

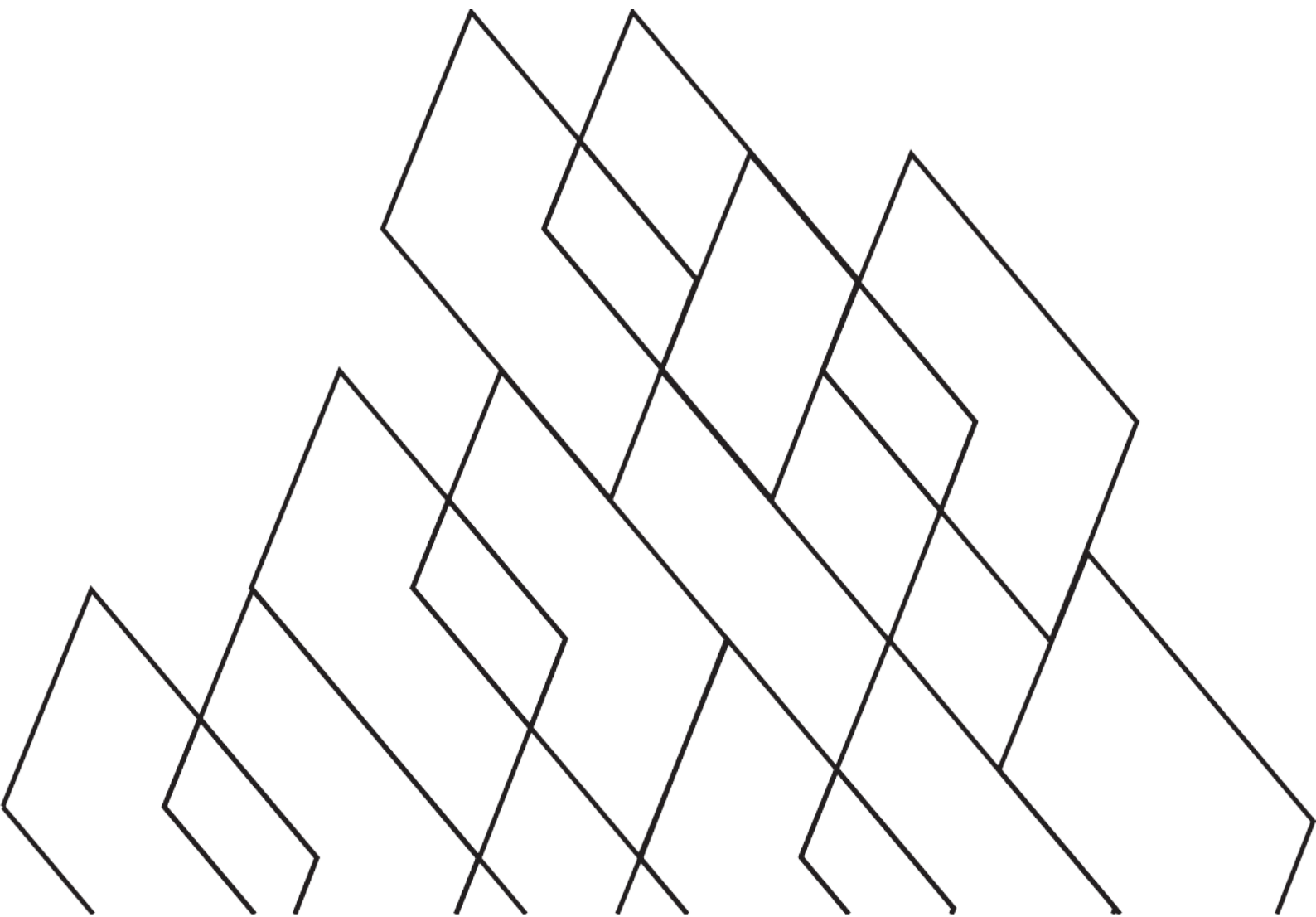


СЕРВИС ПО ВЫЯВЛЕНИЮ МАНИПУЛИРОВАНИЯ РЫНКОМ И
ИНСАЙДЕРСКОЙ ТОРГОВЛИ НА ФИНАНСОВЫХ РЫНКАХ

MARTA

Файловый API

Версия 1.24.0



Содержание

1. Общее описание	1
2. Стандартные загрузчики	2
3. Специализированные загрузчики	3
4. Встроенные загрузчики	4
5. Поля-ссылки	4
6. Примеры	5
7. Поддерживаемые сущности	7
7.1. Контрагенты	7
7.2. Инструменты	11
7.3. Трейдеры	12
7.4. Заявки	13
7.5. Сделки	15
7.6. Сотрудники	18
7.7. Режимы торгов	18
7.8. Курсы валют	19
7.9. Потиковые данные	20
7.10. Текущие цены	20
7.11. Контрольные списки	21
7.12. Инсайдеры	23
8. Часто задаваемые вопросы	25
8.1. Как настроить путь к директории, из которой загружаются файлы?	25
8.2. Как обеспечить требуемый порядок загрузки файлов?	25
8.3. Что делать с файлами, которые не удалось обработать?	26

1. Общее описание

Основным механизмом загрузки данных в сервис MARTA является его файловый API. Для поддержания его работы в сервисе настраивается специальная директория (далее обозначается как `input`), которую сервис постоянно проверяет, и в случае обнаружения в ней новых файлов производит загрузку данных из них.

