Содержание

- 1. Управление биржами
 - 1. Администратор просматривает биржи
 - 2. Администратор добавляет биржу
 - 3. Администратор удаляет биржу
- 2. Работа с символами
 - 1. Администратор просматривает символы
 - 2. Администратор просматривает историю статусов символов
 - 3. Сторонний сервис получает список символов
 - 4. Сторонний сервис получает историю статусов символов
- 3. Фоновые задачи синхронизации
 - 1. Успешное выполнение фоновой задачи
 - 2. Ошибка при обращении к базе данных на старте
 - 3. Ошибка при обращении к API Binance
 - 4. Ошибка при сохранении обновлений в базу данных
 - 5. Отключенные стандартные уведомления
 - 6. Завершение фоновой задачи с ошибкой
- 4. <u>Управление статусами символов</u>
 - 1. <u>Администратор вручную добавляет символ (AddedByAdmin)</u>
 - 2. <u>Администратор вручную удаляет символ (RemovedByAdmin)</u>
 - 3. <u>Символ с RemovedByAdmin вручную возвращён администратором</u>
 - 4. <u>Символ AddedByAdmin найден на бирже автоматическое обновление</u>
 - 5. Символ с RemovedByAdmin присутствует в ответе биржи
 - 6. <u>Символ отсутствует в ответе биржи → статус Delisted</u>
 - 7. <u>Биржа присылает новый символ → создаётся с биржевым статусом</u>
 - 8. Биржа изменила статус символа
 - 9. <u>Символ снова появился после Delisted → создаётся заново</u>
- 5. Управление настройками уведомлений
 - 1. Администратор просматривает настройки уведомлений
 - 2. Администратор изменяет настройки уведомлений
 - 3. Администратор пытается включить несконфигурированные уведомления
 - 4. Отправка критических уведомлений
 - 5. Отправка стандартных уведомлений
- 6. Инициализация и восстановление системы
 - 1. Первый запуск системы
 - 2. Восстановление после сбоя

1. Управление биржами

1.1. Администратор просматривает биржи

Актор: Администратор

Предусловия:

• Система функционирует в рабочем режиме

Основной поток:

- 1. Администратор выполняет HTTP GET запрос /api/v1/exchanges
- 2. Система возвращает список зарегистрированных бирж

Результат:

 Администратор получает список зарегистрированных бирж. В случае отсутствия зарегистрированных бирж, возвращается пустой список.

1.2. Администратор добавляет биржу

Актор: Администратор

Предусловия:

- Система функционирует в рабочем режиме
- Биржа с указанным именем не существует в системе

Основной поток:

Администратор выполняет HTTP POST запрос на /api/v1/exchanges с телом: {
 "exchangeName": "binance" }

```
Post /api/v1/exchanges

Parameters

No parameters

Request body

{
  "name": "BinanceCoinFutures|"
}
```

- 2. Система проверяет наличие биржи с указанным именем
- 3. При отсутствии дублирования, система добавляет новую биржу в базу данных
- 4. Возвращается статус 201 Created с информацией о созданной бирже

```
Code

Response body

{
    "id": "019667fa-9424-7b6a-9803-4fd5f0e56075",
    "name": "BinanceCoinFutures",
    "createdAt": "2025-04-24T13:26:37.3462796Z"
}

Response headers

content-type: application/json; charset=utf-8
date: Thu,24 Apr 2025 13:26:36 GMT
location: http://localhost:61578/api/v1/exchanges/BinanceCoinFutures
server: Kestrel
transfer-encoding: chunked
```

Альтернативный поток:

При обнаружении биржи с таким же именем, система возвращает статус 400 Bad
 Request и соответствующее сообщение об ошибке

Результат:

 Новая биржа зарегистрирована в системе, либо администратор получает уведомление о невозможности регистрации

1.4. Администратор удаляет биржу

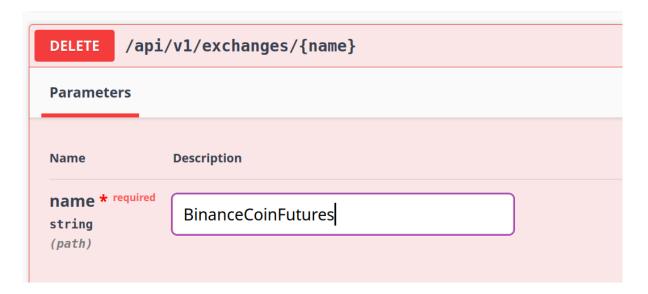
Актор: Администратор

Предусловия:

