните код в консоли JavaScript.

   Этот код сделает `POST` запрос на URL `https://облако.моиотчеты.рф/api/manage/v1/ApiKeys` для создания нового ключа доступа до 2030 года.

4. Обновите страницу браузера и заберите результат.

   ```
   {
     "apiKeys": [
       {
         "value": "cc355oeu1z5d5wncayo33me6c1g5junqdqk4pkupid7t8ynjshey",
         "description": "Generated by js develop panel",
         "expired": "2030-01-01T07:41:23.399Z"
       }
     ],
     "count": 1
   }
   ```

Теперь вы можете использовать API key, в случае выше это `cc355oeu1z5d5wncayo33me6c1g5junqdqk4pkupid7t8ynjshey`.

Повторно получать новый API key через браузер нет необходимости.

Как использовать API key

Ключ следует передавать с каждым запросом в заголовке `Authorization: Basic`. В качестве имени пользователя следует использовать `apikey`, а в качестве пароля значение ключа. Например.

```
Authorization: Basic Base64Encode(apikey:cc355oeu1z5d5wncayo33me6c1g5junqdqk4pkupid7t8ynjshey);
```

Где `Base64Encode` это функция преобразования строки в `base64` при использовании кодировки `UTF8`.

Получение нового API key

Для получения нового ключа сделайте `POST` запрос на точку входа `https://облако.моиотчеты.рф/api/manage/v1/ApiKeys` и передайте JSON в теле запроса по схеме ниже.

```
{
   "description": "string",
   "expired": "string($date-time)"
}
```

Пример запроса.

```
curl -X POST "https://облако.моиотчеты.рф/api/manage/v1/ApiKeys" -H "accept: text/plain" -H "authorization: Basic YXBpa2V5OmNjMzU1b2V1MXo1ZDV3bmNheW8zM21lNmMxZzVqdW5xZHFrNHBrdXBpZDd0OHluanNoZXk=" -H "Content-Type: application/json-patch+json" -d "{ \"description\": \"Generated by js develop panel\", \"expired\": \"2030-01-01T07:41:23.399Z\"}"
```

Схема ответа.

```
{
 "value": "string",
 "description": "string",
 "expired": "2020-12-02T08:47:43.270Z"
}
```

Что дальше?

- Загрузка нового шаблона.
- Работа с группами.
- Добавление новых пользователей в рабочее пространство.

Загрузка нового шаблона

Одна из основных возможностей МоиОтчеты Облако — это хранение шаблонов, отчётов и других данных в облаке. В этой статье будет рассмотрен способ загрузить свой готовый шаблон в хранилище шаблонов МоиОтчеты Облако.

Приступая к работе

Вам понадобятся следующие инструменты и возможности:

1. Знания по использованию API key в МоиОтчеты Облако.

   В этой статье будет пропущена дополнительная информация по аутентификации и авторизации.

2. Инструмент curl.

   Подойдёт любой другой REST клиент, но примеры будут построены для curl.

3. Шаблон отчёта.

   Его можно подготовить с помощью бесплатной программы FastReport Community Designer.

4. Активная подписка для МоиОтчеты Облако.

5. Доступ в интернет.

Инструкция

1. Вам необходимо получить идентификатор корневой папки рабочего пространства. Этот идентификатор никогда не будет меняться, поэтому его можно сохранить и не делать запрос каждый раз перед загрузкой нового шаблона.

   Что бы запросить корневую директорию выполните `GET` запрос на `https://облако.моиотчеты.рф/api/rp/v1/Templates/Root`.

   ```
   curl -X GET "https://облако.моиотчеты.рф/api/rp/v1/Templates/Root" -H "accept: text/plain"
   ```

   В этом случае будет возвращена директория для рабочего пространство по умолчанию. Можно указать конкретное рабочее пространство передав параметр `subscriptionId`.

   ```
   curl -X GET "https://облако.моиотчеты.рф/api/rp/v1/Templates/Root?subscriptionId=your_workspace_id" -H "accept: text/plain"
   ```

   Пример ответа.

```
{
  "name": "RootFolder",
  "parentId": null,
  "tags": [],
  "icon": "",
  "type": "Folder",
  "size": 16384,
  "subscriptionId": "5fa919fa292a8300019349bc",
  "status": "None",
  "id": "5fa919f9292a8300019349b9",
  "createdTime": "2020-11-09T10:29:13.928Z",
  "creatorUserId": "5af5a8dc-8cb0-40f9-ac99-ca2533fa4491",
  "editedTime": "2020-11-13T15:58:45.69Z",
  "editorUserId": "5af5a8dc-8cb0-40f9-ac99-ca2533fa4491"
}
```


Идентификатор директории из примера выше `5fa919f9292a8300019349b9`.

2. Для загрузки нужного шаблона выполните `POST` запрос на `https://облако.моиотчеты.рф/api/rp/v1/Templates/Folder/{id}/File`, где вместо `{id}` следует подставить идентификатор директории, а в теле запроса передать JSON по схеме ниже.

```
{
  "name": "string",
  "content": "base64encoded(template.frx)"
}
```

   - `name` — название файла, который будет отображаться в пользовательской панели.

     К названию файла добавьте расширение `.frx` вручную.

   - `content` — содержимое файла закодированное в base64.

Пример запроса.

```
curl -X POST "https://облако.моиотчеты.рф/api/rp/v1/Templates/Folder/5fa919f9292a8300019349b9/File" -H "accept: text/plain" -H "Content-Type: application/json-patch+json" -d "{ \"name\": \"template.frx\", \"content\": \"77u/PD94bWwgdmVyc2lvbj0iMS4wIiBlbmNvZGluZz0idXRmLTgiPz4NCjxSZXBvcnQgU2NyaXB0TGFuZ3VhZ2U9IkNTaGFycCIgUmVwb3J0SW5mby5DcmVhdGVkPSIxMi8wNC8yMDIwIDEwOjU4OjU3IiBSZXBvcnRJbmZvLk1vZGlmaWVkPSIxMi8wNC8yMDIwIDExOjAwOjIwIiBSZXBvcnRJbmZvLkNyZWF0b3JWZXJzaW9uPSIyMC4yMC40LjEiPg0KICA8RGljdGlvbmFyeS8+DQogIDxSZXBvcnRQYWdlIE5hbWU9IlBhZ2UxIiBXYXRlcm1hcmsuRm9udD0iQXJpYWwsIDYwcHQiPg0KICAgIDxSZXBvcnRUaXRsZUJhbmQgTmFtZT0iUmVwb3J0VGl0bGUxIiBXaWR0aD0iNzE4LjIiIEhlaWdodD0iMzcuOCIvPg0KICAgIDxQYWdlSGVhZGVyQmFuZCBOYW1lPSJQYWdlSGVhZGVyMSIgVG9wPSI0MSIgV2lkdGg9IjcxOC4yIiBlZWlnaHQ9IjI4LjM1Ii8+DQogICAgPERhdGFCYW5kIE5hbWU9IkRhdGExIiBUb3A9IjcyLjU1IiBXaWR0aD0iNzE4LjIiIEhlaWdodD0iNzUuNiI+DQogICAgICA8VGV4dE9iamVjdCBOYW1lPSJUZXh0MSIgV2lkdGg9IjcxOC4yIiBlZWlnaHQ9Ijc1LjYiIFRleHQ9IkhlbGxvLCBGYXN0UmVwb3J0IENsb3VkISEhIiBIb3J6QWxpZ249IkNlbnRlciIgVmVydEFsaWduPSJDZW50ZXIiIEZvbnQ9IkFyaWFsLCAxMHB0Ii8+DQogICAgPC9EYXRhQmFuZD4NCiAgICA8UGFnZUZvb3RlckJhbmQgTmFtZT0iUGFnZUZvb3RlcjEiIFRvcD0iMTUxLjM1IiBXaWR0aD0iNzE4LjIiIEhlaWdodD0iMTguOSIvPg0KICA8L1JlcG9ydFBhZ2U+DQo8L1JlcG9ydD4NCg==\"}"
```

Пример ответа.

```
{
  "reportInfo": {
    "author": null,
    "created": "2020-12-04T10:58:57",
    "creatorVersion": "20.20.4.1",
    "description": null,
    "modified": "2020-12-04T11:00:20",
    "name": null,
    "picture": null,
    "previewPictureRatio": 0,
    "saveMode": "All",
    "savePreviewPicture": false,
    "tag": null,
    "version": null
  },
  "name": "template.frx",
  "parentId": "5fa919f9292a8300019349b9",
  "tags": null,
  "icon": null,
  "type": "File",
  "size": 17159,
  "subscriptionId": "5fa919fa292a8300019349bc",
  "status": "Success",
  "id": "5fc9ece6b792c90001d94b13",
  "createdTime": "2020-12-04T08:01:42.7087229+00:00",
  "creatorUserId": "5af5a8dc-8cb0-40f9-ac99-ca2533fa4491",
  "editedTime": "2020-12-04T08:01:42.7087112+00:00",
  "editorUserId": "5af5a8dc-8cb0-40f9-ac99-ca2533fa4491"
}
```


3. Теперь этот шаблон можно использовать для подготовки (построения) отчёта и экспорта.

Что дальше?

- Управление правами доступа на примере шаблона.
- Построение отчёта.

Добавление источника данных JSON

В этой статье будет рассмотрен способ создать новый источник данных на основе JSON в МоиОтчеты Облако.

Приступая к работе

Вам понадобятся следующие инструменты и возможности:

1. Знания по использованию API key в МоиОтчеты Облако.

   В этой статье будет пропущена дополнительная информация по аутентификации и авторизации.

2. Инструмент curl.

   Подойдёт любой другой REST клиент, но примеры будут построены для curl.

3. JSON файл.

4. JSON схема.

5. Активная подписка для МоиОтчеты Облако.

6. Доступ в интернет.

Создание источника данных

Для того, чтобы создать источник данных, отправьте `CREATE` запрос на `https://облако.моиотчеты.рф/api/data/v1/DataSources` .

Пример тела запроса:


```
{
  "name": "fakeAPI",
  "connectionString":
"Json=aHR0cHM6Ly9qc29ucGxhY2Vob2xkZXIudHlwaWNvZGUuY29tL3Bvc3Rz;JsonSchema=eyJ0eXBlIjoiYXJyYXkiLCJpdGVtcyI
6eyJ0eXBlIjoib2JqZWN0IiwicHJvcGVydGllcyI6eyJ1c2VySWQiOnsidHlwZSI6Im51bWJlciJ9LCJpZCI6eyJ0eXBlIjoibnVtYmVyIn0sInR
pdGxlIjp7InR5cGUiOiJzdHJpbmcifSwiYm9keSI6eyJ0eXBlIjoic3RyaW5nIn19fX0=;Encoding=utf-8",
  "subscriptionId": "604f52e1261a3c19104c0e25",
  "connectionType": "JSON"
}
```


Где

- `name` — имя источника данных (будет отображаться в дизайнере при выборе).
- `connectionString` — строка подключения, в случае с JSON она состоит из 3-х элементов:
  - `Json` — JSON файл или http/https ссылка закодировання в base64;
  - `JsonSchema` — схема, описывающая структуру JSON файла, закодировання в base64;
  - `Encoding` — кодировка, стоит всегда передавать `utf-8`.
- `subscriptionId` — id рабочего пространство(подписки), к которому будет прикреплён источник данных.
- `connectionType` — тип подключения, в этом руководстве используется `JSON`.

Пример запроса.

```
curl -X POST "https://облако.моиотчеты.рф/api/data/v1/DataSources" -H "accept: text/plain" -H "Content-Type:
application/json-patch+json" -d "{ \"name\": \"fakeAPI\", \"connectionString\":
\"Json=aHR0cHM6Ly9qc29ucGxhY2Vob2xkZXIudHlwaWNvZGUuY29tL3Bvc3Rz;JsonSchema=eyJ0eXBlIjoiYXJyYXkiLCJpdGVtcy
I6eyJ0eXBlIjoib2JqZWN0IiwicHJvcGVydGllcyI6eyJ1c2VySWQiOnsidHlwZSI6Im51bWJlciJ9LCJpZCI6eyJ0eXBlIjoibnVtYmVyIn0sInR
pdGxlIjp7InR5cGUiOiJzdHJpbmcifSwiYm9keSI6eyJ0eXBlIjoic3RyaW5nIn19fX0=;Encoding=utf-8\", \"subscriptionId\":
\"604f52e1261a3c19104c0e25\", \"connectionType\": \"JSON\"}"
```

Пример ответа.


```
{
  "id": "60648953db44d83f9c6da98f",
  "name": "fakeAPI",
  "connectionType": "JSON",
  "connectionString":
"Json=aHR0cHM6Ly9qc29ucGxhY2Vob2xkZXIudHlwaWNvZGUuY29tL3Bvc3Rz;JsonSchema=eyJ0eXBlIjoiYXJyYXkiLCJpdGVtcyI
6eyJ0eXBlIjoib2JqZWN0IiwicHJvcGVydGllcyI6eyJ1c2VySWQiOnsidHlwZSI6Im51bWJlciJ9LCJpZCI6eyJ0eXBlIjoibnVtYmVyIn0sInR
pdGxlIjp7InR5cGUiOiJzdHJpbmcifSwiYm9keSI6eyJ0eXBlIjoic3RyaW5nIn19fX0=;Encoding=utf-8",
  "dataStructure": "<JsonDataSourceConnection
ConnectionString=\"rijcmlqrcq6OJBTPt0pNFvRuRtDUSHSHLQy/QlNolifTTaTjsrExzdbf1ifpPblp655sducwkD1lEVzxVZF8qRuE0NT
6UkyTr7kwjGItFOwh7DBsOyL6QkQY4FOZ2ki8Al2R30gpXs6nMUGg1BRwCF0rj3+QvmXbj+2t8x5RerR5y7inP1R+oCuo0wvfcTe
OMfYfZrjdE3whziFh5Qn3mR7vaevmV9peDWQ3LYyK2ec3KpGVeEXSqM+10WyL4ahY7EHuQlzIZROGFGKfW50cUYwdillhKy24g
NdsUzi5klG66DDQtCKEOLbNutDvA0xqCTW3MvRNORSbvckL6g3gM+cStJ5PQ2XUjF9yz9zdwmramnXI6k+MK8V9lrMkc0XFkDM
HOxDlfG2jHhkFuUTgmiKp7hQMg==\">\r\n  <JsonTableDataSource Name=\"JSON\"
DataType=\"FastReport.Data.JsonConnection.JsonParser.JsonArray\" Enabled=\"true\" TableName=\"JSON\">\r\n    <Column
Name=\"index\" DataType=\"System.Int32\"/>\r\n    <Column Name=\"item\" DataType=\"FastReport.JsonBase\">\r\n
<Column Name=\"userId\" DataType=\"System.Double\"/>\r\n      <Column Name=\"id\" DataType=\"System.Double\"/>\r\n
<Column Name=\"title\" DataType=\"System.String\"/>\r\n      <Column Name=\"body\" DataType=\"System.String\"/>\r\n
</Column>\r\n    <Column Name=\"array\" DataType=\"FastReport.JsonBase\"/>\r\n
</JsonTableDataSource>\r\n</JsonDataSourceConnection>\r\n",
  "subscriptionId": "604f52e1261a3c19104c0e25",
  "editedTime": "2021-03-31T14:38:10.5792982Z",
  "editorUserId": "2df79f83-07f1-41ba-96b5-7757bbf377df",
  "createdTime": "0001-01-01T00:00:00",
  "creatorUserId": "2df79f83-07f1-41ba-96b5-7757bbf377df",
  "isConnected": true
}
```

Управление правами доступа на примере шаблона

Ограничение доступа к приватным ресурсам очень важная возможность МоиОтчеты Облако. Гибкая система доступа позволяет задать ограничение или выдать права на каждый ресурс отдельно, задавая круг лиц, которые могут получить доступ.

