?????????? ? ?????????????? NDR API 7.1.x - 8.0

????? ???????? ???????? ????????? REST API

Сервер REST API, который обеспечивает доступ внешних систем к функциональности NDR, функционирует на сервере с компонентом Central Node и обрабатывает запросы с использованием архитектурного стиля взаимодействия REST (Representational State Transfer). Обращения к серверу REST API выполняются по протоколу HTTPS. Вы можете настроить параметры сервера REST API в разделе **Параметры → Серверы подключений** (в том числе заменить используемый по умолчанию самоподписанный сертификат на доверенный).

Для представления данных в запросах и ответах используется формат JSON.

Внешние системы, использующие Kaspersky Anti Targeted Attack Platform API, подключаются к компоненту Central Node через коннекторы. Коннекторы обеспечивают безопасное соединение с использованием сертификатов. Для каждой внешней системы, из которой вы хотите отправлять запросы на сервер REST API, вам нужно создать отдельный коннектор в Kaspersky Anti Targeted Attack Platform.

Для соединения с Kaspersky Anti Targeted Attack Platform внешняя система должна использовать токен аутентификации. Kaspersky Anti Targeted Attack Platform выдает токен аутентификации по запросу внешней системы и использует для токена сертификаты коннектора, который был создан для этой системы. **Время действия токена аутентификации составляет 10 часов.** Внешняя система может обновить токен аутентификации по специальному запросу.

В Kaspersky Anti Targeted Attack Platform API предусмотрены следующие способы работы со внешними системами:

- взаимодействие на основе архитектурного стиля REST;
- взаимодействие по протоколу WebSocket.

Внешние системы могут использовать способ взаимодействия по протоколу WebSocket в Kaspersky Anti Targeted Attack Platform API для создания подписок на изменяемые значения, которые получает приложение.

помощью Kaspersky Anti Targeted Attack Platform API внешние системы могут выполнять следующие действия:

- получать данные об известных приложению устройствах;

- добавлять, изменять и удалять устройства;
- получать данные о зарегистрированных событиях в трафике сети (событиях NDR);
- отправлять события NDR в Kaspersky Anti Targeted Attack Platform (для регистрации используется системный тип события с кодом 4000005400);
- получать данные об обнаруженных уязвимостях;
- получать сообщения приложения и записи аудита;
- получать данные о разрешающих правилах;
- включать, выключать и удалять разрешающие правила;
- получать данные о рисках, связанных с устройствами;
- получать данные об адресных пространствах;
- отправлять отчет о топологической карте сети в Kaspersky Anti Targeted Attack Platform;
- отправлять, получать и удалять сведения о пользователях на устройствах;
- отправлять и получать сведения о приложениях и патчах на устройствах;
- отправлять и удалять сведения об исполняемых файлах на устройствах;
- отправлять содержимое журналов устройств;
- получать следующие данные о приложении:
  - список серверов с компонентами приложения;
  - список точек мониторинга и их параметры;
  - список поддерживаемых стеков протоколов и их параметры;
  - список типов событий NDR и их параметры;
  - текущее состояние и режимы работы технологий;
  - версия приложения и даты выпуска установленных обновлений;
  - информация о добавленном лицензионном ключе;
  - язык локализации приложения.

Полная коллекция API запросов к NDR через POSTMAN:

KATA_API_v4.postman_collection.json

??????????? ??????????? ????????????? ??? ????????????? Kaspersky Anti Targeted Attack Platform API

Внешние системы получают доступ к функциям приложения с использованием Kaspersky Anti Targeted Attack Platform API, устанавливая зашифрованные соединения по протоколу HTTPS. Для обеспечения безопасности соединений используются сертификаты, выданные компонентом Central Node Kaspersky Anti Targeted Attack Platform. Компонент выдает сертификаты для коннекторов, через которые подключаются внешние системы.

Для каждой внешней системы в Kaspersky Anti Targeted Attack Platform должен быть создан отдельный коннектор. Подключение через коннектор возможно с использованием только того сертификата, который был выдан компонентом Central Node и сохранен в файле свертки для этого коннектора. Подключение невозможно установить, если внешняя система предъявляет сертификат от другого коннектора, другого компонента Central Node Kaspersky Anti Targeted Attack Platform, или сертификат, используемый для других подключений (например, сертификат компонента Sensor).

После установки зашифрованного соединения внешняя система должна запросить *токен аутентификации* для коннектора, который будет указываться внешней системой в запросах к Central Node REST API.