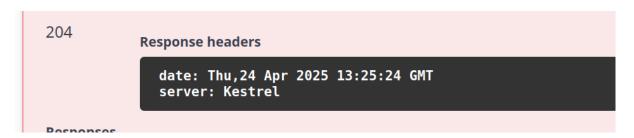
- Система функционирует в рабочем режиме
- Биржа с указанным exchangeName существует в системе

Основной поток:

Администратор выполняет HTTP DELETE запрос на /api/v1/exchanges/{exchangeName}



- 2. Система проверяет существование биржи
- 3. Система удаляет биржу из базы данных
- 4. Возвращается статус 204 No Content



Альтернативный поток:

- При отсутствии биржи с указанным именем:
 - Возвращается статус 404 Not Found

Результат:

 Биржа удалена из системы, либо администратор получает уведомление об отсутствии биржи с указанным именем

2. Работа с символами

2.1. Администратор просматривает символы

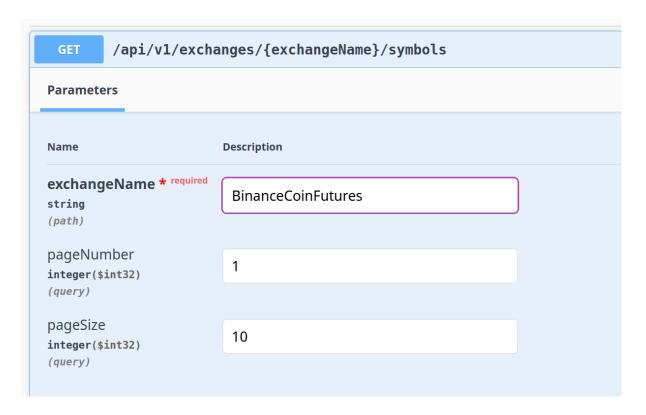
Актор: Администратор

Предусловия:

• Система функционирует в рабочем режиме

Основной поток:

1. Администратор выполняет HTTP GET запрос /api/v1/exchanges/{exchangeName}/symbols, указывая номер страницы и размер (эндпоинт поддерживает пагинацию)



- 2. Система проверяет существование указанной биржи
- 3. Система возвращает информацию о символах для данной биржи

```
Tode

Details

Response body

[
{
    "id": "c08df830-e4d0-41df-913c-05e877eeaa14",
    "exchange4d": "019667fa-9424-7b6a-9803-4fd5f0e56075",
    "symbolName": "AAVEUSD_PERP",
    "marketType": "CoinFutures",
    "baseAsset": "AVEUS",
    "quoteAsset": "USD",
    "status": "Active",
    "pricePrecision": 3,
    "quantityPrecision": 0,
    "contractType": "Perpetual",
    "deliveryDate": "2100-12-25T08:00:002",
    "marginAsset": "AAVE",
    "minOuantity": 1,
    "minNotional": 0,
    "maxQuantity": 100000,
    "updatedAt": "2025-04-24T13:27:00.645478Z"
},
    {
        "id": "9eee5456-db05-4e8f-ab8e-02c60cf27578",
        "exchangeId": "019667fa-9424-7b6a-9803-4fd5f0e56075",
        "symbolName": "ADAUSD_250627",
        "marketType": "CoinFutures",
        "baseAsset": "NDD",
        "quoteAsset": "USD",
        "status": "Active",
        "pricePrecision": 5,
```

Альтернативный поток:

 При указании несуществующего имени биржи, система возвращает статус 404 Not Found

Результат:

Администратор получает список символов в виде страницы с указанным размером



Также администратор может просматривать только активые символы или конкретный символ.