Приступая к работе

Вам понадобятся следующие инструменты и возможности:

1. Знания по использованию API key в МоиОтчеты Облако.

   В этой статье будет пропущена дополнительная информация по аутентификации и авторизации.

2. Инструмент curl.

   Подойдёт любой другой REST клиент, но примеры будут построены для curl.

3. Шаблон отчёта.

   О том, как загрузить шаблон отчёта, можно узнать в статье Загрузка нового шаблона.

4. Активная подписка для МоиОтчеты Облако.

5. Доступ в интернет.

Инструкция

1. Для просмотра текущих прав на ресурс выполните `GET` запрос на `https://облако.моиотчеты.рф/api/rp/v1/Templates/File/{id}/permissions`, где вместо `{id}` следует использовать идентификатор шаблона. пример запроса.

   Пример запроса.

   ```
   curl -X GET "https://облако.моиотчеты.рф/api/rp/v1/Templates/File/5fc9ece6b792c90001d94b13/permissions" -H "accept: text/plain"
   ```

   Пример ответа.

```
{
  "permissions": {
    "ownerId": "5af5a8dc-8cb0-40f9-ac99-ca2533fa4491",
    "owner": {
      "create": "All",
      "delete": "All",
      "execute": "All",
      "get": "All",
      "update": "All",
      "administrate": "All"
    },
    "groups": null,
    "other": {
      "create": "All",
      "delete": "All",
      "execute": "All",
      "get": "All",
      "update": "All",
      "administrate": "All"
    },
    "anon": null
  }
}
```

В этом примере владелец и остальные участники рабочего пространства имеют полный доступ к шаблону отчёта.

2. Для того чтобы изменить разрешения сделайте `POST` запрос на `https://облако.моиотчеты.рф/api/rp/v1/Templates/File/{id}/permissions`, где вместо `{id}` следует использовать идентификатор шаблона.

   Пример модели.

```
{
  "newPermissions": {
    "ownerId": "5af5a8dc-8cb0-40f9-ac99-ca2533fa4491",
    "owner": {
      "create": -1,
      "delete": -1,
      "execute": -1,
      "get": -1,
      "update": 0,
      "administrate": -1
    },
    "other": {
      "create": 0,
      "delete": 0,
      "execute": 0,
      "get": 0,
      "update": 0,
      "administrate": 0
    }
  },
  "administrate": "Owner,Other"
}
```

Обратите внимание! В запросе на обновление прав в свойстве `administrate` были переданы значения `Owner` и `Other`, в этом случае будут обновлены только указанные категории прав, т.е. `owner` и `other`, а категории `anon` и `groups` изменены не будут.

Пример запроса, который убирает у владельца право на обновление отчёта (например - редактирование в онлайн дизайнере), а также все права на файл у остальных участников

рабочего пространства.

```
curl -X POST "https://облако.моиотчеты.рф/api/rp/v1/Templates/File/5fc9ece6b792c90001d94b13/permissions" -H "accept: text/plain" -H "Content-Type: application/json-patch+json" -d "{ \"newPermissions\": { \"ownerId\": \"5af5a8dc-8cb0-40f9-ac99-ca2533fa4491\", \"owner\": { \"create\": -1, \"delete\": -1, \"execute\": -1, \"get\": -1, \"update\": 0, \"administrate\": -1 }, \"other\": { \"create\": 0, \"delete\": 0, \"execute\": 0, \"get\": 0, \"update\": 0, \"administrate\": 0 } }, \"administrate\": \"Owner,Other\"}
```

Пример ответа.

```
200 OK
```

3. Теперь можно снова запросить права, чтобы убедиться, что они действительно обновились.

   Пример запроса.

```
curl -X GET "https://облако.моиотчеты.рф/api/rp/v1/Templates/File/5fc9ece6b792c90001d94b13/permissions" -H "accept: text/plain"
```

   Пример ответа.

```
{
 "permissions": {
   "ownerId": "2df79f83-07f1-41ba-96b5-7757bbf377df",
 "owner": {
     "create": "All",
     "delete": "All",
     "execute": "All",
  "get": "All",
     "update": "None",
  "administrate": "All"
   },
   "groups": null,
   "other": {
     "create": "None",
     "delete": "None",
     "execute": "None",
     "get": "None",
     "update": "None",
     "administrate": "None"
   },
   "anon": null
  }
}
```

Что дальше?

- Построение отчёта.
- Работа с группами.
- Добавление новых пользователей в рабочее пространство.

Построение отчёта

В этой статье рассмотрен процесс построения отчёта из шаблона с помощью процессора отчётов МоиОтчеты Облако.

Приступая к работе

Вам понадобятся следующие инструменты и возможности:

1. Знания по использованию API key в МоиОтчеты Облако.

   В этой статье будет пропущена дополнительная информация по аутентификации и авторизации.

2. Инструмент curl.

   Подойдёт любой другой REST клиент, но примеры будут построены для curl.

3. Шаблон отчёта.

   Как загрузить шаблон отчёта можно узнать в статье Загрузка нового шаблона.

4. Активная подписка МоиОтчеты Облако.

5. Доступ в интернет.

Инструкция

1. Вам необходим идентификатор шаблона для его построения. Сделайте `GET` запрос на `https://облако.моиотчеты.рф/api/rp/v1/Templates/Root` для получения корневой директории.

   Пример запроса.

   ```
   curl -X GET "https://облако.моиотчеты.рф/api/rp/v1/Templates/Root" -H "accept: text/plain"
   ```

   Пример ответа.

   ```
   {
     "name": "RootFolder",
     "parentId": null,
     "tags": [],
     "icon": "",
     "type": "Folder",
     "size": 16384,
     "subscriptionId": "5fa919fa292a8300019349bc",
     "status": "None",
     "id": "5fa919f9292a8300019349b9",
     "createdTime": "2020-11-09T10:29:13.928Z",
     "creatorUserId": "5af5a8dc-8cb0-40f9-ac99-ca2533fa4491",
     "editedTime": "2020-11-13T15:58:45.69Z",
     "editorUserId": "5af5a8dc-8cb0-40f9-ac99-ca2533fa4491"
   }
   ```

   Идентификатор директории из примера выше `5fa919f9292a8300019349b9`.

2. Для получения списка файлов в директории выполните `GET` запрос на `https://облако.моиотчеты.рф/api/rp/v1/Templates/Folder/{id}/ListFiles?skip=0&take=10`, где вместо `{id}` следует подставить идентификатор директории.

   Пример запроса.

```
curl -X GET "https://облако.моиотчеты.рф/api/rp/v1/Templates/Folder/5fa919f9292a8300019349b9/ListFiles?skip=0&take=10" -H "accept: text/plain"
```

Пример ответа.


```
[
  {
    "reportInfo": {
      "author": null,
      "created": "2020-12-04T10:58:57Z",
      "creatorVersion": "20.20.4.1",
      "description": null,
      "modified": "2020-12-04T11:00:20Z",
      "name": null,
      "picture": null,
      "previewPictureRatio": 0,
      "saveMode": "All",
      "savePreviewPicture": false,
      "tag": null,
      "version": null
    },
    "name": "template.frx",
    "parentId": "5fa919f9292a8300019349b9",
    "tags": null,
    "icon": null,
    "type": "File",
    "size": 17159,
    "subscriptionId": "5fa919fa292a8300019349bc",
    "status": "Success",
    "id": "5fc9ece6b792c90001d94b13",
    "createdTime": "2020-12-04T08:01:42.708Z",
    "creatorUserId": "5af5a8dc-8cb0-40f9-ac99-ca2533fa4491",
    "editedTime": "2020-12-04T08:01:42.708Z",
    "editorUserId": "5af5a8dc-8cb0-40f9-ac99-ca2533fa4491"
  }
]
```


Идентификатор шаблона из примера выше `5fc9ece6b792c90001d94b13`.

3. Для построения отчёта вам понадобится директория, в которую можно положить отчёт. Запросите корневую директорию отчётов, для этого сделайте `GET` запрос на `https://облако.моиотчеты.рф/api/rp/v1/Reports/Root`.

   Пример запроса.

```
curl -X GET "https://облако.моиотчеты.рф/api/rp/v1/Reports/Root" -H "accept: text/plain"
```

   Пример ответа.

```
{
  "name": "RootFolder",
  "parentId": null,
  "tags": null,
  "icon": null,
  "type": "Folder",
  "size": 16384,
  "subscriptionId": "5fa919fa292a8300019349bc",
  "status": "None",
  "id": "5fa919f9292a8300019349ba",
  "createdTime": "2020-11-09T10:29:13.993Z",
  "creatorUserId": "5af5a8dc-8cb0-40f9-ac99-ca2533fa4491",
  "editedTime": "0001-01-01T00:00:00Z",
  "editorUserId": null
}
```

Идентификатор директории из примера выше `5fa919f9292a8300019349ba`.

4. Для построения отчёта сделайте `POST` запрос на `https://облако.моиотчеты.рф/api/rp/v1/Templates/File/{id}/Prepare`, где вместо `{id}` следует подставить идентификатор шаблона.

   В теле запроса передайте JSON по схеме ниже.

```
{
  "name": "string",
  "folderId": "string",
  "reportParameters": {
    "additionalProp1": "string",
    "additionalProp2": "string",
    "additionalProp3": "string"
  }
}
```

   - `folderId` — идентификатор директории куда будет помещён отчёт.
   - `name` — название результирующего файла. Добавите расширение `.fpx` вручную.
   - `reportParameters` — параметры, которые указаны в самом отчёте с помощью дизайнера отчёта, в этом примере они не нужны.

> Если не указать folderId, то подготовленный отчёт будет сохранён в корневую папку.

   Пример запроса.

```
curl -X POST "https://облако.моиотчеты.рф/api/rp/v1/Templates/File/5fc9ece6b792c90001d94b13/Prepare" -H "accept: text/plain" -H "Content-Type: application/json-patch+json" -d "{ \"name\": \"awesome_report.fpx\", \"folderId\": \"5fa919f9292a8300019349ba\"}"
```

   Пример ответа.

```
{
  "templateId": "5fc9ece6b792c90001d94b13",
  "reportInfo": null,
  "name": "awesome_report.fpx",
  "parentId": "5fa919f9292a8300019349ba",
  "tags": null,
  "icon": null,
  "type": "File",
  "size": 16384,
  "subscriptionId": "5fa919fa292a8300019349bc",
  "status": "InQueue",
  "id": "5fe4614bcd7c55000148e4c6",
  "createdTime": "2020-12-24T09:37:15.7169531+00:00",
  "creatorUserId": "5af5a8dc-8cb0-40f9-ac99-ca2533fa4491",
  "editedTime": "2020-12-24T09:37:15.7169582+00:00",
  "editorUserId": "5af5a8dc-8cb0-40f9-ac99-ca2533fa4491"
}
```

Обратите внимание на статус файла `InQueue`, он означает, что была создана задача на построение, она попала в очередь построителя, и уже на этом этапе отчёт получил свой идентификатор для работы `5fe4614bcd7c55000148e4c6`.

Следует подождать некоторое время, пока отчёт будет построен. Можно вызывать метод получения отчёта в цикле каждые несколько миллисекунд и проверять статус.

5. Для получения информации об отчёте сделайте `GET` запрос на `https://облако.моиотчеты.рф/api/rp/v1/Reports/File/{id}`, где вместо `{id}` следует указать идентификатор отчёта.

   Пример запроса.

   ```
   curl -X GET "https://облако.моиотчеты.рф/api/rp/v1/Reports/File/5fe4614bcd7c55000148e4c6" -H "accept: text/plain"
   ```

   Пример ответа.

   ```
   {
     "templateId": "5fc9ece6b792c90001d94b13",
     "reportInfo": null,
     "name": "awesome_report.fpx",
     "parentId": "5fa919f9292a8300019349ba",
     "tags": null,
     "icon": null,
     "type": "File",
     "size": 16927,
     "subscriptionId": "5fa919fa292a8300019349bc",
     "status": "Success",
     "id": "5fe4614bcd7c55000148e4c6",
     "createdTime": "2020-12-24T09:37:15.716Z",
     "creatorUserId": "5af5a8dc-8cb0-40f9-ac99-ca2533fa4491",
     "editedTime": "2020-12-24T09:37:15.716Z",
     "editorUserId": "5af5a8dc-8cb0-40f9-ac99-ca2533fa4491"
   }
   ```

   Обратите внимание на статус файла `Success`. Отчёт был успешно построен.

6. Для скачивания отчёта сделайте `GET` запрос на `https://облако.моиотчеты.рф/download/r/{id}`, где вместо `{id}` следует передать идентификатор отчёта.

   Пример запроса.

   ```
   curl -X GET "https://облако.моиотчеты.рф/download/r/5fe4614bcd7c55000148e4c6" -H "accept: text/plain"
   ```

В ответе будет получен файл.

Что дальше?

- Экспорт отчёта в PDF.

Параметры отчёта

В этой статье рассмотрен процесс передачи параметров в отчёт, которые представляют собой словарь параметров, предоставляемых при формировании отчёта. Больше информации на эту тему можно найти в руководстве FastReport .NET в разделе Параметры отчёта.

Приступая к работе

Вам понадобятся следующие инструменты и возможности:

1. Знания по использованию API key в МоиОтчеты Облако.