При обработке файлов соблюдаются следующие правила:

- Файл обрабатывается только если сервису удастся получить эксклюзивный доступ к нему. Это гарантирует, что в случае, если копирование файла в директорию занимает длительное время, сервис не начнет его обработку до тех пор, пока копирование не завершится полностью.
- спешно обработанные файлы удаляются, их копия сервисом нигде не сохраняется.
- Файлы, при обработке которых возникли проблемы, остаются в директории `input` и не удаляются. Основные причины проблем: не удалось определить подходящий загрузчик, не удалось разобрать содержимое файла, при загрузке произошел технический сбой и т. п.
- Порядок обработки файлов в настоящее время никак не гарантируется.

Первым делом при обнаружении нового файла и получении эксклюзивного доступа к нему сервис производит поиск подходящего загрузчика для этого файла. Загрузчик определяется путем сопоставления имени файла с шаблонами имен, определенными всеми зарегистрированными в системе загрузчиками.

Например, загрузчик биржевых отчетов SEM02 срабатывает для файлов с именами формата `_SEM02_*.xml`, загрузчик файлов с информацией по раскрытию существенных фактов от Интерфакса — для файлов с именами формата `events_report_*.xls` и т.п.

Если для файла не удалось найти подходящего загрузчика, он не обрабатывается и остается в директории `input`.

Если загрузчик найти удалось, то производится разбор и импорт данных из файла с помощью этого загрузчика.

В системе имеется несколько встроенных загрузчиков, обрабатывающих стандартные файлы, предоставляемые внешними поставщиками данных (торговыми площадками, новостными агентствами и т. п.)

Кроме того, сервис MARTA поддерживает возможность конфигурирования собственных специализированных загрузчиков, импортирующих данные из файлов CSV.

Подробнее стандартные и специализированные загрузчики описаны в следующих разделах.

Вся информация об обнаруженных файлах, подобранных для них загрузчиках и о результатах обработки выводится в лог-файл сервиса.

2. Стандартные загрузчики

Набор стандартных загрузчиков, входящих в ядро сервиса MARTA:

Маска файлов	Описание
securities.csv	Список ценных бумаг, торгуемых на фондовой секции Московской биржи. Может быть загружен со специальной страницы на сайте биржи. Обычно используется для первоначальной инициализации БД сервиса.
ListingSecurityList.csv	Привязка ценных бумаг, торгуемых на Московской бирже, к эмитентам. Подробнее...
stock_current_price_*.csv	Внутридневные цены ценных бумаг на Московской бирже. Подробнее...
SEM21.xml	Биржевые отчеты SEM21 — информация о результатах торгов на фондовом рынке Московской биржи.
SEM02.xml	Биржевые отчеты SEM02 — информация о заявках участника торгов на фондовом рынке Московской биржи.
SEM03.xml	Биржевые отчеты SEM21 — информация о сделках участника торгов на фондовом рынке Московской биржи.
clients??00.csv	Биржевой отчет clientsXX00 — информация о клиентах участника торгов на срочном рынке Московской биржи.
f07_?????.csv и o07_?????.csv	Биржевые отчеты f07 и o07 — информация о результатах торгов на срочном рынке Московской биржи.
f04_?????.csv и o04_?????.csv	Биржевые отчеты f04 и o04 — информация о сделках участника торгов на срочном рынке Московской биржи.
fordlog?????.csv и oordlog?????.csv	Отчеты с заявками срочного рынка.
????????_fut_tick_.csv и ????????_opt_tick_.csv	Информация обо всех сделках с фьючерсами и опционами на срочном рынке Московской биржи. Формат определен сервисом биржи по распространению этих данных (Подробнее...).
CUX22.xml	Биржевые отчеты CUX22 — информация о заявках участника торгов на валютном рынке Московской биржи.
CUX23.xml	Биржевые отчеты CUX23 — информация о сделках участника торгов на валютном рынке Московской биржи.
CCX20.xml	Биржевые отчеты CUX23 — информация о торгово-клиринговых счетах участника торгов на валютном рынке Московской биржи.
OrderLog*.csv	Информация обо всех сделках на валютном рынке Московской биржи. Формат определен сервисом биржи по распространению этих данных (Подробнее...).
insiders-*.xml	Список инсайдеров участника торгов в формате, используемом в личном кабинете Московской биржи при ведении списка инсайдеров.

Маска файлов	Описание
events_report_*.xls	Информация о раскрытии существенных фактов эмитентами, распространяемая Интерфаксом.
Встроенные загрузчики MARTA	
marta-counterparties-*.csv	Справочник контрагентов и эмитентов (подробнее...).
marta-actual-employees-*.csv	Справочник сотрудников (подробнее...).
marta-instruments-*.csv	Справочник инструментов (подробнее...).
marta-future-contracts-*.csv	Описания контрактов срочного рынка Московской биржи.
marta-boards-*.csv	Справочник торговых режимов (подробнее...).
marta-trading-schedule-*.csv	Расписание торгов на различных рынках.
marta-ccy-rates-*.csv	Курсы валют (подробнее...).
marta-ticks-*.csv	Потиковые данные для расчета "текущих" цен (подробнее...).
marta-prices-*.csv	"Текущие" цены (подробнее...).
marta-control-lists-*.csv	Контрольные списки (подробнее...).
marta-control-list-items-*.csv	
marta-insiders-*.csv	Инсайдеры (подробнее...).

3. Специализированные загрузчики

Специализированные загрузчики определяются самостоятельно администраторами сервиса для решения конкретных задач по загрузке тех или иных данных в сервис. Каждый такой загрузчик позволяет определить правила для импорта данных из файлов формата CSV.