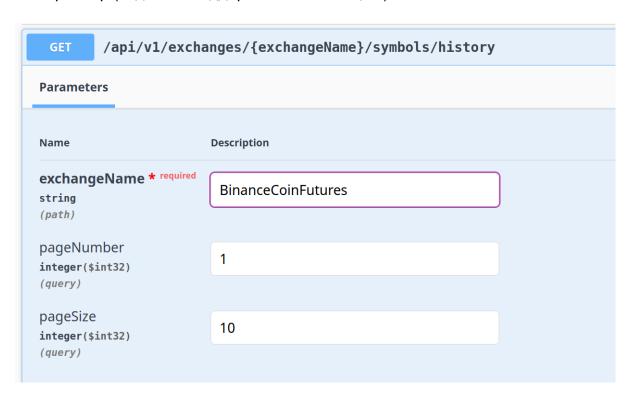
2.2. Администратор просматривает историю статусов символов

Актор: Администратор

Предусловия:

• Система функционирует в рабочем режиме

1. Администратор выполняет HTTP GET запрос /api/v1/exchanges/{exchangeName}/symbols/history, указывая номер страницы и размер (эндпоинт поддерживает пагинацию)



- 2. Система проверяет существование указанной биржи
- 3. Система возвращает историю изменения статусов символов для данной биржи

Альтернативный поток:

 При указании несуществующего имени биржи, система возвращает статус 404 Not Found

Результат:

 Администратор получает историю статусов символов для данной биржи в виде страницы с указанным размером



Также администратор может просматривать историю только активых символов или историю конкретного символа.

2.3. Сторонний сервис получает список символов

Актор: Сторонний сервис

Предусловия:

• Система функционирует в рабочем режиме

Основной поток:

- 1. Сторонний сервис выполняет HTTP GET запрос /api/v1/exchanges/{exchangeName}/symbols-list
- 2. Система проверяет существование указанной биржи
- 3. Система возвращает полный список символов для данной биржи

Альтернативный поток:

 При указании несуществующего имени биржи, система возвращает статус 404 Not Found

Результат:

• Сторонний сервис получает полный список символов для указанной биржи



Эндпоинты с суффиксом -list не поддерживают пагинацию и предназначены для использования другими сервисами, а не администратором.

2.4. Сторонний сервис получает историю статусов символов

Актор: Сторонний сервис

Предусловия:

• Система функционирует в рабочем режиме

Основной поток:

- 1. Сторонний сервис выполняет HTTP GET запрос /api/v1/exchanges/{exchangeName}/symbols-list/history
- 2. Система проверяет существование указанной биржи
- 3. Система возвращает полную историю изменения статусов символов для данной биржи

Альтернативный поток:

При указании несуществующего имени биржи, система возвращает статус 404 Not Found

Результат:

• Сторонний сервис получает полную историю статусов символов указанной биржи



Эндпоинты с суффиксом -list не поддерживают пагинацию и предназначены для использования другими сервисами, а не администратором.

3. Фоновые задачи синхронизации

3.1. Успешное выполнение фоновой задачи

Актор: Система (фоновая задача)

Предусловия:

- Фоновая задача запланирована и активна
- Система имеет корректный доступ к базе данных и API Binance

- 1. Система извлекает список текущих символов из базы данных
- 2. Система выполняет запрос к API Binance

- 3. Полученные данные преобразуются в модель символов
- 4. Система выполняет сравнение полученных символов с сохранёнными в базе:
 - Идентифицирует новые символы
 - Идентифицирует символы с измененным статусом
 - Идентифицирует отсутствующие символы
- 5. При обнаружении изменений:
 - Система формирует стандартные уведомления
 - Отправляет уведомления через Telegram (если Telegram-уведомления сконфигурированы и активированы)
 - Отправляет уведомления по email (если email-уведомления сконфигурированы и активированы)
- 6. Система сохраняет обновленные данные в базу данных

Результат: Фоновая задача завершается успешно, база данных синхронизирована, администратор получает уведомления об изменениях через настроенные каналы связи.

3.2. Ошибка при обращении к базе данных на старте

Актор: Система

Предусловия:

- Фоновая задача активна
- Отсутствует подключение к базе данных

- 1. Система инициирует запрос на получение символов из базы данных
- 2. Возникает ошибка подключения к базе данных
- 3. Система формирует критическое уведомление
- 4. Система отправляет критическое email-уведомление администратору (если критические уведомления сконфигурированы)

Database Error

Operation: operation_name

Time: 2025-04-11 14:34:52 UTC

•••

Error Message: Unexpected error occurred

Exception Details:

Type: InfoSymbolServer. Application. Exceptions. NotFoundException

Message: The requested resource was not found.