   В этой статье будет пропущена дополнительная информация по аутентификации и авторизации.

2. Инструмент curl.

   Подойдёт любой другой REST клиент, но примеры будут построены для curl.

3. Шаблон отчёта с указанными параметрами отчёта.

   Его можно подготовить с помощью бесплатной программы FastReport Community Designer.

4. Активная подписка МоиОтчеты Облако.

5. Доступ в интернет.

Передача параметров в отчёт

Параметры могут быть переданы в следующих случаях:

- при подготовке отчёта;
- при экспорте из шаблона;
- при работе с задачами.

Рассмотрим передачу параметров на примере подготовки отчёта. Подробная инструкция о том, как подготовить отчёт, содержится в разделе Построение отчёта.

Для построения отчёта сделайте `POST` запрос на `https://облако.моиотчеты.рф/api/rp/v1/Templates/File/{id}/Prepare`, где вместо `{id}` следует подставить идентификатор шаблона.

В теле запроса передайте JSON по схеме ниже.

```
{
 "name": "string",
 "folderId": "string",
 "reportParameters": {
  "additionalProp1": "string",
  "additionalProp2": "string"
 }
}
```

- `folderId` — идентификатор директории куда будет помещён отчёт.
- `name` — название результирующего файла. Добавите расширение `.fpx` вручную.
- `reportParameters` — параметры, которые указаны в самом отчёте с помощью дизайнера отчёта.

> Если не указать folderId, то подготовленный отчёт будет сохранён в корневую папку.

Пример запроса.

```
curl -X POST "https://облако.моиотчеты.рф/api/rp/v1/Templates/File/5fc9ece6b792c90001d94b13/Prepare"
-H "accept: text/plain"
-H "Content-Type: application/json-patch+json"
-d "{
 "name": "awesome_report.fpx",
 "folderId": "5fa919f9292a8300019349ba",
 "reportParameters": {
   "Parameter1": "Значение1",
   "Parameter2": "Значение2"
   }
 }"
```

Пример ответа.

```
{
  "templateId": "5fc9ece6b792c90001d94b13",
  "name": "awesome_report.fpx",
  "parentId": "5fa919f9292a8300019349ba",
  "type": "File",
  "size": 16384,
  "subscriptionId": "5fa919fa292a8300019349bc",
  "status": "InQueue",
  "statusReason": "None",
  "id": "5fe4614bcd7c55000148e4c6",
  "createdTime": "2020-12-24T09:37:15.7169531+00:00",
  "creatorUserId": "5af5a8dc-8cb0-40f9-ac99-ca2533fa4491",
  "editedTime": "2020-12-24T09:37:15.7169582+00:00",
  "editorUserId": "5af5a8dc-8cb0-40f9-ac99-ca2533fa4491"
}
```

Экспорт отчёта в PDF

В этой статье рассмотрен процесс экспорта отчёта с помощью процессора отчётов МоиОтчеты Облако.

Приступая к работе

Вам понадобятся следующие инструменты и возможности:

1. Знания по использованию API key в МоиОтчеты Облако. Как получить и использовать API key можно узнать в статье Аутентификация и авторизация

   В этой статье будет пропущена дополнительная информация по аутентификации и авторизации.

2. Инструмент curl.

   Подойдёт любой другой REST клиент, но примеры будут построены для curl.

3. Отчёт.

   Как построить отчёт можно узнать в статье Построение отчёта.

4. Активная подписка МоиОтчеты Облако.

5. Доступ в интернет.

Замечание

Обратите внимание, что экспорт отчёта можно сделать напрямую из шаблона, без промежуточного сохранения отчёта. Для этого выполните те же команды для шаблона отчёта, заменив `Report` в строках запроса на `Template`, также используйте идентификатор шаблона, а не отчёта.

Инструкция

1. Вам будет необходим идентификатор отчёта для экспорта в PDF. Сделайте `GET` запрос на `https://облако.моиотчеты.рф/api/rp/v1/Report/Root` для получения корневой директории.

   Пример запроса.

   ```
   curl -X GET "https://облако.моиотчеты.рф/api/rp/v1/Reports/Root" -H "accept: text/plain"
   ```

   Пример ответа.

   ```
   {
    "name": "RootFolder",
    "parentId": null,
    "tags": null,
    "icon": null,
    "type": "Folder",
    "size": 16384,
    "subscriptionId": "5fa919fa292a8300019349bc",
    "status": "None",
    "id": "5fa919f9292a8300019349ba",
    "createdTime": "2020-11-09T10:29:13.993Z",
    "creatorUserId": "5af5a8dc-8cb0-40f9-ac99-ca2533fa4491",
    "editedTime": "0001-01-01T00:00:00Z",
    "editorUserId": null
   }
   ```

   Идентификатор директории из примера выше `5fa919f9292a8300019349ba`.

2. Получите список файлов в директории, для этого выполните `GET` запрос на `https://облако.моиотчеты.рф/api/rp/v1/Reports/Folder/{id}/ListFiles?skip=0&take=10`, где вместо `{id}` следует подставить идентификатор директории.

   Пример запроса.

   ```
   curl -X GET "https://облако.моиотчеты.рф/api/rp/v1/Reports/Folder/5fa919f9292a8300019349ba/ListFiles?skip=0&take=10" -H "accept: text/plain"
   ```

   Пример ответа.


   ```
   [
     {
       "templateId": "5fc9ece6b792c90001d94b13",
       "reportInfo": null,
       "name": "awesome_report.fpx",
       "parentId": "5fa919f9292a8300019349ba",
       "tags": null,
       "icon": null,
       "type": "File",
       "size": 16927,
       "subscriptionId": "5fa919fa292a8300019349bc",
       "status": "Success",
       "id": "5fe4614bcd7c55000148e4c6",
       "createdTime": "2020-12-24T09:37:15.716Z",
       "creatorUserId": "5af5a8dc-8cb0-40f9-ac99-ca2533fa4491",
       "editedTime": "2020-12-24T09:37:15.716Z",
       "editorUserId": "5af5a8dc-8cb0-40f9-ac99-ca2533fa4491"
     }
   ]
   ```


   Идентификатор отчёта из примера выше `5fe4614bcd7c55000148e4c6`.

3. Для экспорта отчёта понадобится директория, в которую следует положить файл экспорта.

   Получите корневую директорию экспортов, для этого сделайте `GET` запрос на `https://облако.моиотчеты.рф/api/rp/v1/Exports/Root`.

   Пример запроса.

   ```
   curl -X GET "https://облако.моиотчеты.рф/api/rp/v1/Exports/Root" -H "accept: text/plain"
   ```

   Пример ответа.


   ```
   {
     "name": "RootFolder",
     "parentId": null,
     "tags": null,
     "icon": null,
     "type": "Folder",
     "size": 16384,
     "subscriptionId": "5fa919fa292a8300019349bc",
     "status": "None",
     "id": "5fa919fa292a8300019349bb",
     "createdTime": "2020-11-09T10:29:14.002Z",
     "creatorUserId": "5af5a8dc-8cb0-40f9-ac99-ca2533fa4491",
     "editedTime": "0001-01-01T00:00:00Z",
     "editorUserId": null
   }
   ```


   Идентификатор директории из примера выше `5fa919fa292a8300019349bb`.

4. Для экспорта отчёта сделайте `POST` запрос на `https://облако.моиотчеты.рф/api/rp/v1/Reports/File/{id}/Export`, где вместо `{id}` следует подставить идентификатор отчёта.

   В теле запроса передайте JSON по схеме ниже.

```
{
  "fileName": "awesome_result.pdf",
  "folderId": "5fa919fa292a8300019349bb",
  "locale": "en-GB",
  "format": "Pdf",
  "exportParameters": {
    "additionalProp1": {},
    "additionalProp2": {},
    "additionalProp3": {}
  }
}
```

   - `folderId` — идентификатор директории куда будет помещён экспорт. Если оставить пустым, то экспорт будет помещён в корневую папку для экспортов в рабочем пространстве.
   - `fileName` — название результирующего файла. Если не указать расширение, или указать его неправильно - сервер заменит его самостоятельно.
   - `locale` — локализация экспортированного отчёта. Эта опция поменяет форматы даты и чисел, на те, что соответствуют выбранному ISO - коду культуры (например - французский (Швейцария) выглядит так - "fr-CH"). Если оставить это поле пустым или указать несуществующую культуру подставится локаль по умолчанию из подписки или английский (США), если локаль по умолчанию не указана.
   - `format` — формат экспорта.
   - `exportParameters` — параметры экспорта. Задаются аналогично параметрам экспорта из библиотеки FastReport .NET. Более подробное описание хранится в документации пользователя в разделе параметры экспорта.

> Если не указать folderId, то подготовленный отчёт будет сохранён в корневую папку.

   Пример запроса.

```
curl -X POST "https://облако.моиотчеты.рф/api/rp/v1/Reports/File/5fe4614bcd7c55000148e4c6/Export" -H "accept: text/plain" -H "Content-Type: application/json-patch+json" -d "{ \"fileName\": \"awesome_result.pdf\", \"folderId\": \"5fa919fa292a8300019349bb\", \"locale\":\"ru-RU\", \"format\": \"Pdf\"}"
```

   Пример ответа.

```
{
  "format": "Pdf",
  "reportId": "5fe4614bcd7c55000148e4c6",
  "name": "awesome_result.pdf",
  "parentId": "5fa919fa292a8300019349bb",
  "tags": null,
  "icon": null,
  "type": "File",
  "size": 16384,
  "subscriptionId": "5fa919fa292a8300019349bc",
  "status": "InQueue",
  "id": "5fe46a33cd7c55000148e4c7",
  "createdTime": "2020-12-24T10:15:15.8039648+00:00",
  "creatorUserId": "5af5a8dc-8cb0-40f9-ac99-ca2533fa4491",
  "editedTime": "2020-12-24T10:15:15.8039697+00:00",
  "editorUserId": "5af5a8dc-8cb0-40f9-ac99-ca2533fa4491"
}
```

Обратите внимание на статус файла `InQueue`, он означает, что была создана задача на экспорт и она попала в очередь построителя, уже на этом этапе файл получил свой идентификатор для работы `5fe46a33cd7c55000148e4c7`.

Следует подождать некоторое время, пока экспорт будет построен.

5. Для получения информации об файле сделайте `GET` запрос на `https://облако.моиотчеты.рф/api/rp/v1/Exports/File/{id}`, где вместо `{id}` следует указать идентификатор экспорта.

   Пример запроса.

   ```
   curl -X GET "https://облако.моиотчеты.рф/api/rp/v1/Exports/File/5fe46a33cd7c55000148e4c7" -H "accept: text/plain"
   ```

   Пример ответа.

   ```
   {
     "format": "Pdf",
     "reportId": "5fe4614bcd7c55000148e4c6",
     "name": "awesome_result.pdf",
     "parentId": "5fa919fa292a8300019349bb",
     "tags": null,
     "icon": null,
     "type": "File",
     "size": 41142,
     "subscriptionId": "5fa919fa292a8300019349bc",
     "status": "Success",
     "id": "5fe46a33cd7c55000148e4c7",
     "createdTime": "2020-12-24T10:15:15.803Z",
     "creatorUserId": "5af5a8dc-8cb0-40f9-ac99-ca2533fa4491",
     "editedTime": "2020-12-24T10:15:15.803Z",
     "editorUserId": "5af5a8dc-8cb0-40f9-ac99-ca2533fa4491"
   }
   ```

   Обратите внимание на статус файла `Success`. Отчёт был успешно экспортирован.

6. Для скачивания отчёта сделайте `GET` запрос на `https://облако.моиотчеты.рф/download/e/{id}`, где вместо `{id}` следует передать идентификатор отчёта.

   Пример запроса.

   ```
   curl -X GET "https://облако.моиотчеты.рф/download/e/5fe46a33cd7c55000148e4c7" -H "accept: text/plain"
   ```

   В ответе будет получен файл.

Что дальше?

- Работа с группами.
- Добавление новых пользователей в рабочее пространство.

Добавление новых пользователей в рабочее пространство

В этой статье рассмотрен процесс добавления нового пользователя в рабочее пространство, получения списка пользователей в рабочем пространстве и удаление пользователя из рабочего пространства.

> К каждому рабочему пространству всегда привязана подписка. Они имеют общий идентификатор.

Приступая к работе

Вам понадобятся следующие инструменты и возможности:

1. Знания по использованию API key в МоиОтчеты Облако.

   В этой статье будет пропущена дополнительная информация по аутентификации и авторизации.

2. Инструмент curl.

   Подойдёт любой другой REST клиент, но примеры будут построены для curl.

3. Активная подписка для МоиОтчеты Облако, в которой есть хотя бы два слота для пользователя.

4. Доступ в интернет.

Инструкция

Есть два способа добавить пользователя в рабочее пространство:

- Напрямую, указав id пользователя.
- С помощью системы приглашений.

Добавляем пользователя через приглашение

1. Для создания приглашения необходим идентификатор рабочего пространства.

   Получите идентификатор рабочего пространства сделав `GET` запрос на `https://облако.моиотчеты.рф/api/manage/v1/Subscriptions?skip=0&take=10`.

   Пример запроса.

   ```
   curl -X GET "https://облако.моиотчеты.рф/api/manage/v1/Subscriptions?skip=0&take=10" -H "accept: text/plain"
   ```

   Пример ответа.

```
{
  "subscriptions": [
    {
      "id": "604f52e1261a3c19104c0e25",
      "name": "iwantpizza",
      "locale": null,
      "current": {
        "startTime": "2021-03-15T12:28:17.773Z",
        "endTime": "2121-03-15T12:28:17.773Z",
        "plan": {
          "id": "5f89b4d722d2d823440b6d10",
          "isActive": false,
          "displayName": "unlimited",
          "timePeriodType": "Year",
          "timePeriod": 100,
          "readonlyTimeLimitType": "Second",
          "readonlyTimeLimit": 0,
          "templatesSpaceLimit": 1048576000,
          "reportsSpaceLimit": 1048576000,
          "exportsSpaceLimit": 1048576000,
          "fileUploadSizeLimit": 1073741824,
          "dataSourceLimit": 10,
          "maxUsersCount": 10,
          "groupLimit": 5,
          "onlineDesigner": true,
          "isDemo": false,
          "urlToBuy": "https://fastreport.ru",
          "unlimitedPage": true,
          "pageLimit": 0
        }
      },
      "old": null,
      "invites": null,
      "templatesFolder": {
        "folderId": "604f52e1261a3c19104c0e22",
        "bytesUsed": 241247
      },
      "reportsFolder": {
        "folderId": "604f52e1261a3c19104c0e23",
        "bytesUsed": 16384
      },
      "exportsFolder": {
        "folderId": "604f52e1261a3c19104c0e24",
        "bytesUsed": 8059419
      }
    }
  ],
  "count": 1,
  "skip": 0,
  "take": 10
}
```


Идентификатор рабочего пространства (подписки) из примера выше `604f52e1261a3c19104c0e25`.