Для регистрации загрузчика требуется создать файл с расширением `.cfg`, в котором описываются правила обработки файлов с определенной маской имени. Каждая строка файла `.cfg` задает значение одного из параметров загрузчика в виде параметр: значение.

Параметры загрузчика:

Параметр	Обязательный	Описание	Пример
Type	да	Указывает, какие сущности будут содержаться в файлах, обрабатываемых данным загрузчиком.	counterparties
FileMask	да	Маска файлов, которые должны обрабатываться данным загрузчиком.	ctpt-*.csv
IdFields	да	Список полей, используемых для идентификации записей.	inn

Параметр	Обязательный	Описание	Пример
ValueFields	да	Список полей, значения которых загружаются в систему.	name, inn, is_client
OnAbsent	нет	Политика обработки записей, для которых не удалось найти соответствие. Допустимые значения: • <code>create</code> — создать новую запись, • <code>skip</code> — пропустить. По умолчанию <code>create</code>.	create
OnError	нет	Политика поведения при невозможности импортировать какую-либо запись. Допустимые значения: *` <code>stop`</code> — остановить обработку файла, * <code>skip</code> — пропустить запись и продолжить обработку. По умолчанию <code>skip</code>.	skip

Допустимые имена полей, указываемые в `IdFields` и `ValueFields`, зависят от той сущности, которая указана в `Type`. Ниже приведено описание всех поддерживаемых сущностей и их полей.

CSV-файлы, импортируемые с помощью таких специализированных загрузчиков должны быть в кодировке `UTF-8`.

4. Встроенные загрузчики

Встроенные загрузчики, в отличие от специализированных, не требуют создания отдельных конфигураций, так как их конфигурации встроены в сам сервис и структура обрабатываемых ими csv-файлов predetermined.

5. Поля-ссылки

Некоторые поля загружаемых сущностей являются ссылками на другие сущности. Например, у инструмента есть поле `issuer`, которое является ссылкой на эмитента (запись в справочнике контрагентов), а у сделки есть поле `instrument`, которое является ссылкой на финансовый инструмент.

В таких случаях при определении конфигурации загрузчика можно выбирать любые поддерживаемые в системе способы идентификации той сущности, на которую производится ссылка, добавляя к названию поля-ссылки название любого из полей справочной сущности, являющееся уникальным ключом (см. описание каждой поддерживаемой сущности ниже). Например, контрагентов можно идентифицировать по

ИНН (ключевое поле `inn`), по биржевому коду фирмы (ключевое поле `moex_fond_firm_id`) или по биржевому коду клиента (ключевое поле `moex_fond_client_code`), поэтому для привязки загружаемых инструментов к эмитентам (через поле `issuer`) можно использовать поля `issuer_inn`, `issuer_moex_firm_id`, `issuer_fond_client_code` или даже `issuer_id`, если известен внутренний идентификатор эмитента в сервисе MARTA.

Более того, для одного поля-ссылки в конфигурации загрузчика можно использовать сразу несколько идентифицирующих полей. Например, при загрузке инструментов можно указать сразу два поля для определения привязки к эмитенту. В этом случае сервис будет последовательно производить поиск по всем указанным в файле значениям, пытаясь найти нужную связь.

6. Примеры

Конфигурация для загрузки имен клиентов на фондовой секции Мос. биржи:

```
Type: counterparties
FileMask: moex-fond-clients.csv
IdFields: moex_fond_client_code
ValueFields: name
OnAbsent: skip
```

Такая конфигурация загрузчика говорит о следующем:

- `Type: counterparties` — в загружаемых файлах содержится информация о записях из справочника контрагентов.
- `FileMask: moex-fond-clients.csv` — он должен применяться для всех файлов с именем `moex-fond-clients.csv`.
- `IdFields: moex_fond_client_code` — при обработке каждой записи должен осуществляться поиск в справочнике по значению поля `moex_fond_client_code` (по биржевому коду клиента).
- `ValueFields: name` — для найденного контрагента обновляем значение поля `name` на указанное в файле.
- `OnAbsent: skip` — если соответствие в справочнике найти не удалось, то просто пропускаем запись.

Вот пример файла, который можно импортировать с помощью данного загрузчика (должен быть в кодировке UTF-8):

```
moex_fond_client_code,name
83223A,Иванова Е. И.
32733S,ООО Перспектива
```

При обработке такого файла в соответствии с заданной конфигурацией сервис попытается найти контрагента, у которого в качестве биржевого кода клиента указано значение `83223A`. Если ему это удастся, он установит этому контрагенту имя `Иванова Е. И.`. Если не удастся,

запись будет игнорирована.

Предположим теперь, что мы хотим внести в конфигурацию изменение: а) искать клиентов в справочнике контрагентов не только по биржевому коду клиента, но и по ИНН; б) в случае, если такого клиента найти в справочнике не удалось, регистрировать нового клиента, а не игнорировать запись.

Для этого в конфигурацию потребуется доработать следующим образом:

```
Type: counterparties
FileMask: moex-fond-clients.csv
IdFields: inn, moex_fond_client_code
ValueFields: name, inn, moex_fond_client_code, is_client
OnAbsent: create
```

В ней:

- **IdFields: inn, moex_fond_client_code** — мы добавили поле `inn`, чтобы по нему тоже осуществлялся поиск.
- **ValueFields: name, inn, moex_fond_client_code, is_client** — в списке полей со значениями указано не только имя, но и значения ИНН и биржевой код, а также признак того, что данная запись в справочнике контрагентов относится к клиенту (поле-флаг `is_client`).
- **OnAbsent: create** — если запись не удалось найти по указанным ключевым полям, создать новую запись.