Токен аутентификации действителен в течение 10 часов после выдачи. При необходимости дальнейшего использования токена внешняя система должна запросить продление времени его действия до наступления момента прекращения действия.

При обработке запросов от внешних систем Kaspersky Anti Targeted Attack Platform сохраняет в журнале аудита сведения о попытках выполнения следующих операций:

- получение токена аутентификации;
- продление времени действия для токена аутентификации;
- добавление устройства в таблицу устройств;
- изменение сведений об устройстве;
- удаление устройства;
- запрос журнала аудита (при первом чтении записей аудита через коннектор после загрузки веб-сервера).

???????? ? ????????????? ??????????? ??? Kaspersky Anti Targeted Attack Platform API

Для взаимодействия внешней системы с Kaspersky Anti Targeted Attack Platform API вам нужно добавить коннектор для этой системы. При создании коннектора для него требуется указать системный тип Generic.

При добавлении коннектора, а также при создании нового файла свертки для этого коннектора Central Node формирует файл свертки, который вам нужно использовать для работы коннектора.

Файл свертки представляет собой архив, содержащий следующие файлы:

- certificates.pfx – содержит в зашифрованном виде открытый ключ сертификата Central Node, а также сертификат, выданный Central Node для коннектора (с закрытым ключом). Содержимое файла зашифровано с использованием пароля, который был указан при добавлении коннектора или при создании нового файла свертки для этого коннектора.
- metadata.json – содержит конфигурационные данные для коннектора. Данные представлены в формате JSON.

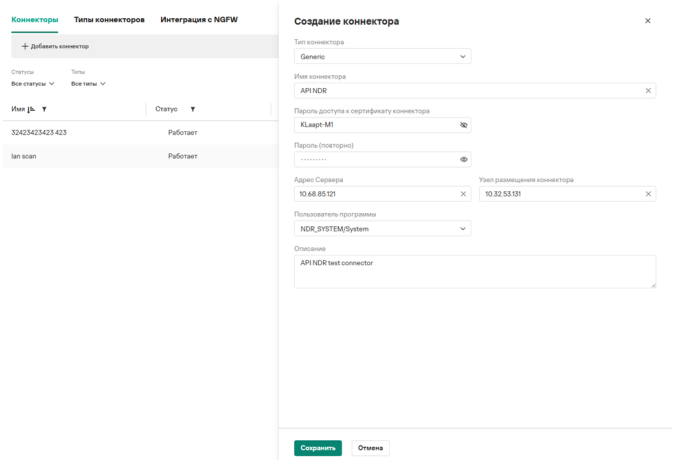
Перечисленные файлы вам нужно использовать для подключения внешней системы через коннектор. Для расшифровки файла certificates.pfx и применения содержащегося в нем сертификата с ключами вы можете использовать стандартные методы обработки файлов этого формата (например, команды `openssl`). Адреса, указанные в файле metadata.json, требуются для работы коннектора и отправки запросов к серверу REST API.

Сертификат и конфигурационные данные в файле свертки действительны до тех пор, пока не создан новый файл свертки или пока не удален коннектор в приложении.

?????????? ??????????

Чтобы добавить коннектор:

1. Войдите в веб-интерфейс под учетной записью администратора приложения.
2. Выберите раздел Параметры, подраздел Коннекторы.
3. На вкладке Коннекторы откройте область деталей по кнопке Добавить коннектор.
4. Настройте общие параметры коннектора:
   1. Выберите нужный тип коннектора и введите имя коннектора.
   2. Если вы хотите добавить неуправляемый коннектор (или коннектор с включенным режимом игнорирования функций управляемого коннектора), введите пароль для доступа к сертификату коннектора.
      С использованием заданного пароля будет зашифрован сертификат в файле свертки коннектора.
   3. Укажите адрес сервера Central Node. По указанному адресу коннектор будет подключаться к Central Node.
   4. Укажите узел размещения коннектора. Чтобы добавить неуправляемый коннектор, вам нужно ввести IP-адрес компьютера, на котором будут работать программные модули коннектора.
   5. Выберите пользователя, под которым сторонняя система будет подключаться к приложению через коннектор. Требуется указать имя одного из пользователей приложения. (ndr/system).
   6. Нажмите на кнопку **Сохранить**. Новый коннектор появится в таблице коннекторов.