2. Для создания приглашения выполните `POST` запрос на `https://облако.моиотчеты.рф/api/manage/v1/Subscriptions/{subscriptionId}/invite`, где вместо `{subscriptionId}` нужно будет указать идентификатор рабочего пространства.

   Указав в теле запроса свойства приглашения.

   Пример тела.

```
{
  "usages": 1,
  "durable": true,
  "expiredDate": "2021-03-25T11:53:29.351Z"
}
```

- `usages` означает возможное количество использований,
- `durable` установленный в true указывает, что ***количество*** использований не ограничено,
- `expiredDate` указывает на срок окончания действия приглашение (свойство `durable` не сделает приглашение вечно действующим).

  Если оставить тело пустым, создастся одноразовое приглашение, которое перестаёт работать на следующий день.

Пример запроса.

```
curl -X POST "https://облако.моиотчеты.рф/api/manage/v1/Subscriptions/604f52e1261a3c19104c0e25/invite" -H "accept: text/plain" -H "Content-Type: application/json-patch+json" -d "{ \"usages\": 1, \"durable\": true, \"expiredDate\": \"2021-03-25T11:53:29.351Z\"}"
```

Пример ответа.

```
{
  "id": "604f52e1261a3c19104c0e25",
  "name": "iwantpizza",
  "locale": null,
  "current": {
    "startTime": "2021-03-15T12:28:17.773Z",
    "endTime": "2121-03-15T12:28:17.773Z",
    "plan": {
      "id": "5f89b4d722d2d823440b6d10",
      "isActive": false,
      "displayName": "unlimited",
      "timePeriodType": "Year",
      "timePeriod": 100,
      "readonlyTimeLimitType": "Second",
      "readonlyTimeLimit": 0,
      "templatesSpaceLimit": 1048576000,
      "reportsSpaceLimit": 1048576000,
      "exportsSpaceLimit": 1048576000,
      "fileUploadSizeLimit": 1073741824,
      "dataSourceLimit": 10,
      "maxUsersCount": 10,
      "groupLimit": 5,
      "onlineDesigner": true,
      "isDemo": false,
      "urlToBuy": "https://быстрыеотчеты.рф",
      "unlimitedPage": true,
      "pageLimit": 0
    }
  },
  "old": null,
  "invites": [
    {
      "usages": 1,
      "durable": true,
      "accessToken": "fj3534g341ir7dyytfiaap9z1r",
      "expiredDate": "2021-03-25T11:53:29.351Z",
      "addedUsers": [],
      "creatorUserId": "2df79f83-07f1-41ba-96b5-7757bbf377df"
    }
  ],
  "templatesFolder": {
    "folderId": "604f52e1261a3c19104c0e22",
    "bytesUsed": 241247
  },
  "reportsFolder": {
    "folderId": "604f52e1261a3c19104c0e23",
    "bytesUsed": 16384
  },
  "exportsFolder": {
    "folderId": "604f52e1261a3c19104c0e24",
    "bytesUsed": 8059419
  }
}
```


В подписке появилось приглашение с токеном доступа `fj3534g341ir7dyytfiaap9z1r`.

3. Нужно передать ссылку на принятие приглашения (или сам токен доступа) пользователю, которого вы хотите пригласить (любым удобным для вас способом). Итоговая ссылка будет выглядеть так - `https://облако.моиотчеты.рф/api/manage/v1/Subscriptions/{subscriptionId}/invite/{accessToken}/accept`, где вместо `{subscriptionId}` нужно будет указать идентификатор рабочего пространства, а вместо `{accessToken}` - токен доступа.

4. Теперь приглашаемый пользователь может перейти по этой ссылке прямо в браузере или же отправить `GET` запрос на `https://облако.моиотчеты.рф/api/manage/v1/Subscriptions/{subscriptionId}/invite/{accessToken}/accept`, где вместо `{subscriptionId}` нужно будет указать идентификатор рабочего пространства, а вместо `{accessToken}` - токен доступа.

   Пример запроса.

   ```
   curl -X GET
   "https://облако.моиотчеты.рф/api/manage/v1/Subscriptions/6051f2a06c07a10001737394/invite/to9kxrxdz4iwbfyz3pq
   4fktdcr/accept" -H "accept: text/plain"
   ```

Удаление приглашения

- Когда у приглашения заканчиваются использования, или срок его действия подходит к концу, оно не удаляется автоматически. Нужно будет сделать это вручную, отправив `DELETE` запрос на `https://облако.моиотчеты.рф/api/manage/v1/Subscriptions/{subscriptionId}/invite/{accessToken}` Пример запроса.

  ```
  curl -X DELETE
  "https://облако.моиотчеты.рф/api/manage/v1/Subscriptions/6051f2a06c07a10001737394/invite/to9kxrxdz4iwbfyz3pq
  4fktdcr" -H "accept: text/plain"
  ```

  В ответ будет получено пустое сообщение с кодом `NO CONTENT 204`.

Добавляем пользователя напрямую

1. Для добавления нового пользователя в рабочее пространство необходим идентификатор рабочего пространства.

   Получите идентификатор подписки сделав `GET` запрос на `https://облако.моиотчеты.рф/api/manage/v1/Subscriptions?skip=0&take=10`.

   Пример запроса.

   ```
   curl -X GET "https://облако.моиотчеты.рф/api/manage/v1/Subscriptions?skip=0&take=10" -H "accept: text/plain"
   ```

   Пример ответа.

```
{
  "subscriptions": [
    {
      "id": "5fa919fa292a8300019349bc",
      "name": "Awesome Corp",
      "current": {
        "startTime": "2020-11-17T10:22:58.584Z",
        "endTime": "2025-11-17T10:22:58.584Z",
        "plan": {
          "id": "5f43924b0231500001225686",
          "isActive": false,
          "displayName": "The greatest power",
          "timePeriodType": "Year",
          "timePeriod": 5,
          "readonlyTimeLimitType": "Second",
          "readonlyTimeLimit": 0,
          "templatesSpaceLimit": 1048576000,
          "reportsSpaceLimit": 1048576000,
          "exportsSpaceLimit": 1048576000,
          "fileUploadSizeLimit": 1048576000000,
          "dataSourceLimit": 10,
          "maxUsersCount": 10,
          "groupLimit": 5,
          "onlineDesigner": true,
          "isDemo": false,
          "urlToBuy": "https://fastreport.ru",
          "unlimitedPage": true,
          "pageLimit": 15
        }
      },
      "old": [],
      "templatesFolder": {
        "folderId": "5fa919f9292a8300019349b9",
        "bytesUsed": 1668491
      },
      "reportsFolder": {
        "folderId": "5fa919f9292a8300019349ba",
        "bytesUsed": 6085990
      },
      "exportsFolder": {
        "folderId": "5fa919fa292a8300019349bb",
        "bytesUsed": 8336710
      }
    }
  ],
  "count": 1,
  "skip": 0,
  "take": 10
}
```


Идентификатор рабочего пространства (подписки) из примера выше `5fa919fa292a8300019349bc`.

2. Для добавления нового пользователя сделайте `PUT` запрос `https://облако.моиотчеты.рф/api/manage/v1/Subscriptions/{subscriptionId}/users/{userId}`, где вместо `{subscriptionId}` следует передать идентификатор рабочего пространства, а вместо `{userId}` следует передать идентификатор пользователя.

   Пример запроса.

   ```
   curl -X PUT "https://облако.моиотчеты.рф/api/manage/v1/Subscriptions/5fa919fa292a8300019349bc/users/5af5a8dc-8cb0-40f9-ac99-ca2533fa4492" -H "accept: text/plain"
   ```

   В ответ вы получите пустое сообщение со статус кодом `OK 200`.

3. Для получения списка пользователей сделайте `GET` запрос на `https://облако.моиотчеты.рф/api/manage/v1/Subscriptions/{id}/users?skip=0&take=10`, где место `{id}` следует указать идентификатор рабочего пространства.

   Пример запроса.

   ```
   curl -X GET "https://облако.моиотчеты.рф/api/manage/v1/Subscriptions/5fa919fa292a8300019349bc/users?skip=0&take=10" -H "accept: text/plain"
   ```

   Пример ответа.

   ```
   {
    "users": [
      {
        "userId": "5af5a8dc-8cb0-40f9-ac99-ca2533fa4491"
      },
      {
        "userId": "5af5a8dc-8cb0-40f9-ac99-ca2533fa4492"
      }
    ],
    "count": 2,
    "take": 10,
    "skip": 0
   }
   ```

4. Для удаления пользователя из рабочего пространства сделайте `DELETE` запрос на `https://облако.моиотчеты.рф/api/manage/v1/Subscriptions/{subscriptionId}/users/{userId}`, где вместо `{subscriptionId}` следует передать идентификатор рабочего пространства, а вместо `{userId}` следует передать идентификатор пользователя.

   Пример запроса.

   ```
   curl -X DELETE "https://облако.моиотчеты.рф/api/manage/v1/Subscriptions/5fa919fa292a8300019349bc/users/5af5a8dc-8cb0-40f9-ac99-ca2533fa4492" -H "accept: text/plain"
   ```

   В ответ будет получено пустое сообщение с кодом `OK 200`.

Что дальше?

- Работа с группами
- Помощь и обратная связь

Работа с группами

В этой статье рассмотрен процесс создания новой группы, добавление пользователя в группу и получение списка пользователей группы.

Приступая к работе

Вам понадобятся следующие инструменты и возможности:

1. Знания по использованию API key в МоиОтчеты Облако.

   В этой статье будет пропущена дополнительная информация по аутентификации и авторизации.

2. Инструмент curl.

   Подойдёт любой другой REST клиент, но примеры будут построены для curl.

3. Активная подписка для МоиОтчеты Облако, в которой есть два слота для пользователя.

4. Доступ в интернет.

Замечание

Обратите внимание! Добавить пользователя в группу возможно только если пользователь существует в рабочем пространстве.

Обратите внимание! Добавить пользователя в группу возможно только по его идентификатору.

Инструкция

1. Для создания новой группы необходим идентификатор рабочего пространства и название новой группы.

   Получите идентификатор рабочего пространства сделав `GET` запрос на `https://облако.моиотчеты.рф/api/manage/v1/Subscriptions?skip=0&take=10`.

   Пример запроса.

   ```
   curl -X GET "https://облако.моиотчеты.рф/api/manage/v1/Subscriptions?skip=0&take=10" -H "accept: text/plain"
   ```

   Пример ответа.

```
{
  "subscriptions": [
    {
      "id": "5fa919fa292a8300019349bc",
      "name": "Awesome Corp",
      "current": {
        "startTime": "2020-11-17T10:22:58.584Z",
        "endTime": "2025-11-17T10:22:58.584Z",
        "plan": {
          "id": "5f43924b0231500001225686",
          "isActive": false,
          "displayName": "The greatest power",
          "timePeriodType": "Year",
          "timePeriod": 5,
          "readonlyTimeLimitType": "Second",
          "readonlyTimeLimit": 0,
          "templatesSpaceLimit": 1048576000,
          "reportsSpaceLimit": 1048576000,
          "exportsSpaceLimit": 1048576000,
          "fileUploadSizeLimit": 1048576000000,
          "dataSourceLimit": 10,
          "maxUsersCount": 10,
          "groupLimit": 5,
          "onlineDesigner": true,
          "isDemo": false,
          "urlToBuy": "https://быстрыеотчеты.рф",
          "unlimitedPage": true,
          "pageLimit": 15
        }
      },
      "old": [],
      "templatesFolder": {
        "folderId": "5fa919f9292a8300019349b9",
        "bytesUsed": 1668491
      },
      "reportsFolder": {
        "folderId": "5fa919f9292a8300019349ba",
        "bytesUsed": 6085990
      },
      "exportsFolder": {
        "folderId": "5fa919fa292a8300019349bb",
        "bytesUsed": 8336710
      }
    }
  ],
  "count": 1,
  "skip": 0,
  "take": 10
}
```


Идентификатор рабочего пространства (подписки) из примера выше `5fa919fa292a8300019349bc`.

2. Для создания новой группы сделайте `POST` запрос `https://облако.моиотчеты.рф/api/manage/v1/Groups`, в тело запроса передайте JSON по схеме ниже.

```
{
  "name": "string",
  "subscriptionId": "string id"
}
```

Пример запроса.

```
curl -X POST "https://облако.моиотчеты.рф/api/manage/v1/Groups" -H "accept: text/plain" -H "Content-Type: application/json-patch+json" -d "{ \"name\": \"Моя первая группа\", \"subscriptionId\": \"5fa919fa292a8300019349bc\"}"
```

Пример ответа.

```
{
  "id": "5fe5d7866882ca0001760fcb",
  "name": "Моя первая группа",
  "subscriptionId": "5fa919fa292a8300019349bc"
}
```

Идентификатор группы из примера выше `5fe5d7866882ca0001760fcb`.

3. Для добавления нового пользователя в группу сделайте `PUT` запрос на `https://облако.моиотчеты.рф/api/manage/v1/Groups/{groupId}/Users/{userId}`, вместо `{groupId}` следует ввести идентификатор группы, а вместо `{userId}` следует ввести идентификатор пользователя.

   Пример запроса.

   ```
   curl -X PUT "https://облако.моиотчеты.рф/api/manage/v1/Groups/5fe5d7866882ca0001760fcb/Users/5af5a8dc-8cb0-40f9-ac99-ca2533fa4492" -H "accept: text/plain"
   ```

   В ответе будет получено пустое сообщение с кодом `OK 200`.

4. Для получения списка пользователей в группе сделайте `GET` запрос на `https://облако.моиотчеты.рф/api/manage/v1/Groups/{id}/Users?skip=0&take=10`, где вместо `{id}` следует указать идентификатор группы.

   Пример запроса.

   ```
   curl -X GET "https://облако.моиотчеты.рф/api/manage/v1/Groups/5fe5d7866882ca0001760fcb/Users?skip=0&take=10" -H "accept: text/plain"
   ```

   Пример ответа.