Теперь можно импортировать такой файл:

```
inn,moex_fond_client_code,name,is_client
837459283744,83223A,Иванова Е. И.,yes
847239093200,32733S,ООО Перспектива,yes
```

При его импорте уже существующим клиентам будут устанавливаться указанные имена (и значения других полей), а отсутствующие клиенты будут регистрироваться в справочнике контрагентов.

Теперь предположим, что нам потребуется импортировать не клиентов, а информацию по сделкам, которые нельзя получить из биржевых отчетов (например, внебиржевые сделки). При этом мы будем в файле идентифицировать клиентов и контрагентов по ИНН, инструмент по коду ISIN, а трейдеров по логину в системе Bloomberg или по нашему внутреннему идентификатору. В этом случае конфигурация загрузчика будет выглядеть следующим образом:

```
Type: trades
FileMask: otc-trades-*.csv
IdFields: id
ValueFields: id, type, scope, date, operation_datetime, trader_bloomberg_user_login,
trader_internal_id, market_code, board_code, client_inn, counterparty_inn, side, instrument_isin,
```

```
price, quantity, value, settle_ccy_code, original_id  
OnAbsent: create
```

Пример файла, который можно загрузить с помощью такого загрузчика:

```
id,type,scope,date,operation_datetime,trader_bloomberg_user_login,trader_internal_id,market_code,board_code,client_inn,counterparty_inn,side,instrument_isin,price,quantity,value,settle_ccy_code,original_id  
BL-20180330-49480E3,BS,B,2018-03-30,2018-03-30T17:20:33,ivanov382,IVANOV,OTC,MAIN,837237093821,629937201297,B,RU000A0JQLB6,227.2,40,9080,RUB,20180330-49480E2
```

7. Поддерживаемые сущности

В данном подразделе приводится описание структуры сущностей, загрузка которых поддерживается на данный момент.

Следует иметь в виду, что обязательность поля в данном контексте означает, что оно должно присутствовать в списке `ValueFields` в тех случаях, когда установлено `OnAbsent=create`, то есть когда по результатам обработки записи в файле может быть создан новый экземпляр сущности. Если новые экземпляры не создаются, указывать в конфигурации и заполнять в файлах обязательные поля не требуется.

В описании полей используются следующие типы данных:

- `int, long` — целочисленные значения. Например: 0, 20, 388447.
- `decimal(n, m)` — число с плавающей запятой максимальной длины `n` с количеством знаков после запятой `m`.
- `str(n)` — строка длиной не более `n` символов. Например: Foo, Алибаба.
- `bool` — флаг да/нет, допустимые значения: `true, false, yes, no, y, n`.
- `date` — дата в формате уууу-ММ-дд. Например: 2019-01-29.
- `time` — время в формате hh:mm:ss. Например: 21:30:38.
- `datetime` — дата/время в формате уууу-ММ-дд`T`hh:mm:ss. Например: 2018-03-05T22:25:03.

7.1. Контрагенты

Загрузчик контрагентов и эмитентов обрабатывает файлы с именами `marta-counterparties-*.csv` со следующей структурой:

Поле	Ключевое	Тип	Обязательное	Описание
id	да	int	-	Уникальный идентификатор контрагента. Формируется автоматически.
name	-	str(200)	да	Краткое наименование контрагента.
full_name	-	str(200)	-	Полное наименование контрагента.
scope	-	str(1)	-	Тип области торговых операций. • U - неизвестно, • M - Д/У, • B - брокерская, • P - собственная.
type	-	str(1)	-	Тип контрагента. Одно из значений: • L — юр. лицо, • P — физ. лицо, • U — тип неизвестен. Если значение не указано, тип определяется по идентификационным данным (ИНН / номер паспорта). а^
country_id country_oksm_code country_name	-	country	-	Страна контрагента: ссылка на справочник стран. Для загрузки нужно указать одно любое из перечисленных полей.

Поле	Ключевое	Тип	Обязательное	Описание
residency	-	str(1)	-	Признак резидентства контрагента. Одно из значений: • R — резидент, • N — нерезидент, • U — тип резидентства неизвестен. Если значение не указано, тип определяется по идентификационным данным (формат ИНН или номера паспорта).
inn	да	str(12)	-	ИНН контрагента.
ogrn	-	str(200)	-	ОГРН контрагента.
moex_fond_firm_id	да	str(12)	-	Идентификатор фирмы на фондовой секции Московской биржи.
moex_fond_client_code	да	str(12)	-	Код клиента на фондовой секции Московской биржи.
moex_forts_client_code	да	str(12)	-	Код клиента на срочной секции Московской биржи.
moex_fx_client_code	да	str(12)	-	Код клиента на валютной секции Московской биржи.
spb_fond_client_code	да	str(12)	-	Код клиента на фондовой секции СПБ биржи.
passport_num	да	str(20)	-	Паспорт контрагента.
passport_date	-	date	-	Дата выдачи паспорта.
passport_issuer	-	str(5))	-	Кем выдан паспорт.
internal_id	-	str(5)	-	Внутренний идентификатор контрагента.
is_client	-	bool	-	Признак того, что данный контрагент является клиентом.