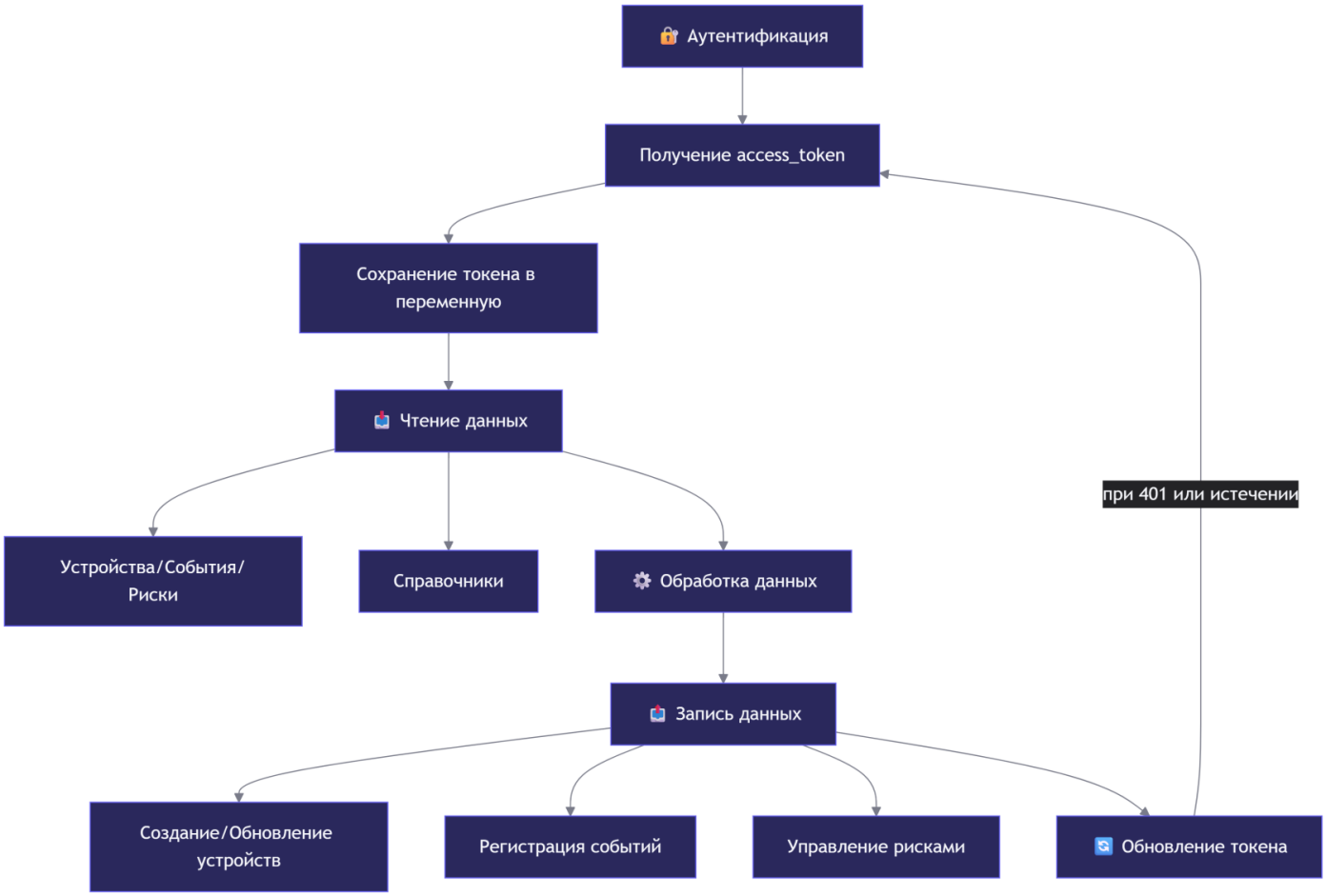
Если добавлен неуправляемый коннектор, Central Node сформирует файл свертки для нового коннектора, после чего браузер сохранит загруженный файл. В зависимости от параметров, заданных в браузере, на экране может появиться окно для изменения пути и имени сохраняемого файла. Содержимое полученного файла свертки вам нужно загрузить в приложение, которое будет использовать коннектор.

Неуправляемый коннектор не предоставляет функции управляемого коннектора. Регистрацию такого коннектора, а также запуск, остановку и удаление его программных модулей требуется выполнять вручную на узле размещения коннектора. При включении и выключении неуправляемого коннектора приложение соответственно разрешает и запрещает взаимодействие с коннектором на стороне сервера Central Node.