   ```
   {
     "users": [
       {
         "userId": "5af5a8dc-8cb0-40f9-ac99-ca2533fa4491",
         "userId": "5af5a8dc-8cb0-40f9-ac99-ca2533fa4492"
       }
     ],
     "count": 2,
     "take": 10,
     "skip": 0
   }
   ```

Что дальше?

- Помощь и обратная связь

Работа с задачами

Задачи в МоиОтчеты Облако - это действия по преобразованию и доставке документов потребителям. Подробно описаны в разделе Задачи.

Приступая к работе

Вам понадобятся следующие инструменты и возможности:

1. Знания по использованию API key в МоиОтчеты Облако.

   В этой статье будет пропущена дополнительная информация по аутентификации и авторизации.

2. Инструмент curl.

   Подойдёт любой другой REST клиент, но примеры будут построены для curl.

3. Активная подписка для МоиОтчеты Облако.

4. Доступ в интернет.

Общие сведения

В этой статье рассмотрим как работать с задачами.

Приступая к работе

Вам понадобятся следующие инструменты и возможности:

1. Знания по использованию API key в МоиОтчеты Облако.

   В этой статье будет пропущена дополнительная информация по аутентификации и авторизации.

2. Инструмент curl.

   Подойдёт любой другой REST клиент, но примеры будут построены для curl.

3. Активная подписка для МоиОтчеты Облако.

4. Доступ в интернет.

О ViewModel и типах

При работе вы будете передавать разные ViewModel в виде JSON. Важно, что первое поле любого ViewModel должно быть "$t": "Название". Подробные примеры будут приведены ниже в этой статье.

VM'ы для создания задач:

- CreatePrepareTaskVM
- CreateExportTemplateTaskVM
- CreateExportReportTaskVM
- CreateFetchTaskVM
- CreateEmailTaskVM
- CreateWebhookTaskVM
- CreateFTPUploadTaskVM

VM'ы для запуска задач напрямую:

- RunPrepareTaskVM
- RunExportTemplateTaskVM
- RunExportReportTaskVM
- RunFetchTaskVM
- RunEmailTaskVM
- RunWebhookTaskVM
- RunFTPUploadTaskVM

VM'ы для обновления свойств задач:

- UpdatePrepareTaskVM
- UpdateExportTemplateTaskVM
- UpdateExportReportTaskVM
- UpdateFetchTaskVM
- UpdateEmailTaskVM
- UpdateWebhookTaskVM
- UpdateFTPUploadTaskVM

VM для обновления прав задач:

- UpdateTaskPermissionsVM

Создание задачи

Для создания задач следует использовать метод CreateTask. Он принимает VM для создания задач.

Рассмотрим создание задачи экспорта отчёта и последующей отправкой на Webhook:

```
curl -X 'POST' \
 'https://облако.моиотчеты.рф/api/tasks/v1/tasks' \
 -H 'accept: application/json' \
 -H 'Content-Type: application/json' \
 -d '{
 "$t": "CreateExportTemplateTaskVM",
 "subscriptionId": "23e0134c816935c1e11b3737",
 "name": "SendViaWebhookExport",
 "inputFile": {
   "entityId": "61e0134c816935c1e11b3787"
 },
 "transports": [
   {
     "$t": "CreateWebhookTaskVM",
     "Endpoints": [
{
    "BearerToken": "allotoken",
          "Url": "https://localhost:7104/api",
          "Headers": {"header1":"val1", "header2":"val2"}
}
     ]
   }
 ],
 "format": "Pdf"
}'
```

В поля EntityId и FolderId следует записывать реальные идентификаторы объектов. Иначе задача будет прервана с ошибкой.

> Обратите внимание! Если не указать OutputFile, то он будет сохранён во временную папку. Если указать пустой OutputFile - в рутовую папку. Если указать id папки - в неё.

Список всех доступных вьюмоделей доступен по ссылке - https://облако.моиотчеты.рф/api/swagger/index.html

Получение списка задач

```
// Get first 10 tasks from the subscription
curl -X 'GET' \
 'https://облако.моиотчеты.рф/api/tasks/v1/tasks?skip=0&take=10' \
 -H 'accept: application/json'
```

Выполнение задачи по указанному идентификатору

```
curl -X 'POST' \
 'https://облако.моиотчеты.рф/api/tasks/v1/tasks/42d134ae3130aaad37by345f/run' \
 -H 'accept: */*' \
 -d ''
```

Удаление задачи из хранилища

```
curl -X 'DELETE' \
  'https://облако.моиотчеты.рф/api/tasks/v1/tasks/42d134ae3130aaad37by345f' \
  -H 'accept: */*'
```

> Обратите внимание! В примерах нет заголовка Authorization потому что используется модель аутентификации, основанная на cookie. Подробнее про авторизацию читайте в разделе Аутентификация и авторизация

Сохранение на FTP-сервер

В этой статье рассмотрим способ отправки отчёта на FTP сервер. Инструкции о том как работать с задачами описаны в разделе Общие сведения.

Приступая к работе

Вам понадобятся следующие инструменты и возможности:

1. Знания по использованию API key в МоиОтчеты Облако.

   В этой статье будет пропущена дополнительная информация по аутентификации и авторизации.

2. Инструмент curl.

   Подойдёт любой другой REST клиент, но примеры будут построены для curl.

3. Активная подписка для МоиОтчеты Облако.

4. Доступ в интернет.

5. Настроенный и доступный FTP сервер.

> Обратите внимание! Пункты выше описывают рекомендуемые инструменты.

> Обратите внимание! Данное руководство предполагает, что у вас есть настроенный FTP-сервер для принятия файлов от внешних источников.

Создание задачи

Рассмотрим создание задачи отправки шаблона на FTP-сервер:

```
// Создание задачи
 curl -X 'POST' \
  'https://облако.моиотчеты.рф/api/tasks/v1/Tasks' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
  "$t": "CreateFTPUploadTaskVM",
  "name": "Задача отправки по FTP",
  "subscriptionId": "23e0134c816935c1e11b3737",
  "ftpHost": "ftp://localhost",
  "ftpPort": 21,
  "ftpUsername": "FtpUser",
  "ftpPassword": "password",
  "archive": false,
  "archiveName": "Имя архива",
  "useSFTP": false,
  "inputFile": {
    "entityId": "61e0134c816935c1e11b3787",
    "type": "Template"
  },
  "destinationFolder": "/путь_назначения/"
}'
```

```
// Запуск задачи по идентификатору
curl -X 'POST' \
  'https://облако.моиотчеты.рф/api/tasks/v1/Tasks/{идентификатор задачи}/run' \
  -H 'accept: */*' \
  -d ''
```

В поля EntityId и SubscriptionId следует записывать реальные идентификаторы объектов. Иначе задача будет прервана с ошибкой.

Выполнение задачи из тела запроса

```
curl -X 'POST' \
  'https://облако.моиотчеты.рф/api/tasks/v1/Tasks/run' \
  -H 'accept: */*' \
  -H 'Content-Type: application/json' \
  -d '{
  "$t": "RunFTPUploadTaskVM",
  "name": "Задача отправки по FTP",
  "subscriptionId": "23e0134c816935c1e11b3737",
  "ftpHost": "ftp://localhost",
  "ftpPort": 21,
  "ftpUsername": "FtpUser",
  "ftpPassword": "password",
  "archive": false,
  "archiveName": "Имя архива",
  "useSFTP": false,
  "inputFile": {
    "entityId": "61e0134c816935c1e11b3787",
    "type": "Template"
  },
  "destinationFolder": "/путь_назначения/"
}'
```

> Обратите внимание! В этом случае отправка на FTP будет выполнена непосредственно этим запросом и задача не будет сохранена в базе данных.

> Обратите внимание! В примерах нет заголовка Authorization потому что используется модель аутентификации, основанная на cookie. Подробнее про авторизацию читайте в разделе Аутентификация и авторизация

Сохранение по Webhook

В этой статье рассмотрим способ отправки отчёта по вебхуку. Инструкции о том как работать с задачами описаны в разделе Общие сведения.

Приступая к работе

Вам понадобятся следующие инструменты и возможности:

1. Знания по использованию API key в МоиОтчеты Облако.

   В этой статье будет пропущена дополнительная информация по аутентификации и авторизации.

2. Инструмент curl.

   Подойдёт любой другой REST клиент, но примеры будут построены для curl.

3. Активная подписка для МоиОтчеты Облако.

4. Доступ в интернет.

5. Клиентское приложение, принимающее запросы.

> Обратите внимание! Пункты выше описывают рекомендуемые инструменты.

> Обратите внимание! Данное руководство предполагает, что у вас есть клиентское приложение, принимающее и обрабатывающее входящие вебхуки.

Создание задачи

Рассмотрим создание задачи отправки шаблона по вебхуку:

```
// Создание задачи
curl -X 'POST' \
  'https://облако.моиотчеты.рф/api/tasks/v1/Tasks' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
  "$t": "CreateWebhookTaskVM",
  "name": "Задача отправки по Webhook",
  "subscriptionId": "{идентификатор рабочего пространства}",
  "url": "https://example.com/",
  "headers": {
      "Authorization": "Bearer <token>",
      "Content-Type": "multipart/form-data"
  },
  "inputFile": {
    "entityId": "{идентификатор шаблона}",
    "type": "Template"
  }
}'
```

```
// Запуск задачи по идентификатору
curl -X 'POST' \
  'https://облако.моиотчеты.рф/api/tasks/v1/Tasks/{идентификатор задачи}/run' \
  -H 'accept: */*' \
  -d ''
```

> Обратите внимание! В поля EntityId и SubscriptionId следует записывать реальные идентификаторы

> объектов. Иначе задача будет прервана с ошибкой.

> Обратите внимание! В EntityId может передаваться идентификатор шаблона, отчёта и экспорта.

Выполнение задачи из тела запроса

```
curl -X 'POST' \
  'https://облако.моиотчеты.рф/api/tasks/v1/Tasks/run' \
  -H 'accept: */*' \
  -H 'Content-Type: application/json' \
  -d '{
  "$t": "RunWebhookTaskVM",
  "name": "Задача отправки по Webhook",
  "subscriptionId": "{идентификатор рабочего пространства}",
  "url": "https://example.com/",
  "headers": {
    "Authorization": "Bearer <token>",
    "Content-Type": "multipart/form-data",
    "Content-Length": "1278"
  },
  "inputFile": {
    "entityId": "{идентификатор шаблона}",
    "type": "Template"
  }
}'
```

> Обратите внимание! В этом случае отправка по вебхуку будет выполнена непосредственно этим запросом и задача не будет сохранена в базе данных.

Пример запроса, который придёт на эндпоинт вебхука:

```
{
 "startedDateTime": "2024-06-07 08:37:09",
 "request": {
  "method": "POST",
  "url": "https://example.com/6e259560-25e5-482b-a7b2-8c5267ba6ae3/",
  "headers": [
   {
    "name": "connection",
    "value": "close"
   },
   {
    "name": "content-length",
    "value": "1421"
   },
   {
    "name": "content-type",
    "value": "multipart/form-data; boundary=\"62ec6524-9360-4e91-834f-e9531e2d2c30\""
   },
   {
    "name": "host",
    "value": "example.com"
   }
  ],
  "bodySize": 0,
  "postData": {
   "mimeType": "application/json",
   "text": ""
  }
 },
 "response": {
  "status": 200,
  "httpVersion": "HTTP/1.1",
  "headers": [
   {
    "name": "Content-Type",
    "value": "text/html"
   }
  ],
  "content": {
   "size": 145,
   "text": "This URL has no default content configured.",
   "mimeType": "text/html"
  }
 }
}
```


Обратите внимание! В примерах нет заголовка Authorization потому что используется модель аутентификации, основанная на cookie. Подробнее про авторизацию читайте в разделе Аутентификация и авторизация.

C# .NET

В этом разделе приведены пошаговые инструкции и руководства для выполнения типовых задач по применению МоиОтчеты Облако с использованием языка программирования C#/.NET и FastReport.Cloud.SDK.

Приступая к работе

Вам понадобится следующие инструменты и возможности.

1. .NET SDK.
2. Редактор C# кода или текстовый редактор, например Visual Studio Code.
3. Дизайнер отчётов FastReport Community Designer.

   Для некоторых руководств нужен будет дизайнер для создания отчёта.
4. Активная подписка МоиОтчеты Облако.
5. Доступ в интернет.

> Обратите внимание! Эти руководства рассчитаны, что вы уже знаете, как разработать своё приложение на языке программирования C#.

> Обратите внимание! Пункты выше описывают рекомендуемые инструменты.

Установка и настройка

Для подключения SDK следует установить пакет `FastReport.Cloud.SDK` последней стабильной версии. Это можно сделать следующей командой.

```
dotnet add package FastReport.Cloud.SDK
```

Для добавления поддержки ASP.NET следует установить пакет `FastReport.Cloud.SDK.Web`.

```
dotnet add package FastReport.Cloud.SDK.Web
```

Также в класс `Startup` необходимо добавить строчку.

```
services.AddFastReportCloud();
```

Что дальше?

Советуем начать своё ознакомление со статьей:

- Статус коды.
- Аутентификация и авторизация.

Аутентификация и авторизация

Процесс проверки подлинности пользовательских данных и выдача пользователю определённых прав в МоиОтчеты Облако осуществляется по одному из двух доступных способов:

1. Через JWT token.

   В этом случае аутентификацию нужно пройти лично, а токен будет действовать только 5 минут, за которые пользователь должен войти в своё приложение. При подключении к серверу браузер перенаправит на сервер аутентификации, после чего сгенерирует токен доступа. С точки зрения безопасности мы ограничиваем возможность получения JWT токена только лично пользователем.

   > Если в течении 5 минут пользователь не зашёл в приложение, то аутентификацию необходимо пройти заново. Если пользователь зашёл в приложение повторная аутентификация не требуется.

2. Через API key.

   В этом случае процесс получения прав доступа осуществляется для серверных приложений. Для получения ключа доступа (API key) необходимо присутствие пользователя. Однако сам ключ может действовать продолжительное время, например, год.

Получение первого API key

Для получения первого API key воспользуйтесь пользовательской панелью. Если по какой то причине доступа к пользовательской панели нет, можно запросить ключ по описанию ниже.

1. Откройте ссылку в браузере: https://облако.моиотчеты.рф/account/signin?r=https://облако.моиотчеты.рф/api/manage/v1/ApiKeys.

   Переход по этой ссылке направит вас на автоматический процесс аутентификации браузера.

2. Теперь когда аутентификация пройдена, необходимо запросить новый ключ.

   Нажмите `F12` или `Ctrl+Shift+I`, что бы открыть панель разработчика. Сочетание клавиш могут отличаться от стандартных, в этом случае откройте панель разработчика через меню браузера.