Поле	Ключевое	Тип	Обязательное	Описание
risk	-	int	-	Уровень риска клиента. Задаётся только если признак <code>is_client</code> = true.
config_profile	-	int	-	Идентификатор профиля риск-модели. Задаётся только если признак <code>is_client</code> = true.
is_issuer	-	bool	-	Признак того, что данный контрагент является эмитентом.
is_under_control	-	bool	-	Признак того, что данный клиент находится на контроле (в пользовательском интерфейсе отображается со специальным маркером).
details_1	-	str(100)	-	Произвольная дополнительная информация 1.
details_2	-	str(100)	-	Произвольная дополнительная информация 2.
client_status	-	str(500)	-	Статус клиента.
first_name	-	str(50)	-	Имя контрагента.
middle_name	-	str(50)	-	Отчество контрагента.
last_name	-	str(50)	-	Фамилия контрагента.
birth_date	-	date	-	Дата рождения контрагента.
birth_place	-	str(500)	-	Место рождения контрагента.
reg_address	-	str(500)	-	Адрес регистрации / юридический адрес контрагента.
real_address	-	str(500)	-	Фактический / почтовый адрес контрагента.
phone_number	-	str(50)	-	Номер телефона.
email	-	str(500)	-	E-mail контрагента.
website	-	str(500)	-	Web-сайт контрагента.

Поле	Ключевое	Тип	Обязательное	Описание
job_employer	-	counterparty	-	Идентификатор контрагента-работодателя: ссылка на справочник контрагентов. Для загрузки нужно указать одно любое ключевое поле, например, job_employer_inn.
job_title	-	str(500)	-	Должность.
tin	-	str(150)	-	TIN контрагента.
kio	-	str(10)	-	КИО контрагента.

7.2. Инструменты

Загрузчик инструментов обрабатывает файлы с именами `marta-instruments-*.csv` со следующей структурой:

Поле	Ключевое	Тип	Обязательное	Описание
id	да	int	-	Уникальный идентификатор инструмента.
symbol	да	str(100)	да	Уникальный код инструмента (обычно биржевой код).
alt_symbol	да	str(100)	-	Альтернативный код инструмента (не обязан быть уникальным).
type	-	str(2)	да	Тип. Допустимые значения: • EQ — акция, • BD — облигация, • DR - депозитарная расписка, • FX - валютная пара, • IX - индекс, • CD - товар, • FT — фьючерс, • OP - опцион.
name	-	str(200)	да	Наименование инструмента.

Поле	Ключевое	Тип	Обязательное	Описание
isin	да	str(12)	-	Код ISIN.
is_under_control	-	bool	-	Признак того, что данный клиент находится на контроле (в пользовательском интерфейсе отображается со специальным маркером).
is_blue_chip	-	bool	-	Признак голубой фишки.
issuer	-	counterparty	-	Эмитент — ссылка на справочник контрагентов (например, issuer_inn).
underlying_instrument	-	instrument	-	Для деривативов — ссылка на подлежащий актив (на справочник инструментов).
underlying_volume	-	long	-	Для деривативов — количество подлежащего актива.
expiration_date	-	date	-	Дата экспирации инструмента (для облигаций и деривативов).

7.3. Трейдеры

Загрузчик трейдеров обрабатывает файлы с именами `marta-traders-*.csv` со следующей структурой:

Поле	Ключевое	Тип	Обязательное	Описание
id	да	int	-	Уникальный идентификатор трейдера.
name	-	str(100)	да	Наименование.
internal_id	да	str(50)	-	Внутренний идентификатор (устанавливается на собственное усмотрение).

Поле	Ключевое	Тип	Обязательное	Описание
bloomberg_user_login	да	str(50)	-	Логин пользователя в терминале Bloomberg.
reuters_dealer_id	да	str(50)	-	Идентификатор трейдера в терминале Reuters.
moex_fond_user_id	да	str(12)	-	Идентификатор пользователя на фондовой секции Московской биржи.
moex_forts_user_id	да	str(12)	-	Идентификатор пользователя на срочной секции Московской биржи.
moex_fx_user_id	да	str(12)	-	Идентификатор пользователя на валютной секции Московской биржи.
spbxfond_user_id	да	str(12)	-	Идентификатор пользователя на фондовой секции СПБ биржи.

7.4. Заявки

Поле	Ключевое	Тип	Обязательное	Описание
id	да	str(50)	да	Уникальный идентификатор заявки.
status	-	str(1)	да	Статус. Допустимые значения: • F — исполнена, • C — отменена (по инициативе торговой площадки), • W — снята (по инициативе участника торгов или его клиента), • A — активна, • R — отклонена.

Поле	Ключевое	Тип	Обязательное	Описание
scope	-	str(1)	да	Вид. Допустимые значения: - Р — собственная, - В — брокерская, - М — дов. управление.
type	-	str(2)	да	Тип. Допустимые значения: BS — купля/продажа.
client	-	counterparty	да	Ссылка на клиента (в справочнике контрагентов, например: client_inn).
instrument	-	instrument	да	Ссылка на инструмент.
trader	-	trader	да	Ссылка на трейдера.
counterparty	-	counterparty	-	Ссылка на контрагента (например, counterparty_inn).
account	-	account	-	Ссылка на счет.
market_code	-	str(5)	да	Код рынка. Допустимые значения: - XMOS — Фондовая секция Московской биржи, - RTSX — Срочная секция Московской биржи, - SPBX — Фондовая секция Санкт-Петербургской биржи, - MXFX - Валютная секция Московской биржи, - ОТС — внебиржевой рынок.
board_code	-	str(10)	да	Код режима торгов на указанном рынке.