Stack Trace:

Please take immediate action to resolve this database issue.

5. Фоновая задача завершает выполнение

Результат: Синхронизация не выполнена. Администратор получает критическое уведомление о возникшей ошибке.

3.3. Ошибка при обращении к API Binance

Актор: Система

Предусловия:

- Получение данных из базы данных выполнено успешно
- Запрос к API Binance завершается с ошибкой

- 1. Система отправляет запрос к API Binance
- 2. Возникает ошибка при обработке запроса (например, ошибка 500 Internal Server Error, недоступность API)
- 3. Система формирует критическое уведомление
- 4. Система отправляет критическое email-уведомление администратору (если критические уведомления сконфигурированы)

ALERT: Exchange API Error - Binance Inbox x



Exchange API Error

Exchange: Binance **Time:** 2025-04-11 09:43:47 UTC

Error Message: Failed to fetch data from external API

Please take immediate action to resolve this issue.



5. Фоновая задача завершает выполнение

Результат: Синхронизация не выполнена, администратор получает критическое уведомление о возникшей ошибке.

3.4. Ошибка при сохранении обновлений в базу данных

Актор: Система

Предусловия:

- Символы успешно получены и обработаны
- Изменения идентифицированы
- Возникает ошибка при сохранении данных

Основной поток:

- 1. Система пытается записать обновленные данные в базу
- 2. Возникает ошибка при выполнении операции записи
- 3. Система формирует критическое уведомление
- 4. Система отправляет критическое email-уведомление (если критические уведомления сконфигурированы)
- 5. Фоновая задача завершает выполнение

Результат: База данных не обновлена, администратор получает критическое уведомление о возникшей ошибке.

3.5. Отключенные стандартные уведомления

Актор: Система

Предусловия:

- Процесс обновления символов выполнен успешно
- Конфигурация системы указывает на отключенные стандартные уведомления (Telegram и/или email)

Основной поток:

- 1. Система обнаруживает изменения в данных символов
- 2. Система проверяет настройки уведомлений:
 - Если Telegram-уведомления отключены, система не отправляет сообщения в Telegram
 - Если email-уведомления отключены, система не отправляет стандартные emailсообщения
- 3. Обновленные данные сохраняются в базу данных

Результат: База данных обновлена, уведомления отправляются только по активированным каналам связи в соответствии с конфигурацией системы.

3.6. Завершение фоновой задачи с ошибкой

Актор: Система

Предусловия:

• Текущая фоновая задача завершается с ошибкой

- 1. Фоновая задача завершается из-за ошибки (сценарии 3.2, 3.3, 3.4)
- 2. Основная система продолжает функционировать
- 3. Другие запланированные задачи продолжают выполняться по расписанию

System Error

Component: component

Time: 2025-04-11 09:41:21 UTC

Error Message: Unexpected error occurred

Exception Details:

Type: System.Exception

Message: gjajgrpgn epng apreng ag

Stack Trace:

Please investigate the system error as soon as possible.

Результат: Ошибка фоновой задачи обработана, основная система сохраняет работоспособность, администратор получает соответствующее критическое уведомление (если сконфигурировано).

4. Управление статусами символов

4.1. Администратор вручную добавляет символ (AddedByAdmin)

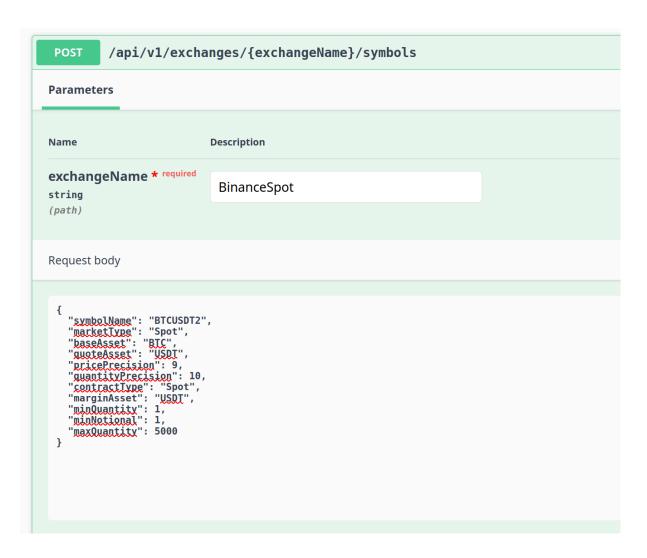
Актор: Администратор

Предусловия:

- Символ отсутствует в базе данных
- Фоновая задача для соответствующей биржи активна

Основной поток:

1. Администратор отправляет запрос на добавление нового символа



2. Система создает запись для нового символа в базе данных со статусом AddedByAdmin

```
Code

Response body

{
    "id": "7a87ee09-7d57-4fd4-bb87-e07028ee30b8",
    "exchangeId": "bc27b1f1-8c70-40e8-892e-e299b5be0ea7",
    "symbolName": "BTCUSDT2",
    "marketType": "Spot",
    "baseAsset": "BTC",
    "quoteAsset": "USDT",
    "status": "AddedByAdmin",
    "pricePrecision": 9,
    "quantityPrecision": 10,
    "contractType": "Spot",
    "deliveryDate": null,
    "marginAsset": "USDT",
    "minQuantity": 1,
    "minNotional": 1,
    "maxQuantity": 5000,
    "updatedAt": "2025-04-24T13:34:39.2936296Z"
}
```

Результат: Символ регистрируется в системе со статусом AddedByAdmin и ожидает потенциального появления на бирже.

4.2. Администратор вручную удаляет символ (RemovedByAdmin)

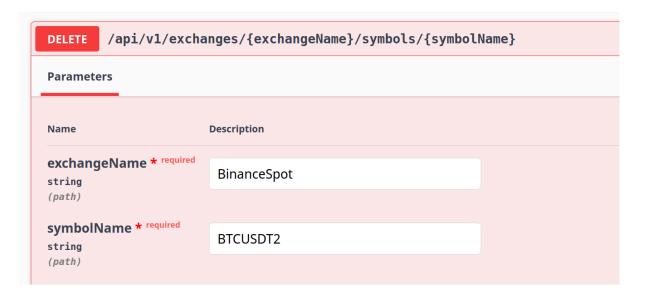
Актор: Администратор

Предусловия:

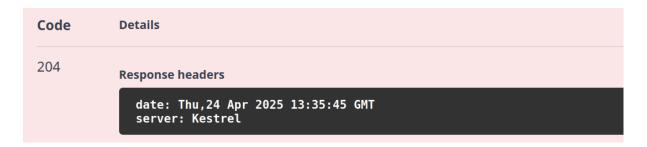
- Символ присутствует в базе данных
- Символ может иметь любой текущий статус

Основной поток:

1. Администратор отправляет запрос на удаление символа



2. Система не удаляет запись из базы данных, а изменяет её статус на RemovedByAdmin



3. Символ исключается из последующих процессов синхронизации

Результат: Символ сохраняется в базе данных, но исключается из процессов обновления и обработки системой.

4.3. Символ с RemovedByAdmin вручную возвращён администратором

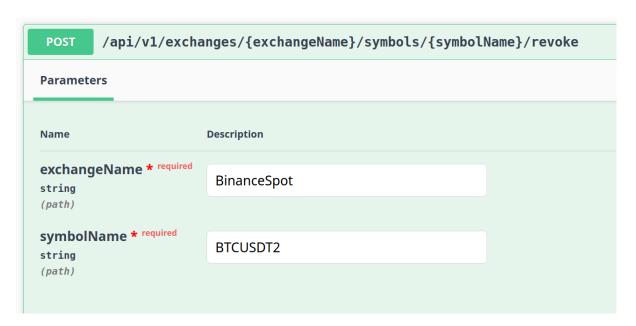
Актор: Администратор

Предусловия:

• Символ имеет статус RemovedByAdmin

Основной поток:

1. Администратор отправляет запрос на изменение статуса



2. Система изменяет статус на AddedByAdmin

```
Code Details

Response body

{
    "id": "7a87ee09-7d57-4fd4-bb87-e07028ee30b8",
    "exchangeId": "bc27b1f1-8c70-40e8-892e-e299b5be0ea7",
    "symbolName": "BTCUSDT2",
    "marketType": "Spot",
    "duoteAsset": "USDT",
    "status": "AddedByAdmin",
    "pricePrecision": 9,
    "quantityPrecision": 10,
    "contractType": "Spot",
    "deliveryDate": null,
    "marginAsset": "USDT",
    "minQuantity": 1,
    "minNotional": 1,
    "maxQuantity": 5000,
    "updatedAt": "2025-04-24T13:36:24.57186792"
}
```

3. Символ включается в процесс синхронизации

Результат: Символ возвращается в процесс синхронизации.

4.4. Символ AddedByAdmin найден на бирже — автоматическое обновление

Актор: Система (фоновая задача)

Предусловия:

- В базе данных существует символ со статусом AddedByAdmin
- Биржа возвращает данный символ в ответе API

Основной поток:

- 1. В процессе синхронизации система получает список символов от биржи
- 2. Среди полученных символов идентифицируется символ, ранее добавленный вручную (AddedByAdmin)
- 3. Система обновляет статус символа на полученный от биржи (например, Active, Suspended и т.д.)

Результат: Символ включается в стандартный процесс синхронизации и получает статус, соответствующий состоянию на бирже.

4.5. Символ с RemovedByAdmin присутствует в ответе биржи

Актор: Система (фоновая задача)

Предусловия:

• Символ со статусом RemovedByAdmin возвращается биржей через API

Основной поток:

- 1. Фоновая задача получает данные символа от биржи
- 2. Система идентифицирует в базе данных символ со статусом RemovedByAdmin
- 3. Система игнорирует полученные данные не обновляет статус символа и не включает его в уведомления

Результат: Символ сохраняет специальный статус и исключается из процесса обработки.

4.6. Символ отсутствует в ответе биржи \rightarrow статус Delisted

Актор: Система (фоновая задача)

Предусловия:

- Символ ранее существовал в базе данных и имел статус биржи (например, Active, Suspended)
- Символ отсутствует в текущем ответе биржи

Основной поток:

- 1. Система не обнаруживает символ в ответе API
- 2. Система регистрирует данное событие как делистинг символа
- 3. Статус символа в базе данных обновляется на Delisted
- 4. Формируется и отправляется уведомление через Telegram (если функция активирована)

Результат: Символ получает статус Delisted и исключается из ожидаемых обновлений от биржи.



Если в последующих обновлениях символ снова появится в ответе биржи, см. сценарий <u>4.9</u>.

4.7. Биржа присылает новый символ → создаётся с биржевым статусом

Актор: Система (фоновая задача)

Предусловия:

- Символ отсутствует в базе данных
- Биржа включает символ в ответ API

Основной поток:

- 1. Фоновая задача получает данные нового символа от биржи
- 2. Система создает запись в базе данных и присваивает статус, полученный от биржи (например, Active, Pre-launch и т.д.)
- 3. Система формирует и отправляет уведомление через Telegram (если функция активирована)

Результат: Новый символ регистрируется в системе и включается в процесс мониторинга.

4.8. Биржа изменила статус символа

Актор: Система (фоновая задача)

Предусловия:

- Символ существует в базе данных с определенным статусом
- Биржа возвращает новый статус для данного символа

Основной поток:

- 1. Система обнаруживает изменение статуса символа
- 2. Система обновляет статус символа в базе данных
- 3. Система формирует и отправляет уведомление через Telegram (если функция активирована)

Результат: Статус символа актуализируется в соответствии с данными биржи.

4.9. Символ снова появился после Delisted → создаётся заново

Актор: Система (фоновая задача)

Предусловия:

- Символ ранее имел статус Delisted
- Символ снова появляется в ответе биржи

Основной поток:

- 1. Система реактивирует существующую запись, обновляя статус символа
- 2. Системя присваивает символу статус, полученный от биржи

Результат: Символ возвращается в процесс активного мониторинга.