3. Скопируйте и выполните код в консоли JavaScript.

   Этот код сделает `POST` запрос на URL `https://облако.моиотчеты.рф/api/manage/v1/ApiKeys` для создания нового ключа доступа до 2030 года.

4. Обновите страницу браузера и получите результат.

   ```
   {
      "apiKeys": [
         {
            "value": "cc355oeu1z5d5wncayo33me6c1g5junqdqk4pkupid7t8ynjshey",
            "description": "Generated by js develop panel",
            "expired": "2030-01-01T07:41:23.399Z"
         }
      ],
      "count": 1
   }
   ```

Теперь вы можете использовать API key, в случае выше это `cc355oeu1z5d5wncayo33me6c1g5junqdqk4pkupid7t8ynjshey`.

Повторно получать новый API key через браузер нет необходимости.

Как использовать API key

Ключ следует передавать с каждым запросом в заголовке `Authorization: Basic`. В качестве имени пользователя следует использовать apikey в качестве пароля значение ключа. Например.

```
Authorization: Basic Base64Encode(apikey:cc355oeu1z5d5wncayo33me6c1g5junqdqk4pkupid7t8ynjshey);
```

Где `Base64Encode` это функция кодировки строки в base64.

Для `FastReport.Cloud.SDK` есть специальный класс, который позволяет добавлять ключ к заголовку запроса <xref:FastReport.Cloud.FastReportCloudApiKeyHeader>.

Для добавления нужного заголовка создайте новый `HttpClient`.

```
HttpClient httpClient = new HttpClient();
httpClient.DefaultRequestHeaders.Authorization = new FastReportCloudApiKeyHeader(apiKey);
```

Теперь этот `HttpClient` можно использовать для всех запросов.

Получение нового API key

Для получения нового ключа следует вызвать метод <xref:FastReport.Cloud.IApiKeysClient.CreateApiKeyAsync(FastReport.Cloud.CreateApiKeyVM,System.Threading.CancellationToken)>

```
CreateApiKeyVM model = new CreateApiKeyVM()
{
   Description = "Created by FastReport.Cloud.SDK",
   Expired = DateTime.UtcNow.AddYears(1)
};

IApiKeysClient apiKeysClient = new ApiKeysClient(httpClient);
await apiKeysClient.CreateApiKeyAsync(model);
```

> По возможности используйте асинхронные аналоги методов вместо синхронных.

В результате выполнения этой функции будет получена модель <xref:FastReport.Cloud.ApiKeyVM>.

Что дальше?

- Загрузка нового шаблона.
- Работа с группами.
- Добавление новых пользователей в рабочее пространство.

Загрузка нового шаблона

Одна из основных возможностей МоиОтчеты Облако — это хранение шаблонов, отчётов и других данных в облаке. В этой статье будет рассмотрен способ загрузить свой готовый шаблон в хранилище шаблонов МоиОтчеты Облако.

Приступая к работе

Вам понадобятся следующие инструменты и возможности:

1. Знания по использованию API key в МоиОтчеты Облако.

   В этой статье будет пропущена дополнительная информация по аутентификации и авторизации.

2. .NET SDK.

3. Редактор C# кода или текстовый редактор, например Visual Studio Code.

4. Шаблон отчёта.

   Его можно подготовить с помощью бесплатной программы FastReport Community Designer.

5. Активная подписка для МоиОтчеты Облако.

6. Доступ в интернет.

> Обратите внимание! Это руководство рассчитано, что вы уже знаете, как разработать своё приложение на языке программирования C#.

> Обратите внимание! Пункты выше описывают рекомендуемые инструменты.

Инструкция

1. Вам необходимо получить идентификатор корневой папки рабочего пространства. Этот идентификатор никогда не будет меняться, поэтому его можно сохранить и не делать запрос каждый раз перед загрузкой нового шаблона.

   Что бы запросить корневую директорию вызовите метод <xref:FastReport.Cloud.ITemplateFoldersClient.GetRootFolderAsync(System.String,System.Threading.CancellationToken)>.

   ```
   public async Task<string> GetTemplatesRoot(HttpClient httpClient, string subscriptionId = null)
   {
      ITemplateFoldersClient templateFoldersClient = new TemplateFoldersClient(httpClient);

      FileVM result = await templateFoldersClient.GetRootFolderAsync(subscriptionId);

      return result.Id;
   }
   ```

   В этом примере параметр `subscriptionId` указывает идентификатор рабочего пространства (подписки), если он не задан (равен null), то будет возвращен идентификатор рабочего пространства пользователя по умолчанию.

2. Для загрузки нужного шаблона воспользуетесь методом <xref:FastReport.Cloud.ITemplatesClient.UploadFileAsync(System.String,FastReport.Cloud.TemplateCreateVM,System.Threading.CancellationToken)>.

```
public async Task<string> UploadFrx(HttpClient httpClient, string folderId, string filePath)
{
    ITemplatesClient templatesClient = new TemplatesClient(httpClient);

    byte[] bytes = File.ReadAllBytes(filePath);

    TemplateCreateVM model = new TemplateCreateVM()
    {
        Name = Path.GetFileName(filePath),
        Content = Convert.ToBase64String(bytes)
    };

    TemplateVM result = await templatesClient.UploadFileAsync(folderId, model);

    return result.Id;
}
```

В этом примере функция получает файл с диска и загружает его в директорию `folderId`, подробнее о параметрах:

- `folderId` — идентификатор директории шаблона.
- `model.Name` — название файла, так он будет отображаться внутри МоиОтчеты Облако.

  В названии файла должно быть расширение `.frx`, если его нет, то добавьте его вручную.
- `model.Content` — содержимое файла закодированное в base64.

Метод возвращает идентификатор загруженного шаблона.

Теперь этот шаблон можно использовать для подготовки (построения) отчёта и экспорта.

Что дальше?

- Управление правами доступа на примере шаблона.
- Построение отчёта.

Управление правами доступа на примере шаблона

Ограничение доступа к приватным ресурсам очень важная часть МоиОтчеты Облако. Гибкая система доступа позволяет задать ограничение или выдать права на каждый ресурс отдельно, задавая круг лиц которые могут получить к ним доступ.

Приступая к работе

Вам понадобятся следующие инструменты и возможности:

1. Знания по использованию API key в МоиОтчеты Облако.

   В этой статье будет пропущена дополнительная информация по аутентификации и авторизации.

2. .NET SDK.

3. Редактор C# кода или текстовый редактор, например Visual Studio Code.

4. Шаблон отчёта.

   Как загрузить шаблон отчёта можно узнать в статье Загрузка нового шаблона.

5. Активная подписка для МоиОтчеты Облако.

6. Доступ в интернет.

> Обратите внимание! Это руководство рассчитано, что вы уже знаете, как разработать своё приложение на языке программирования C#.

> Обратите внимание! Пункты выше описывают рекомендуемые инструменты.

Инструкция

1. Для просмотра текущих прав на ресурс воспользуйтесь методом <xref:FastReport.Cloud.ITemplatesClient.GetPermissionsAsync(System.String,System.Threading.CancellationToken)>.

   ```
   public async Task<FilePermission> GetOwnerPermissions(HttpClient httpClient, string templateId)
   {
    ITemplatesClient templatesClient = new TemplatesClient(httpClient);

      FilePermissions permissions = await
         templatesClient.GetPermissionsAsync(templateId);

      return permissions.Owner;
   }
   ```

   В этом примере метод возвращает права доступа для владельца шаблона.

   > Обратите внимание! Классы <xref:FastReport.Cloud.FilePermissions> и <xref:FastReport.Cloud.FilePermission> отличаются буквой `s` на конце, первый содержит полное описание прав доступа к сущности, второй – только описание для одной из категории.

2. Для изменения разрешений воспользуйтесь методом <xref:FastReport.Cloud.ITemplatesClient.UpdatePermissionsAsync(System.String,FastReport.Cloud.Update

FilePermissionsVM,System.Threading.CancellationToken)>.

```
public async Task<FilePermission> ShareTemplate(HttpClient httpClient, string templateId)
{
   ITemplatesClient templatesClient = new TemplatesClient(httpClient);

   UpdateFilePermissionsVM viewModel = new UpdateFilePermissionsVM()
   {
      NewPermissions = new FilePermissions() {
 Anon = new FilePermission() { Get = FilePermissionGet.Entity | FilePermissionGet.Download }
      },
      Administrate = UpdateFilePermissionsVMAdministrate.Anon
   };

   var result = await templatesClient.AddPermissionAsync(templateId, viewModel);

   return result.Other;
}
```

В этом примере функция даёт возможность анонимным пользователям информацию о шаблоне и скачивать его.

Обратите внимание! В данном примере мы передаём только те права, которые хотим отредактировать и указываем их в свойстве `Administrate`.

Если нужно изменить сразу несколько категорий прав, можно перечислить их через побитовый оператор ИЛИ (`|`)

Что дальше?

- Построение отчёта.
- Работа с группами.
- Добавление новых пользователей в рабочее пространство.

Построение отчёта

В этой статье рассмотрен процесс построения отчёта из шаблона с помощью процессора отчётов МоиОтчеты Облако.

Приступая к работе

Вам понадобятся следующие инструменты и возможности:

1. Знания по использованию API key в МоиОтчеты Облако.

   В этой статье будет пропущена дополнительная информация по аутентификации и авторизации.

2. .NET SDK.

3. Редактор C# кода или текстовый редактор, например Visual Studio Code.

4. Шаблон отчёта.

   Как загрузить шаблон отчёта можно узнать в статье Загрузка нового шаблона.

5. Активная подписка для МоиОтчеты Облако.

6. Доступ в интернет.

> Обратите внимание! Это руководство рассчитано, что вы уже знаете, как разработать своё приложение на языке программирования C#.

> Обратите внимание! Пункты выше описывают рекомендуемые инструменты.

Инструкция

1. Вам необходим идентификатор шаблона для его построения, для его получения воспользуйтесь методом <xref:FastReport.Cloud.ITemplatesClient.GetFilesListAsync(System.String,System.Nullable{System.Int32},System.Nullable{System.Int32},System.String,System.Nullable{FastReport.Cloud.FileSorting},System.Nullable{System.Boolean},System.Nullable{System.Boolean},System.Threading.CancellationToken)>

   ```
   public async Task<string> GetTemplateId(HttpClient httpClient)
   {
      ITemplateFoldersClient templateFoldersClient =
         new TemplateFoldersClient(httpClient);
      ITemplatesClient templatesClient = new TemplatesClient(httpClient);

      FileVM rootFolder = await templateFoldersClient.GetRootFolderAsync(null);

      IEnumerable<TemplateVM> templates =
         await templatesClient.GetFilesListAsync(rootFolder.Id, 0, 10);

      TemplateVM template = templates.First();

      return template.Id;
   }
   ```

   В этом примере функция запрашивает корневую директорию рабочего пространства пользователя по умолчанию, затем запрашивает 10 шаблонов и возвращает первый.

2. Для построения отчёта вам понадобится директория, в которую можно положить отчёт. Запросите

корневую директорию отчётов, для этого воспользуйтесь методом <xref:FastReport.Cloud.IReportFoldersClient.GetRootFolderAsync(System.String,System.Threading.CancellationToken)>.

```
public async Task<string> GetReportsRoot(HttpClient httpClient,
                                         string subscriptionId = null)
{
   IReportFoldersClient reportFoldersClient = new ReportFoldersClient(httpClient);

  FileVM result = await reportFoldersClient.GetRootFolderAsync(subscriptionId);

   return result.Id;
}
```

В этом примере функция запрашивает корневую директорию, идентификатор рабочего пространства можно не задавать, в этом случае будет возвращена корневая директория для рабочего пространства пользователя по умолчанию.

3. Для построения отчёта воспользуйтесь методом <xref:FastReport.Cloud.ITemplatesClient.PrepareAsync(System.String,FastReport.Cloud.PrepareTemplateVM,System.Threading.CancellationToken)>.

```
public async Task<string> BuildReport(HttpClient httpClient,
                                      string folderId,
                                      string templateId,
                                      string fileName)
{
   ITemplatesClient templatesClient = new TemplatesClient(httpClient);

   PrepareTemplateVM task = new PrepareTemplateVM()
   {
      Name = Path.ChangeExtension(fileName, ".fpx"),
      FolderId = folderId
   };

   ReportVM result = await templatesClient.PrepareAsync(templateId, task);

   return result.Id;
}
```

В этом примере функция создаёт задачу на подготовку отчёта.

Обратите внимание! Хотя отчёт ещё не построен, но уже на этом этапе у него присвоен идентификатор, через некоторое время очередь построителя дойдёт до этой задачи и отчёт будет построен.

> Если не указать FolderId, то подготовленный отчёт будет сохранён в корневую папку.

4. Для получения информации об отчёте воспользуйтесь методом <xref:FastReport.Cloud.IReportsClient.GetFileAsync(System.String,System.Threading.CancellationToken)>.

```
public async Task<ReportVMStatus> CheckStatus(HttpClient httpClient, string reportId)
{
   IReportsClient reportsClient = new ReportsClient(httpClient);

   ReportVM result = await reportsClient.GetFileAsync(reportId);

   return result.Status.GetValueOrDefault();
}
```

В этом примере функция запрашивает отчёт по его идентификатору и возвращает статус.

Необходимо дождаться статуса <xref:FastReport.Cloud.FileStatus.Success>, проверяйте статус каждые несколько секунд.

5. Проверяем статус в цикле и скачиваем файл.

```
int tries = 10;
FileStatus status;
do
{
    status = await CheckStatus(httpClient, reportId);
    tries--;
} while (status != FileStatus.Success && tries > 0);

var report = await DownloadReport(httpClient, reportId);
```

6. Для скачивания отчёта воспользуйтесь методом <xref:FastReport.Cloud.IDownloadClient.GetReportAsync(System.String,System.Threading.CancellationToken)>.

```
public async Task<byte[]> DownloadReport(HttpClient httpClient, string reportId)
{
   IDownloadClient downloadClient = new DownloadClient(httpClient);

 FileResponse file = await downloadClient.GetReportAsync(reportId);

   using(MemoryStream ms = new MemoryStream())
   {
      file.Stream.CopyTo(ms);

      return ms.ToArray();
   }
}
```

В этом примере функция запрашивает файл и копирует его в память.

Что дальше?

- Экспорт отчёта в PDF.

Параметры отчёта

В этой статье рассмотрен процесс передачи параметров в отчёт, которые представляют собой словарь параметров, предоставляемых при формировании отчёта. Больше информации на эту тему можно найти в руководстве FastReport .NET в разделе Параметры отчёта.