Поле	Ключевое	Тип	Обязательное	Описание
date	-	date	да	Дата торгов, к которой относится заявка. Может отличаться от реальной даты выставления, если торговая сессия активна в две разные даты.
operation_datetime	-	datetime	да	Дата/время выставления заявки.
last_datetime	-	datetime	-	Дата/время, когда заявка получила свой окончательный статус. Например, дата/время снятия, если она была снята, или дата/время последней сделки, если она была полностью исполнена.
side	-	str(1)	да	Направление. Допустимые значения: • B — купля, • S — продажа.
quantity	-	long	да	Количество актива.
price	-	decimal(18,10)	да	Цена.
value	-	decimal(21,4)	да	Сумма платежа.
settle_ccy_code		str(4)	-	Валюта платежа. Указывается трехсимвольный ISO-код. По умолчанию системная валюта (RUB).
original_id		str(50)	-	Оригинальный идентификатор заявки.

7.5. Сделки

Поле	Ключевое	Тип	Обязательное	Описание
id	да	str(50)	да	Уникальный идентификатор сделки (см. уточнение ниже).

Поле	Ключевое	Тип	Обязательное	Описание
scope	-	str(1)	да	Вид. Допустимые значения: - Р — собственная, - В — брокерская, - М — дов. управление.
type	-	str(2)	да	Тип. Допустимые значения: BS — купля/продажа.
client	-	counterparty	да	Ссылка на клиента (в справочнике контрагентов, например: client_inn).
instrument	-	instrument	да	Ссылка на инструмент.
trader	-	trader	да	Ссылка на трейдера.
counterparty	-	counterparty	-	Ссылка на контрагента, например: counterparty_inn.
account	-	account	-	Ссылка на счет.
market_code	-	str(5)	да	Код рынка. Допустимые значения: - XMOS — Фондовая секция Московской биржи, - RTSX — Срочная секция Московской биржи, - SPBX — Фондовая секция Санкт-Петербургской биржи, - MXFX - Валютная секция Московской биржи, - ОТС — внебиржевой рынок.
board_code	-	str(10)	да	Код режима торгов на указанном рынке.

Поле	Ключевое	Тип	Обязательное	Описание
date	-	date	да	Дата торгов, к которой относится сделка. Может отличаться от реальной даты заключения, если торговая сессия активна в две разные даты.
operation_datetime	-	datetime	да	Дата/время заключения сделки.
side	-	str(1)	да	Направление. Допустимые значения: • B — купля, • S — продажа.
quantity	-	long	да	Количество актива.
price	-	decimal(18,10)	да	Цена.
value	-	decimal(21,4)	да	Сумма платежа.
ai	-	decimal(21,4)	-	Сумма накопленного купонного дохода (НКД). Если не указана, принимается равной 0.
settle_ccy_code	-	str(4)	-	Валюта платежа. Указывается трехсимвольный ISO-код. По умолчанию системная валюта (RUB).
original_id	-	str(50)	-	Оригинальный идентификатор сделки (см. уточнение ниже).
order_id	-	str(50)	-	Уникальный идентификатор заявки, по которой была заключена сделка.

В структуре сделок присутствует два поля с идентификатором сделки: `id` и `original_id`. Разница между ними заключается в том, что `original_id` является идентификатором, присвоенным сделке той системой, через которую эта сделка была заключена. Например, это может быть биржевой код или код, присвоенный FIX-сервером. Теоретически, значения таких идентификаторов в разных системах могут пересекаться.

Поле `id` предназначено для хранения идентификатора сделки, уникального в рамках всей системы. Обычно для его формирования используется значение `original_id` с каким-либо префиксом. Например, сам сервис MARTA при загрузке сделок с фондовой секции Московской биржи для формирования уникального идентификатора добавляет к биржевому идентификатору префикс `MX-` и суффикс, указывающий на направление (`-B` или `-S`).

Задача обеспечения уникальности лежит на том, кто производит загрузку данных в сервис MARTA через данный API, так как при пересечении значений поля id существующая сделка перезапишется.

7.6. Сотрудники

Загрузчик списка сотрудников обрабатывает файлы с именами `marta-actual-employees-{дата в формате ГГГГ-ММ-ДД}*.csv` со следующей структурой:

Поле	Ключевое	Тип	Обязательное	Описание
employee_passport_num	да	str(20)	да	Номер паспорта сотрудника.
comment	-	str(100)	нет	Комментарий.

Наличие в файле записи с номером паспорта интерпретируется как то, что запись о контрагенте с указанным номером паспорта относится к сотруднику, причем запись о найме действует на дату, указанную в имени файла.

Отсутствие в файле записи с номером паспорта какого-либо сотрудника интерпретируется как то, что запись об этом сотруднике больше не является актуальной на дату, указанную в имени файла.

Используя эти правила, сервис самостоятельно создает и редактирует записи о найме. То есть для автоматического ведения списка сотрудников желательно регулярно загружать в сервис файл с полным списком номеров паспортов сотрудников, по которым требуется отслеживать операции.

Следует иметь в виду, что на основании загружаемых данных сервис управляет записями о найме, относящимися только к уже имеющимся в справочнике контрагентов записям. На основании информации из данного файла новые записи в справочнике контрагентов автоматически не регистрируются.