5. Управление настройками уведомлений

5.1. Администратор просматривает настройки уведомлений

Актор: Администратор

Предусловия:

• Система функционирует в рабочем режиме

Основной поток:

- 1. Администратор выполняет HTTP GET запрос на /api/v1/notification-settings
- 2. Система возвращает текущие настройки уведомлений в формате:

```
"isTelegramEnabled": true|false,
"isEmailEnabled": true|false,
"isTelegramConfigured": true|false,
"isEmailConfigured": true|false
}
```

Результат:

- Администратор получает информацию о текущем состоянии уведомлений:
 - Включены ли Telegram-уведомления
 - Включены ли стандартные email-уведомления
 - Корректно ли сконфигурированы Telegram-уведомления
 - Корректно ли сконфигурированы email-уведомления

5.2. Администратор изменяет настройки уведомлений

Актор: Администратор

Предусловия:

- Система функционирует в рабочем режиме
- Уведомления, которые администратор планирует включить, корректно сконфигурированы

Основной поток:

1. Администратор выполняет HTTP PUT запрос на /api/v1/notification-settings

```
PUT /api/v1/notification-settings

Parameters

No parameters

Request body

{
    "isTelegramEnabled": false,
    "isEmailEnabled": true
}
```

- 2. Система проверяет корректность запроса
- 3. Система обновляет настройки уведомлений
- 4. Система возвращает статус 200 ОК с обновленными настройками

```
Response body

{
    "isTelegramEnabled": false,
    "isEmailEnabled": true,
    "isTelegramConfigured": true,
    "isEmailConfigured": true
}
```

Результат:

- Настройки уведомлений обновлены согласно запросу администратора
- Система будет отправлять уведомления в соответствии с новыми настройками

5.3. Администратор пытается включить несконфигурированные уведомления

Актор: Администратор

Предусловия:

• Система функционирует в рабочем режиме

• Один или оба типа уведомлений (Telegram, email) не сконфигурированы

Основной поток:

- 1. Администратор выполняет HTTP PUT запрос на /api/v1/notification-settings, пытаясь включить несконфигурированный тип уведомлений
- 2. Система определяет, что запрашиваемый тип уведомлений не сконфигурирован
- 3. Система возвращает статус 400 Bad Request с сообщением об ошибке

```
### Response body

{

| "type": "https://tools.ietf.org/html/rfc7231#section-6.5.1",
| "title": "One or more validation errors occurred.",
| "status": 400,
| "detail": "See the errors field for details.",
| "instance": "/api/v1/notification-settings",
| "errors": {
| "IsEmailEnabled": [
| "Cannot enable Email notifications because they are not properly configured"
| ]
| }
| }
```

Альтернативный поток:

- Если администратор пытается отключить уведомления (независимо от их конфигурации):
 - 1. Система обновляет настройки, отключая указанные типы уведомлений
 - 2. Система возвращает статус 200 ОК с обновленными настройками

Результат:

- Уведомления не могут быть включены без корректной конфигурации
- Администратор получает сообщение об ошибке и инструкции по настройке соответствующего типа уведомлений

5.4. Отправка критических уведомлений

Актор: Система

Предусловия:

• Возникла критическая ситуация, требующая уведомления администратора

- 1. Система определяет необходимость отправки критического уведомления
- 2. Система проверяет конфигурацию критических email-уведомлений

- 3. Если критические уведомления сконфигурированы:
 - Система формирует критическое уведомление
 - Система отправляет уведомление на указанный email-адрес

System Error

Component: component

Time: 2025-04-11 09:41:21 UTC

Error Message: Unexpected error occurred

Exception Details:

Type: System.Exception

Message: gjajgrpgn epng apreng ag

Stack Trace:

Please investigate the system error as soon as possible.