Приступая к работе

Вам понадобятся следующие инструменты и возможности:

1. Знания по использованию API key в МоиОтчеты Облако.

   В этой статье будет пропущена дополнительная информация по аутентификации и авторизации.

2. .NET SDK.

3. Редактор C# кода или текстовый редактор, например Visual Studio Code.

4. Шаблон с указанными параметрами отчёта.

   Его можно подготовить с помощью бесплатной программы FastReport Community Designer.

5. Активная подписка для МоиОтчеты Облако.

6. Доступ в интернет.

> Обратите внимание! Это руководство рассчитано, что вы уже знаете, как разработать своё приложение на языке программирования C#.

> Обратите внимание! Пункты выше описывают рекомендуемые инструменты.

Передача параметров в отчёт

Параметры могут быть переданы в следующих случаях:

- при подготовке отчёта;
- при экспорте из шаблона;
- при работе с задачами.

Рассмотрим передачу параметров на примере подготовки отчёта. Подробная инструкция о том, как подготовить отчёт, содержится в разделе Построение отчёта.

```
public async Task PassingReportParameters(HttpClient httpClient,
                        string folderId,
                        string templateId,
                        string fileName)
{
   ITemplatesClient templatesClient = new TemplatesClient(httpClient);
   PrepareTemplateVM task = new PrepareTemplateVM()
   {
    Name = Path.ChangeExtension(fileName, ".fpx"),
    FolderId = folderId,
    ReportParameters = new Dictionary<string, string>
    {
       { "Parameter1", "Значение1" },
       { "Parameter2", "Значение2" }
    }
 };

 // Добавить новый параметр в словарь
 task.ReportParameters.Add("Parameter3", "Значение3");

 ReportVM result = await templatesClient.PrepareAsync(templateId, task);}
```

Экспорт отчёта в PDF

В этой статье рассмотрен процесс экспорта отчёта с помощью процессора отчётов МоиОтчеты Облако.

Приступая к работе

Вам понадобятся следующие инструменты и возможности:

1. Знания по использованию API key в МоиОтчеты Облако.

   В этой статье будет пропущена дополнительная информация по аутентификации и авторизации.

2. .NET SDK.

3. Редактор C# кода или текстовый редактор, например Visual Studio Code.

4. Отчёт.

   Как построить отчёт можно узнать в статье Построение отчёта.

5. Активная подписка для МоиОтчеты Облако.

6. Доступ в интернет.

> Обратите внимание! Это руководство рассчитано, что вы уже знаете, как разработать своё приложение на языке программирования C#.

> Обратите внимание! Пункты выше описывают рекомендуемые инструменты.

Замечание

Обратите внимание, что экспорт отчёта можно сделать напрямую из шаблона, без промежуточного сохранения отчёта. Для этого выполните те же команды для шаблона отчёта, заменив `Report` в методах на `Template`, также используйте идентификатор шаблона, а не отчёта.

Инструкция

1. Вам будет необходим идентификатор отчёта для экспорта в PDF, для его получения воспользуйтесь методом <xref:FastReport.Cloud.ITemplatesClient.GetFilesListAsync(System.String,System.Nullable{System.Int32},System.Nullable{System.Int32},System.String,System.Nullable{FastReport.Cloud.FileSorting},System.Nullable{System.Boolean},System.Nullable{System.Boolean},System.Threading.CancellationToken)>

   ```
   public async Task<string> GetReportId(HttpClient httpClient)
   {
      IReportFoldersClient reportFoldersClient = new ReportFoldersClient(httpClient);
      IReportsClient reportsClient = new ReportsClient(httpClient);

      FileVM rootFolder = await reportFoldersClient.GetRootFolderAsync(null);

      IEnumerable<ReportVM> reports =
         await reportsClient.GetFilesListAsync(rootFolder.Id, 0, 10);

      ReportVM report = reports.First();

      return report.Id;
   }
   ```

В этом примере функция запрашивает корневую директорию рабочего пространства пользователя по умолчанию, затем запрашивает 10 отчётов и возвращает первый.

2. Для экспорта отчёта понадобится директория, в которую следует сохранить экспорт.

   Получите корневую директорию экспортов, для этого воспользуйтесь методом <xref:FastReport.Cloud.IExportFoldersClient.GetRootFolderAsync(System.String,System.Threading.CancellationToken)>.

```
public async Task<string> GetExportsRoot(HttpClient httpClient,
                                         string subscriptionId = null)
{
   IExportFoldersClient exportFoldersClient = new ExportFoldersClient(httpClient);

   FileVM result = await exportFoldersClient.GetRootFolderAsync(subscriptionId);

   return result.Id;
}
```

   В этом примере функция запрашивает корневую директорию, идентификатор рабочего пространства можно не задавать, в этом случае будет возвращена корневая директория для рабочего пространства пользователя по умолчанию.

3. Для экспорта отчёта воспользуйтесь методом <xref:FastReport.Cloud.IReportsClient.ExportAsync(System.String,FastReport.Cloud.ExportReportVM,System.Threading.CancellationToken)>.

```
public async Task<string> ExportReport(HttpClient httpClient,
                                       string folderId,
                                       string reportId,
                                       string fileName)
{
   IReportsClient reportsClient = new ReportsClient(httpClient);

   ExportReportVM task = new ExportReportVM()
   {
      FileName = Path.ChangeExtension(fileName, ".pdf"),
      FolderId = folderId,
      Format = ExportReportTaskVMFormat.Pdf,
      ExportParameters = new Dictionary<string, string>
      {
         { "additionalProp1", ""},
         { "additionalProp2", ""},
         { "additionalProp3", ""}
      }
   };

   ExportVM result = await reportsClient.ExportAsync(reportId, task);

   return result.Id;
}
```

   - `FileName` — название результирующего файла. Если не указать расширение, или указать его неправильно - сервер заменит его самостоятельно.
   - `FolderId` — идентификатор директории куда будет помещён экспорт. Если оставить пустым, то экспорт будет помещён в корневую папку для экспортов в рабочем пространстве.
   - `Format` — формат экспорта.
   - `ExportParameters` — параметры экспорта. Задаются аналогично параметрам экспорта из библиотеки FastReport .NET. Более подробное описание хранится в документации пользователя в разделе параметры экспорта.

В этом примере функция создаёт задачу на экспорт отчёта.

Обратите внимание! Хотя отчёт ещё не экспортирован, но уже на этом этапе у экспорта присвоен идентификатор, через некоторое время очередь построителя дойдёт до этой задачи и отчёт будет экспортирован.

> Если не указать FolderId, то экспорт будет сохранён в корневую папку.

4. Для получения информации о файле воспользуйтесь методом <xref:FastReport.Cloud.IExportsClient.GetFileAsync(System.String,System.Threading.CancellationToken)>.

```
public async Task<ExportVMStatus> CheckStatus(HttpClient httpClient, string exportId)
{
    IExportsClient exportsClient = new ExportsClient(httpClient);

    ExportVM result = await exportsClient.GetFileAsync(exportId);

    return result.Status.GetValueOrDefault();
}
```

В этом примере функция запрашивает экспорт по его идентификатору и возвращает статус. Необходимо дождаться статуса <xref:FastReport.Cloud.FileStatus.Success>, проверяйте статус каждые несколько секунд.

5. Проверяем статус в цикле и скачиваем экспорт.

```
int tries = 10;
FileStatus status;
do
{
    status = await CheckStatus(httpClient, reportId);
    tries--;
} while (status != FileStatus.Success && tries > 0);
var report = await DownloadExport(httpClient, reportId);
```

6. Для скачивания отчёта воспользуйтесь методом <xref:FastReport.Cloud.IDownloadClient.GetExportAsync(System.String,System.Nullable{System.Boolean} System.Threading.CancellationToken)>.

```
public async Task<byte[]> DownloadExport(HttpClient httpClient, string exportId)
{
    IDownloadClient downloadClient = new DownloadClient(httpClient);

    FileResponse file = await downloadClient.GetExportAsync(exportId);

    using (MemoryStream ms = new MemoryStream())
    {
        file.Stream.CopyTo(ms);

        return ms.ToArray();
    }
}
```

В этом примере функция запрашивает файл и копирует его в память.

Что дальше?

- Работа с группами.
- Добавление новых пользователей в рабочее пространство.

Добавление новых пользователей в рабочее пространство

В этой статье рассмотрен процесс добавления нового пользователя в подписку, получения списка пользователей в подписке и удаления пользователя из подписки.

Приступая к работе

Вам понадобятся следующие инструменты и возможности:

1. Знания по использованию API key в МоиОтчеты Облако.

   В этой статье будет пропущена дополнительная информация по аутентификации и авторизации.

2. .NET SDK.

3. Редактор C# кода или текстовый редактор, например Visual Studio Code.

4. Активная подписка для МоиОтчеты Облако, в которой есть два слота для пользователя.

5. Доступ в интернет.

> Обратите внимание! Это руководство рассчитано, что вы уже знаете, как разработать своё приложение на языке программирования C#.

> Обратите внимание! Пункты выше описывают рекомендуемые инструменты.

Замечание

Обратите внимание! Добавить пользователя в рабочее пространство возможно только по идентификатору пользователя.

Инструкция

1. Для добавления нового пользователя в рабочее пространство необходим идентификатор рабочего пространства(подписки).

   Получите идентификатор рабочего пространства используя метод <xref:FastReport.Cloud.ISubscriptionsClient.GetSubscriptionsAsync(System.Nullable{System.Int32},System
   Nullable{System.Int32},System.Threading.CancellationToken)>.

   ```
   public async Task<string> GetSubscriptionId(HttpClient httpClient)
   {
      ISubscriptionsClient subscriptionsClient = new SubscriptionsClient(httpClient);
    SubscriptionsVM subscriptions =
         await subscriptionsClient.GetSubscriptionsAsync(0, 10);
    SubscriptionVM subscription = subscriptions.Subscriptions.First();
      return subscription.Id;
   }
   ```

   В этом примере функция запрашивает первые 10 рабочих пространств (подписок) из список рабочих пространств пользователя, выбирает первое рабочее пространство и возвращает его идентификатор.

   > Рабочее пространство всегда соотносится с одной подпиской, поэтому у них совпадает идентификатор.

2. Для добавления нового пользователя воспользуйтесь методом <xref:FastReport.Cloud.ISubscriptionUsersClient.AddUserAsync(System.String,System.String,System.Threading.CancellationToken)>.

```
public async Task AddUser(HttpClient httpClient, string subscriptionId, string userId)
{
   ISubscriptionUsersClient subscriptionUsersClient =
       new SubscriptionUsersClient(httpClient);
   await subscriptionUsersClient.AddUserAsync(subscriptionId, userId);
}
```

В этом примере функция добавляет пользователя с идентификатором `userId` в рабочее пространство с идентификатором `subscriptionId`.

3. Для получения списка пользователей рабочего пространство воспользуйтесь методом <xref:FastReport.Cloud.ISubscriptionUsersClient.GetUsersAsync(System.String,System.Nullable{System.Int32},System.Nullable{System.Int32},System.Threading.CancellationToken)>.

```
public async Task<IEnumerable<string>> GetUsers(HttpClient httpClient, string subscriptionId)
{
   ISubscriptionUsersClient subscriptionUsersClient =
       new SubscriptionUsersClient(httpClient);
  SubscriptionUsersVM users =
       await subscriptionUsersClient.GetUsersAsync(subscriptionId, 0, 10);
  return users.Users.Select(m => m.UserId);
}
```

В этом примере функция запрашивает 10 первых пользователей из рабочего пространства с идентификатором `subscriptionId`.

4. Для удаления пользователя из рабочего пространства воспользуетесь методом <xref:FastReport.Cloud.ISubscriptionUsersClient.RemoveUserAsync(System.String,System.String,System.Threading.CancellationToken)>.

```
public async Task RemoveUser(HttpClient httpClient, string subscriptionId, string userId)
{
   ISubscriptionUsersClient subscriptionUsersClient =
       new SubscriptionUsersClient(httpClient);
  await subscriptionUsersClient.RemoveUserAsync(subscriptionId, userId);
}
```

В этом методе функция удаляет пользователя с идентификатором `userId` из рабочего пространства с идентификатором `subscriptionId`.

Что дальше?

- Работа с группами
- Помощь и обратная связь

Работа с группами

В этой статье рассмотрен процесс создания новой группы, добавление пользователя в группу и получение списка пользователей группы.

Приступая к работе

Вам понадобятся следующие инструменты и возможности:

1. Знания по использованию API key в МоиОтчеты Облако.

   В этой статье будет пропущена дополнительная информация по аутентификации и авторизации.

2. .NET SDK.

3. Редактор C# кода или текстовый редактор, например Visual Studio Code.

4. Активная подписка для МоиОтчеты Облако, в которой есть два слота для пользователя.

5. Доступ в интернет.

> Обратите внимание! Это руководство рассчитано, что вы уже знаете, как разработать своё приложение на языке программирования C#.

> Обратите внимание! Пункты выше описывают рекомендуемые инструменты.

Замечание

Обратите внимание! Добавить пользователя в группу возможно только если пользователь существует в рабочем пространстве.

Обратите внимание! Добавить пользователя в группу возможно только по его идентификатору.

Инструкция

1. Для создания новой группы необходим идентификатор рабочего пространства и название новой группы.

   Для получения идентификатора рабочего пространства воспользуйтесь методом <xref:FastReport.Cloud.ISubscriptionsClient.GetSubscriptionsAsync(System.Nullable{System.Int32},System.Nullable{System.Int32},System.Threading.CancellationToken)>.

   ```
   public async Task<string> GetSubscriptionId(HttpClient httpClient)
   {
      ISubscriptionsClient subscriptionsClient = new SubscriptionsClient(httpClient);

      SubscriptionsVM subscriptions =
       await subscriptionsClient.GetSubscriptionsAsync(0, 10);

      SubscriptionVM subscription = subscriptions.Subscriptions.First();

      return subscription.Id;
   }
   ```

   В этом примере функция запрашивает первые 10 рабочих пространств из списка рабочих пространств пользователя, выбирает первое рабочее пространство и возвращает его идентификатор.

> Идентификатор рабочего пространства и подписки совпадает, т.к. к каждому рабочему пространству соотносится одна подписка.