Пример:

```
employee_passport_num,comment
11 22 334455,Главный бухгалтер
99 88 726556,Старший трейдер
80 70 605040,Старший трейдер
```

7.7. Режимы торгов

Загрузчик режимов торгов обрабатывает файлы с именами `marta-boards-*.csv` со следующей структурой:

Поле	Ключевое	Тип	Обязательное	Описание
code	да	str(10)	да	Код торгового режима (уникальный в рамках рынка).
name	-	str(100)	да	Наименование торгового режима.
market_code	да	str(4)	да	Код рынка, к которому относится торговый режим.
type	-	str(2)	да	Тип торгового режима: • MN — основной, • NG — РПС, • AU — аукционный, • SL — малые лоты, • LL — большие лоты, • SP — специальный. Для загрузки ОТС сделок возможны режимы MN /NG.
trade_type	-	str(2)	да	Тип сделок, заключаемых в данном режиме: • BS — купля/продажа, • PR — РЕПО, • SW — своп.
ccy_code	-	str(3)	да	Код валюты, в которой происходят расчеты в данном режиме.

7.8. Курсы валют

Загрузчик курсов валют обрабатывает файлы с именами `marta-ccy-rates-*.csv` со следующей структурой:

Поле	Ключевое	Тип	Обязательное	Описание
date	да	date	да	Дата, на которую действует курс валюты.
ccy_code	да	time	да	Код валюты, для которой задан курс.

Поле	Ключевое	Тип	Обязательное	Описание
rate	-	str(4)	да	Курс указанной валюты относительно системной валюты.

7.9. Потокные данные

Загрузчик потокных данных обрабатывает файлы с именами `marta-ticks-*.csv` со следующей структурой:

Поле	Ключевое	Тип	Обязательное	Описание
market_code	-	str(4)	да	Код рынка, на котором заключена сделка.
board_code	-	str(10)	да	Код торгового режима, в котором заключена сделка.
date	-	date	да	Дата заключения сделки.
time	-	time	да	Время заключения сделки.
instrument_symbol	-	str(20)	да	Торговый код инструмента.
price	-	decimal(20, 10)	да	Цена в сделке.
quantity	-	decimal(20, 10)	да	Количество инструмента.

Пример:

```
market_code,board_code,date,time,instrument_symbol,price,quantity
SPBX,SPBX,2021-06-12,07:12:15,SPBX:AAPL,330.23,800
SPBX,SPBX,2021-06-12,07:12:15,SPBX:AAPL,330.25,100
SPBX,SPBX,2021-06-12,07:12:16,SPBX:AAPL,330.24,50
SPBX,SPBX,2021-06-12,10:59:09,SPBX:AAPL,330.30,9
```

7.10. Текущие цены

Загрузчик текущих цен (с которыми сравниваются цены анализируемых торговых операций) обрабатывает файлы с именами `marta-prices-*.csv` со следующей структурой:

Поле	Ключевое	Тип	Обязательное	Описание
market_code	-	str(4)	да	Код рынка, на котором заключена сделка.

Поле	Ключевое	Тип	Обязательное	Описание
board_code	-	str(10)	да	Код торгового режима, в котором заключена сделка.
date	-	date	да	Дата заключения сделки.
time	-	time	да	Время заключения сделки с точностью до минуты.
instrument_symbol	-	str(20)	нет	Торговый код инструмента. Поле может отсутствовать или не заполняться если присутствует ISIN инструмента.
instrument_isin	-	str(20)	нет	ISIN инструмента. Поле может отсутствовать или не заполняться если присутствует торговый код инструмента.
price	-	decimal(20, 10)	да	Цена в сделке.

Пример:

```
market_code,board_code,date,time,instrument_symbol,price
SPBX,SPBX,2021-06-12,07:12,SPBX:AAPL,330.23
SPBX,SPBX,2021-06-12,07:13,SPBX:AAPL,330.25
SPBX,SPBX,2021-06-12,07:14,SPBX:AAPL,330.24
SPBX,SPBX,2021-06-12,10:59,SPBX:AAPL,330.30
```

7.11. Контрольные списки

Для загрузки в сервис контрольных списков используются два загрузчика: первый загружает общую информацию о контрольных списках, второй — информацию о входящих в списки эмитентах и инструментах.

Загрузчик общей информации о контрольных списках обрабатывает файлы с именами `marta-control-lists-*.csv` со следующей структурой:

Поле	Ключевое	Тип	Обязательное	Описание
internal_id	да	str(50)	да	Внутренний (в мастер-системе) идентификатор контрольного списка, по которому он будет идентифицироваться в других загрузчиках.
name	-	str(50)	да	Наименование списка.
type	-	str(1)	да	Назначение списка: • S — стоп-лист, • W — лист наблюдения, • L — обычный список.
legal_type	-	str(1)	да	Пункт статьи 4 224-ФЗ (тип списка).
organization	-	counterparty	да	Ссылка на организацию списка (например, counterparty_inn).
comment	-	str(500)	-	Комментарий.

Пример:

```
internal_id,name,type,start_date,end_date,comment,legal_type,organization_inn
list-1,Все бумаги Am Airlines,L,2021-01-01,,Список под клиента 164449,5,7788998877
list-2,Тестовый список,L,2025-02-05,2025-10-11,Проверка и тестирование,4,7788998877
```

Загрузчик состава контрольных списков обрабатывает файлы с именами marta-control-list-items-*.csv со следующей структурой:

Поле	Ключевое	Тип	Обязательное	Описание
internal_id	да	str(50)	да	Внутренний идентификатор записи.
control_list_internal_id	да	str(50)	да	Внутренний идентификатор контрольного списка, к которому относится запись.
start_date	-	date	да	Дата начала действия записи.