Безопасность соединений коннекторов с сервером Central Node приложения обеспечивается с использованием сертификатов. Сертификаты для коннекторов создаются при добавлении коннекторов в приложение. Для программных модулей управляемых коннекторов приложение автоматически передает созданные сертификаты. При добавлении неуправляемого коннектора (или при добавлении управляемого коннектора с включенным режимом игнорирования функций управляемого коннектора) сертификат для программных модулей этого коннектора требуется загрузить вручную с помощью файла свертки. При необходимости заменить (выпустить новый) сертификат для такого коннектора вам нужно создать новый файл свертки и использовать этот файл для загрузки нового сертификата. Замена сертификатов управляемых коннекторов возможна только путем удаления и повторного добавления этих коннекторов.

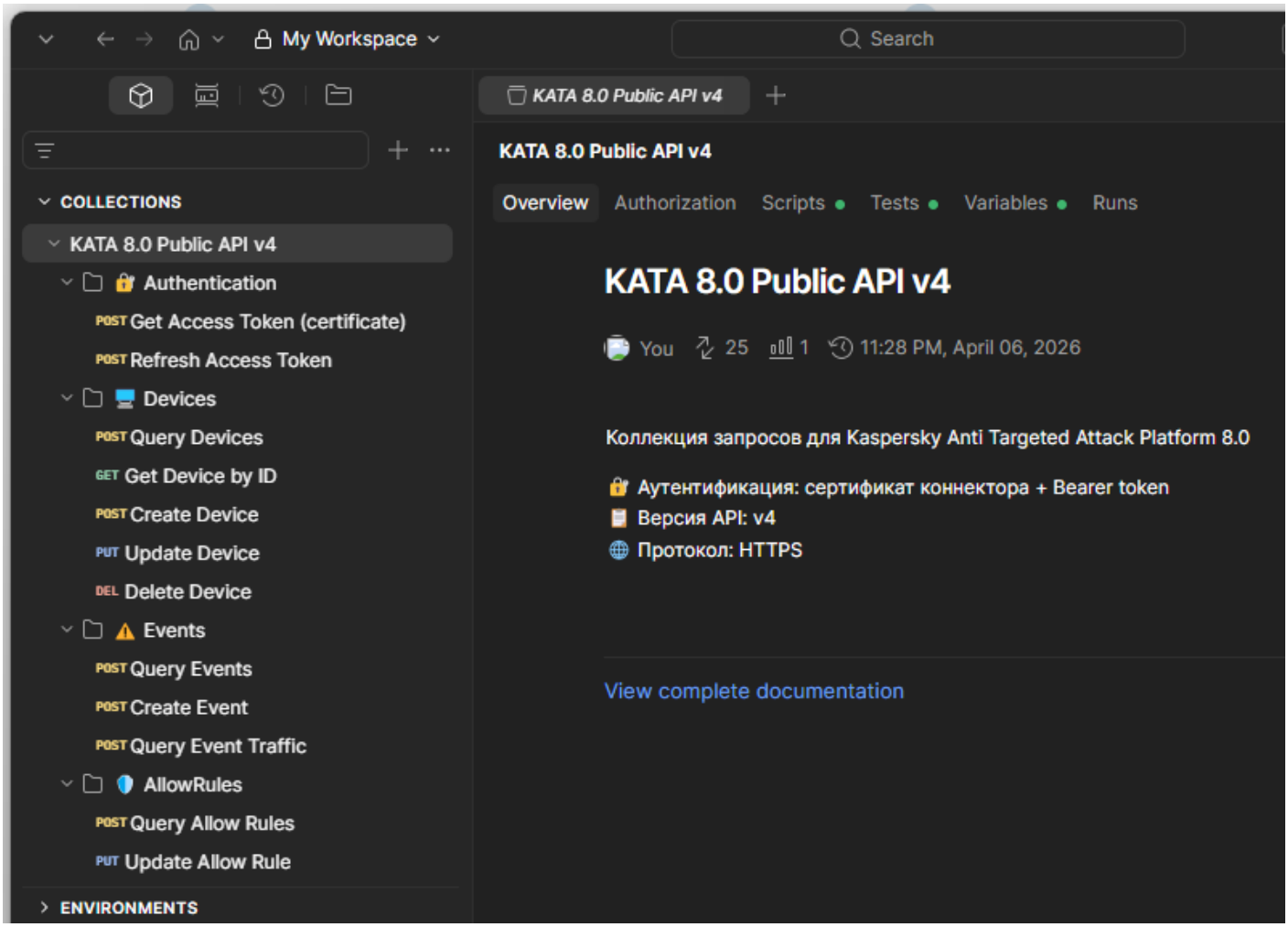
Пошаговая настройка Postman

??????? ??????????? ????????