2. Для создания новой группы воспользуйтесь методом <xref:FastReport.Cloud.IGroupsClient.CreateGroupAsync(FastReport.Cloud.CreateGroupVM,System.Threading.CancellationToken)>.

```
public async Task<string> CreateGroup(HttpClient httpClient,
                                      string subscriptionId,
                                      string groupName)
{
   IGroupsClient groupsClient = new GroupsClient(httpClient);

   CreateGroupVM viewModel = new CreateGroupVM()
   {
      Name = groupName,
      SubscriptionId = subscriptionId
   };

   GroupVM group = await groupsClient.CreateGroupAsync(viewModel);

   return group.Id;
}
```

В этом примере функция создаёт новую группу с именем группы `groupName` для рабочего пространства с идентификатором `subscriptionId`, в результате функция вернёт идентификатор созданной группы.

3. Для добавления нового пользователя в группу воспользуйтесь методом <xref:FastReport.Cloud.IGroupUsersClient.AddUserToGroupAsync(System.String,System.String,System.Threading.CancellationToken)>.

```
public async Task AddUser(HttpClient httpClient, string groupId, string userId)
{
   IGroupUsersClient groupUsersClient = new GroupUsersClient(httpClient);

   await groupUsersClient.AddUserToGroupAsync(groupId, userId);
}
```

В этом примере функция добавляет пользователя с идентификатором `userId` в группу с идентификатором `groupId`.

4. Для получения списка пользователей в группе воспользуйтесь методом <xref:FastReport.Cloud.IGroupUsersClient.GetUsersInGroupAsync(System.String,System.Nullable{System.Int32},System.Nullable{System.Int32},System.Threading.CancellationToken)>.

```
public async Task<IEnumerable<string>> GetUsers(HttpClient httpClient, string groupId)
{
   IGroupUsersClient groupUsersClient = new GroupUsersClient(httpClient);

   GroupUsersVM users =
    await groupUsersClient.GetUsersInGroupAsync(groupId, 0, 10);

 return users.Users.Select(m => m.UserId);
}
```

В этом примере функция запрашивает первые 10 пользователей из группы с идентификатором `groupId`.

Что дальше?

- Помощь и обратная связь

Работа с задачами

Задачи в МоиОтчеты Облако - это действия по преобразованию и доставке документов потребителям. Подробно описаны в разделе Задачи.

Приступая к работе

Вам понадобятся следующие инструменты и возможности:

1. Знания по использованию API key в МоиОтчеты Облако.

   В этой статье будет пропущена дополнительная информация по аутентификации и авторизации.

2. .NET SDK.

3. Редактор C# кода или текстовый редактор, например Visual Studio Code.

4. Шаблон отчёта.

   Его можно подготовить с помощью бесплатной программы FastReport Community Designer.

5. Активная подписка для МоиОтчеты Облако.

6. Доступ в интернет.

> Обратите внимание! Это руководство рассчитано, что вы уже знаете, как разработать своё приложение на языке программирования C#.

> Обратите внимание! Пункты выше описывают рекомендуемые инструменты.

Общие сведения

В этой статье рассмотрим как работать с задачами.

Приступая к работе

Вам понадобятся следующие инструменты и возможности:

1. Знания по использованию API key в МоиОтчеты Облако.

   В этой статье будет пропущена дополнительная информация по аутентификации и авторизации.

2. .NET SDK.

3. Редактор C# кода или текстовый редактор, например Visual Studio Code.

4. Шаблон отчёта.

   Его можно подготовить с помощью бесплатной программы FastReport Community Designer.

5. Активная подписка для МоиОтчеты Облако.

6. Доступ в интернет.

> Обратите внимание! Это руководство рассчитано, что вы уже знаете, как разработать своё приложение на языке программирования C#.

> Обратите внимание! Пункты выше описывают рекомендуемые инструменты.

Во первых, нужно создать HttpClient, который мы далее будем использовать:

```
var httpClient = new HttpClient();
httpClient.BaseAddress = new Uri("https://облако.моиотчеты.рф");
httpClient.DefaultRequestHeaders.Authorization = new FastReportCloudApiKeyHeader(ApiKey);
```

Далее получим подписку, относительно которой будем работать с задачами:

```
var subscriptions = new SubscriptionsClient(httpClient);
var subscription = (await subscriptions.GetSubscriptionsAsync(0, 10)).Subscriptions.FirstOrDefault();
```

Теперь нам нужно создать клиент для работы с задачами:

```
TasksClient tasksClient = new TasksClient(httpClient);
```

Теперь можно приступать непосредственно к работе с задачами.

Создание задачи

Для создания задач следует использовать метод CreateTask или CreateTaskAsync. Он может принимать любой класс-наследник TaskBaseVM:

- CreatePrepareTaskVM
- CreateExportTemplateTaskVM
- CreateExportReportTaskVM
- CreateFetchTaskVM
- CreateEmailTaskVM
- CreateWebhookTaskVM

- CreateFTPUploadTaskVM.

Рассмотрим создание задачи подготовки отчёта, сохранения его в папку и последующего экспорта в PDF с сохранение в папку:

```
await tasksClient.CreateTaskAsync(new CreatePrepareTaskVM
        {
            Name = "Моя первая задача",
            Type = TaskType.Prepare,
            InputFile = new InputFileVM
            {
                EntityId = "{templateId}"
            },
            OutputFile = new OutputFileVM
            {
                FileName = "Мой первый отчёт сгенерированный задачей.fpx",
                FolderId = "{идентификатор папки с отчётами}"
            },
            Exports = new List<CreateExportReportTaskVM>
            {
                new CreateExportReportTaskVM
                {
                    Format = ExportFormat.Pdf,
                    OutputFile = new OutputFileVM
                    {
                        FileName = "pdfИзFpxИзFrx.pdf",
                        FolderId = "{идентификатор папки с экспортами}"
                    }
                }
            }
        });
```

В поля EntityId и FolderId следует записывать реальные идентификаторы объектов. Иначе задача будет прервана с ошибкой.

> Обратите внимание! Если не указать OutputFile, то он будет сохранён во временную папку. Если указать пустой OutputFile - в рутовую папку. Если указать id папки - в неё.

Получение списка задач

```
// Получить первые 100 задач из рабочего пространства
var tasks = await tasksClient.GetListAsync(0, 100, subscription.Id);
```

Выполнение задачи по указанному идентификатору

```
// Запуск задачи по идентификатору
await tasksClient.RunTaskByIdAsync(id);
```

Удаление задач из хранилища

```
// Удалить все задачи
foreach(var t in tasks.Tasks)
{
    await tasksClient.DeleteTaskAsync(t.Id);
}
```

Сохранение на FTP-сервер

В этой статье рассмотрим способ отправки отчёта на FTP сервер. Инструкции о том как работать с задачами описаны в разделе Общие сведения.

Приступая к работе

Вам понадобятся следующие инструменты и возможности:

1. Знания по использованию API key в МоиОтчеты Облако.

   В этой статье будет пропущена дополнительная информация по аутентификации и авторизации.

2. .NET SDK.

3. Редактор C# кода или текстовый редактор, например Visual Studio Code.

4. Шаблон отчёта.

   Его можно подготовить с помощью бесплатной программы FastReport Community Designer.

5. Активная подписка для МоиОтчеты Облако.

6. Доступ в интернет.

7. Настроенный и доступный FTP сервер.

> Обратите внимание! Это руководство рассчитано, что вы уже знаете, как разработать своё приложение на языке программирования C#.

> Обратите внимание! Пункты выше описывают рекомендуемые инструменты.

> Обратите внимание! Данное руководство предполагает, что у вас есть опыт настройки и конфигурации FTP сервера для принятия файлов от внешних источников.

Создание задачи

Рассмотрим создание задачи отправки шаблона на FTP-сервер:

```
// Инициализация объекта
 CreateFTPUploadTaskVM ftpUploadTaskVM = new CreateFTPUploadTaskVM
 {
    Name = "Задача отправки на FTP",
    InputFile = new InputFileVM
    {
       EntityId = "{идентификатор шаблона}",
       Type = FileKind.Template
    },
    FtpHost = "{адрес FTP-сервера}",
    FtpPort = 21,
    FtpUsername = "{имя пользователя FTP-сервера}",
    FtpPassword = "{пароль}",
    UseSFTP = false,
    DestinationFolder = "{/путь_к_папке/}",
    Archive = false,
    ArchiveName = "Имя архива",
    SubscriptionId = "{идентификатор рабочего пространства}"
 };

 // Создание задачи
 TaskBaseVM ftpUploadTask = await tasksClient.CreateTaskAsync(ftpUploadTaskVM);

 // Запуск задачи по идентификатору
 await tasksClient.RunTaskByIdAsync(ftpUploadTask.Id);
```

В поля EntityId и SubscriptionId следует записывать реальные идентификаторы объектов. Иначе задача будет прервана с ошибкой.

Выполнение задачи из тела запроса

```
// Запуск задачи из тела запроса
 await tasksClient.RunTaskAsync(new RunFTPUploadTaskVM
 {
    InputFile = new RunInputFileVM
    {
       EntityId = "{идентификатор шаблона}",
       Type = FileKind.Template
    },
    FtpHost = "{адрес FTP-сервера}",
    FtpPort = 21,
    FtpUsername = "{имя пользователя FTP-сервера}",
    FtpPassword = "{пароль}",
    UseSFTP = false,
    DestinationFolder = "{/путь_к_папке/}",
    Archive = false,
    ArchiveName = "Архивированный шаблон",
    SubscriptionId = "{идентификатор рабочего пространства}"
 });
```

> Обратите внимание! В этом случае отправка на FTP будет выполнена непосредственно этим запросом и задача не будет сохранена в базе данных.

Обновление задачи по идентификатору

```
await tasksClient.UpdateTaskAsync("{идентификатор старой задачи}", new UpdateFTPUploadTaskVM()
{
    InputFile = new RunInputFileVM
    {
        EntityId = "{обновлённый идентификатор шаблона}",
        Type = FileKind.Template
    },
    FtpHost = "{Обновлённый адрес FTP-сервера}",
    FtpPort = 21,
    FtpUsername = "{Обновлённое имя пользователя FTP-сервера}",
    FtpPassword = "{пароль}",
    UseSFTP = false,
    DestinationFolder = "/обновлённый_путь_к_папке",
    Archive = false,
    ArchiveName = "Обновлённое имя архива"
});
```

Сохранение по Webhook

В этой статье рассмотрим способ отправки отчёта по вебхуку. Инструкции о том как работать с задачами описаны в разделе Общие сведения.

Приступая к работе

Вам понадобятся следующие инструменты и возможности:

1. Знания по использованию API key в МоиОтчеты Облако.

   В этой статье будет пропущена дополнительная информация по аутентификации и авторизации.

2. .NET SDK.

3. Редактор C# кода или текстовый редактор, например Visual Studio Code.

4. Шаблон отчёта.

   Его можно подготовить с помощью бесплатной программы FastReport Community Designer.

5. Активная подписка для МоиОтчеты Облако.

6. Доступ в интернет.

7. Клиентское приложение, принимающее запросы.

> Обратите внимание! Это руководство рассчитано, что вы уже знаете, как разработать своё приложение на языке программирования C#.

> Обратите внимание! Пункты выше описывают рекомендуемые инструменты.

> Обратите внимание! Данное руководство предполагает, что у вас есть клиентское приложение, принимающее и обрабатывающее входящие вебхуки.

Создание задачи

Рассмотрим создание задачи отправки шаблона по вебхуку и ее последующий запуск:

```
// Инициализация объекта
CreateWebhookTaskVM webhookTaskVM = new CreateWebhookTaskVM
{
    Name = "Задача отправки по Webhook",
    InputFile = new InputFileVM
    {
        EntityId = "{идентификатор шаблона}",
        Type = FileKind.Template
    },
    Url = new Uri("https://example.com/"),
    Headers = new Dictionary<string, string> {
        { "Authorization", "Bearer <token>" },
        { "Content-Length", "1238" }
    },
    SubscriptionId = "{идентификатор рабочего пространства}"
};

 // Создание задачи
 TaskBaseVM webhookTask = await tasksClient.CreateTaskAsync(webhookTaskVM);

 // Запуск задачи по идентификатору
 await tasksClient.RunTaskByIdAsync(webhookTask.Id);
```

> Обратите внимание! В поля EntityId и SubscriptionId следует записывать реальные идентификаторы объектов. Иначе задача будет прервана с ошибкой.

> Обратите внимание! В EntityId может передаваться идентификатор шаблона, отчёта и экспорта.

Выполнение задачи из тела запроса

```
// Запуск задачи из тела запроса
 await tasksClient.RunTaskAsync(new RunWebhookTaskVM
 {
     InputFile = new RunInputFileVM
     {
         EntityId = "{идентификатор шаблона}",
         Type = FileKind.Template
     },
     Url = new Uri("https://example.com/"),
     Headers = new Dictionary<string, string> {
         { "Authorization", "Bearer <token>" },
         { "Content-Type", "multipart/form-data" }
     },
     SubscriptionId = "{идентификатор рабочего пространства}"
 });
```

> Обратите внимание! В этом случае отправка по вебхуку будет выполнена непосредственно этим запросом и задача не будет сохранена в базе данных.

Обновление задачи по идентификатору

```
await tasksClient.UpdateTaskAsync("{идентификатор старой задачи}", new UpdateWebhookTaskVM()
 {
    InputFile = new RunInputFileVM
    {
        EntityId = "{обновлённый идентификатор шаблона}",
        Type = FileKind.Template
    },
    Url = new Uri("{Обновлённый адрес}"),
    Headers = new Dictionary<string, string> {
        { "{Обновлённый заголовок}", "{Обновлённое значение}" },
    }
 });
```

Пример запроса, который придёт на эндпоинт вебхука:


```
{
  "startedDateTime": "2024-06-07 08:37:09",
  "request": {
    "method": "POST",
    "url": "https://example.com/6e259560-25e5-482b-a7b2-8c5267ba6ae3/",
    "headers": [
      {
        "name": "connection",
        "value": "close"
      },
      {
        "name": "content-length",
        "value": "1421"
      },
      {
        "name": "content-type",
        "value": "multipart/form-data; boundary=\"62ec6524-9360-4e91-834f-e9531e2d2c30\""
      },
      {
        "name": "host",
        "value": "example.com"
      }
    ],
    "bodySize": 0,
    "postData": {
      "mimeType": "application/json",
      "text": ""
    }
  },
  "response": {
    "status": 200,
    "httpVersion": "HTTP/1.1",
    "headers": [
      {
        "name": "Content-Type",
        "value": "text/html"
      }
    ],
    "content": {
      "size": 145,
      "text": "This URL has no default content configured.",
      "mimeType": "text/html"
    }
  }
}
```