Поле	Ключевое	Тип	Обязательное	Описание
end_date	-	date	-	Дата окончания действия записи. Если не указана, запись действует по настоящее время.
issuer	-	counterparty	-	Эмитент — ссылка на справочник контрагентов (например, issuer_inn).
instrument	-	instrument	-	Инструмент — ссылка на справочник инструментов (например, instrument_isin).
is_derivatives_included	-	boolean	-	Признак включения производных финансовых инструментов.

В каждой записи должна быть обязательно заполнена либо ссылка на эмитента, либо ссылка на инструмент, но не оба поля одновременно.

Обновление состава происходит только для списков, для которых в файле есть записи. Отсутствующие записи не удаляются. Для удаления необходимо выгрузить запись с закрытой датой окончания действия.

Пример (здесь эмитенты идентифицируются по ИНН, а инструменты по ISIN):

```
internal_id,control_list_internal_id,start_date,end_date,issuer_inn,instrument_isin,is_derivatives_included
items-1,list-1,2021-01-01,,,US02376R1023,1
```

7.12. Инсайдеры

Загрузчик инсайдеров обрабатывает файлы с именами marta-insiders-*.csv со следующей структурой:

Поле	Ключевое	Тип	Обязательное	Описание
internal_id	да	str(50)	да	Внутренний идентификатор инсайдера.
counterparty_internal_id	-	str(50)	да	Контрагент-инсайдер (чаще всего клиент) — ссылка на справочник контрагентов.

Поле	Ключевое	Тип	Обязательное	Описание
type	-	str(1)	да	Тип инсайдера. Всегда устанавливается значение L — инсайдер в отношении определенного списка эмитентов/инструментов.
control_list_internal_id	да	str(50)	да	Внутренний идентификатор контрольного списка, в отношении которого указанный контрагент признан инсайдером.
start_date	-	date	да	Дата начала действия записи.
end_date	-	date	-	Дата окончания действия записи. Если не указана, запись действует по настоящее время.
comment	-	str(500)	-	Комментарий.
legal_type	-	int	да	Тип инсайдера согласно статье 4 224-ФЗ
legal_type_description	-	str(5000)	-	Текстовое описание типа инсайдера. Используется для уточнения legal_type в исключительных случаях. Не рекомендуется к заполнению.
instrument_description	-	str(5000)	-	Текстовое описание инструментов в отношении которых инсайдер обладает инсайдерской информацией
include_reason	-	str(5000)	да	Основание включения инсайдера в список
exclude_reason	-	str(5000)	*	Основание исключения инсайдера из списка. Обязательно должно быть заполнено, если end_date установлена.

Пример (здесь инсайдеры идентифицируются по `internal_id`):

```
internal_id,counterparty_internal_id,type,control_list_internal_id,start_date,end_date,comment,legal_type,instrument_description,include_reason,exclude_reason
ТДК/21-321,test_insider_1,L,default_self_list,2021-05-10,2022-02-08,,5,,Трудовой договор ТДК/21-321 от 10.05.2021,Увольнение с занимаемой должности
ТДК/22-11,test_insider_1,L,default_self_list,2022-02-09,,,5,,Трудовой договор ТДК/22-11 от 02.09.2022,
```

8. Часто задаваемые вопросы

8.1. Как настроить путь к директории, из которой загружаются файлы?

Путь к директории, в которой сервис производит поиск файлов для загрузки, настраивается параметром `marta.input.dir` в конфигурационном файле сервиса. Допускается указание как полного пути, так и относительного из рабочей директории сервиса (по умолчанию ею является директория установки).

Возможно размещение директории `input` на внешних сетевых устройствах. Главным требованием является наличие доступа на чтение и запись к этой директории у того пользователя, от имени которого запускается сервис MARTA.

При старте сервис производит проверку наличия директории `input`, а также наличия достаточных прав на работу с ней. Если директория не найдена, либо у сервиса недостаточно прав, в лог-файл будет выведено соответствующее сообщение и сервис не запустится.

8.2. Как обеспечить требуемый порядок загрузки файлов?

Как описано в начале статьи, в настоящее время порядок загрузки файлов никак не гарантируется. Единственным способом обеспечить требуемый порядок обработки является контролируемое копирование файлов в директорию `input`. Это значит, что необходимо копировать файлы по одному и дожидаться удаления (или начала обработки, о которой будет свидетельствовать невозможность получить эксклюзивный доступ к файлу) уже скопированного файла прежде, чем приступить к копированию нового.

В будущем, возможно, мы добавим возможность определять для загрузчиков приоритеты и обрабатывать файлы в том порядке, который соответствует приоритетам соответствующих загрузчиков, но даже при этом будут оставаться определенные сложности в соблюдении порядка. Это связано с тем, что при копировании множества файлов в директорию с помощью стандартных средств операционной системы (например, через проводник в Windows) порядок тоже не гарантируется, а это значит, что даже несмотря на наличие упорядочивания на стороне сервиса гарантировать порядок загрузки глобально можно будет только с помощью описанного выше контролируемого копирования.

8.3. Что делать с файлами, которые не удалось обработать?

В настоящее время файлы, которые по каким-либо причинам не удалось обработать, остаются в директории `input`. Сервис запоминает имена таких проблемных файлов и при очередной проверке игнорирует их.

При этом в случае удаления файла информация о его имени удаляется из списка проблемных файлов и при необходимости его можно скопировать снова, устранив причину изначальной проблемы.

Например, если в директорию `input` был скопирован файл с некорректным форматом данных, его можно просто удалить и через несколько секунд поместить в директорию заново, тогда сервис постарается обработать его как полностью новый.