- 4. Если критические уведомления не сконфигурированы:
 - Система регистрирует неудачную попытку отправки уведомления в лог

Результат:

- Критическое уведомление отправлено администратору (если сконфигурировано)
- Критические уведомления не могут быть отключены администратором, кроме как через изменение конфигурации системы

5.5. Отправка стандартных уведомлений

Актор: Система

Предусловия:

- Произошли изменения, требующие уведомления администратора
- Система функционирует в рабочем режиме

- 1. Система определяет необходимость отправки стандартного уведомления
- 2. Система проверяет настройки уведомлений:

- Если Telegram-уведомления включены и сконфигурированы:
 - Система формирует и отправляет Telegram-уведомление

Symbol changes on BinanceCoinFutures (CoinFutures)

2025-04-24 13:27:00 UTC

New Symbols (65)

- BTCUSD PERP (CoinFutures)

Base: BTC, Quote: USD

Status: Active

BTCUSD_250627 (CoinFutures)

Base: BTC, Quote: USD

Status: Active

BTCUSD_250926 (CoinFutures)

Base: BTC, Quote: USD

Status: Active

- ETHUSD PERP (CoinFutures)

Base: ETH, Quote: USD

Status: Active

- ETHUSD 250627 (CoinFutures)

Base: ETH, Quote: USD

Status: Active

...and 60 more symbols

Total changes: 65

4:27 PI

- Если email-уведомления включены и сконфигурированы:
 - Система формирует и отправляет email-уведомление

Symbol changes on BinanceSpot (Spot)

2025-04-24 11:08:03 UTC

New Symbols (3075)

ETHBTC (Spot) Base: ETH, Quote: BTC Status: Active LTCBTC (Spot) Base: LTC, Quote: BTC Status: Active BNBBTC (Spot) Base: BNB , Quote: BTC Status: Active NEOBTC (Spot) Base: NEO, Quote: BTC Status: Active QTUMETH (Spot) Base: QTUM, Quote: ETH

Status: Active

...and 3070 more symbols

Total changes: 3075

- 3. Если все типы уведомлений отключены или не сконфигурированы:
 - Уведомления не отправляются, информация только фиксируется в логе системы

Альтернативный поток:

- Если происходит ошибка при отправке уведомления:
 - 1. Система регистрирует неудачную попытку в лог
 - 2. Система продолжает выполнение операции

Результат:

- Стандартные уведомления отправляются через настроенные и активированные каналы связи
- Администратор получает информацию об изменениях в системе

 В случае отключения всех уведомлений, изменения фиксируются только в логе системы

6. Инициализация и восстановление системы

6.1. Первый запуск системы

Актор: Система

Предусловия:

- Первоначальная установка системы
- База данных не инициализирована

Основной поток:

- 1. Система инициализирует схему базы данных
- 2. Система автоматически добавляет поддерживаемые биржи: BinanceSpot, BinanceUsdtFutures и BinanceCoinFutures
- 3. Система проверяет конфигурацию для стандартных уведомлений (Telegram, email) и критических уведомлений (email)
- 4. При обнаружении некорректной конфигурации, система регистрирует соответствующие предупреждения в лог

infosymbolserver-1 | [13:40:18 WRN] Email notification service is not properly configured. Please provide valid SMTP server settings and at least one recipient in the configuration. Email notifications will be disabled until properly configured.

Система начинает функционировать в стандартном режиме

Альтернативный поток:

- При невозможности подключения к базе данных:
 - 1. Система регистрирует критическую ошибку в лог
 - 2. Процесс инициализации прерывается

Результат:

- Система успешно инициализирована с базовой конфигурацией
- Администратор информирован о потенциальных проблемах с конфигурацией через логи системы
- Система готова к выполнению задач синхронизации

6.2. Восстановление после сбоя

Актор: Система/Администратор

Предусловия:

- Система была ранее инициализирована и настроена
- Произошел сбой в работе системы

Основной поток:

- 1. Администратор инициирует перезапуск системы
- 2. Система восстанавливает подключение к базе данных
- 3. Система выполняет проверку работоспособности основных компонентов
- 4. При следующем запланированном выполнении фоновой задачи, система актуализирует информацию о символах
- 5. Система возобновляет стандартное функционирование

Альтернативный поток:

- При невозможности подключения к базе данных:
 - 1. Система регистрирует критическую ошибку в лог
 - 2. Процесс восстановления прерывается

Примечание:

В текущей версии, если система прекратила работу после обновления информации о символах, но до отправки уведомлений, при перезапуске эти уведомления не будут повторно отправлены. Данная функциональность будет реализована в следующих версиях.

Результат:

- Система возобновляет работу в стандартном режиме
- Информация о символах актуализируется при следующем выполнении фоновой задачи
- Система продолжает выполнение запланированных задач