?????? ?????????

1. **Загрузите полную коллекцию API запросов к NDR через POSTMAN:**
   KATA_API_v4.postman_collection.json
2. Откройте **Postman**
3. Нажмите **Import** (левый верхний угол) → **Upload Files**
4. Выберите файл `KATA_API_v4.postman_collection.json`
5. Коллекция появится в левой панели



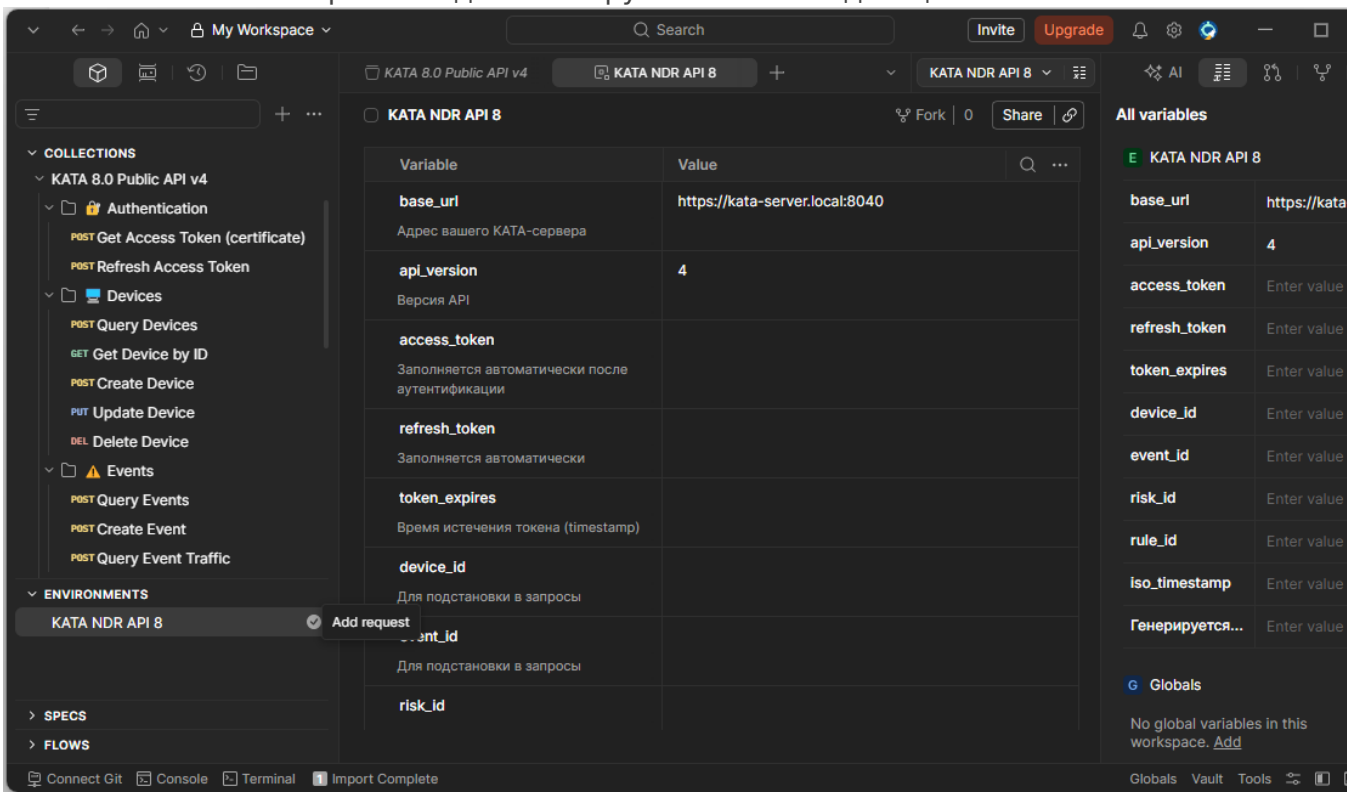
???????? ? ????????? ????????? (Environment)

1. В правом верхнем углу нажмите на выпадающий список окружений → **Manage Environments**
2. Нажмите **Add**
3. Заполните:

Variable	Initial Value	Current Value	Описание

base_url	https://kata-server.local	https://kata-server.local	Адрес вашего KATA-сервера
api_version	4	4	Версия API
access_token	(пусто)	(пусто)	Заполняется автоматически после аутентификации
refresh_token	(пусто)	(пусто)	Заполняется автоматически
token_expires	(пусто)	(пусто)	Время истечения токена (timestamp)
device_id	(пусто)	(пусто)	Для подстановки в запросы
event_id	(пусто)	(пусто)	Для подстановки в запросы
risk_id	(пусто)	(пусто)	Для подстановки в запросы
rule_id	(пусто)	(пусто)	Для подстановки в запросы
iso_timestamp	(пусто)	(пусто)	Генерируется автоматически

4. Нажмите **Add** → Выберите созданное окружение в выпадающем списке

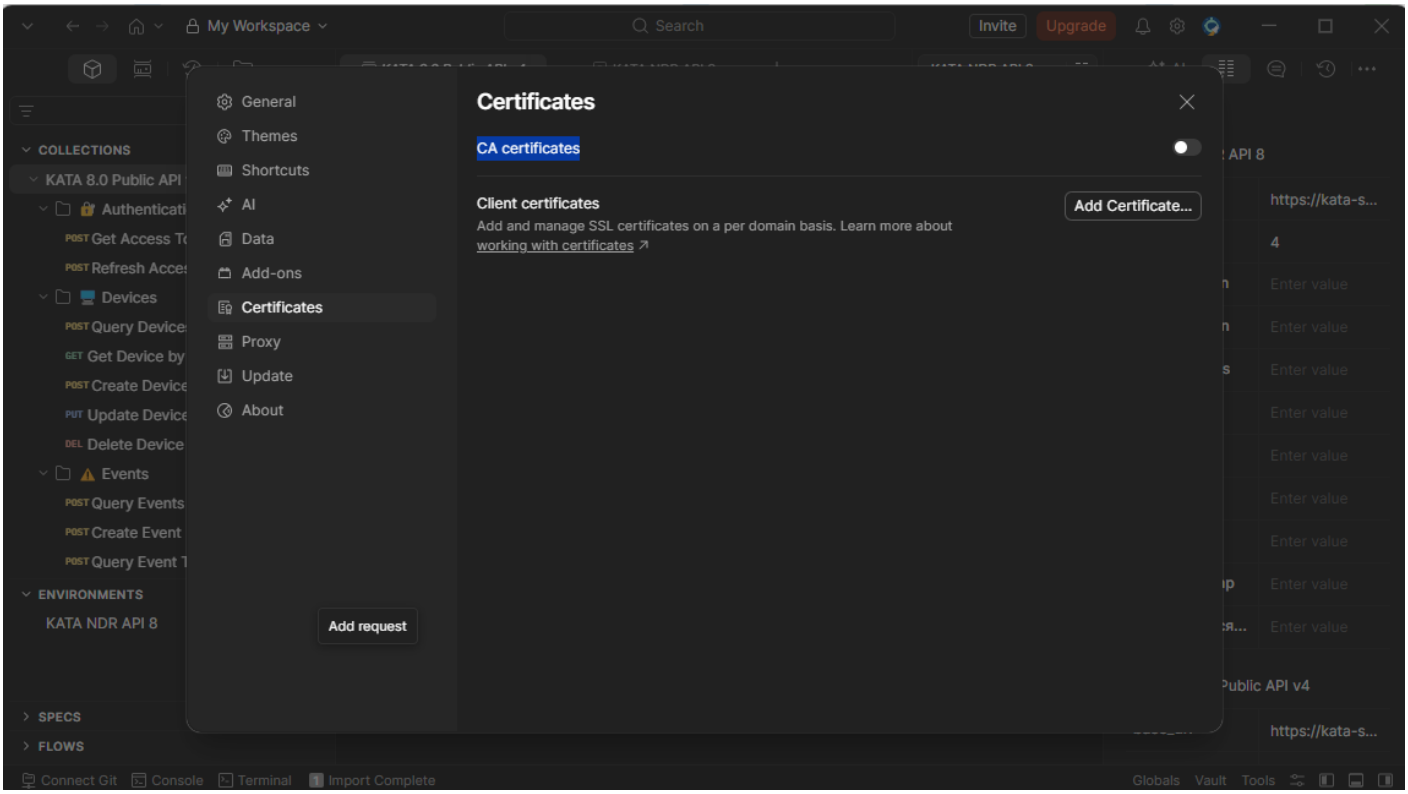


????????? ?????????? ???????????? (????? !)

KATA API требует **двусторонней TLS-аутентификации** (mTLS).

??????? A: ?????????? ????????? ??????????? (????????????)

1. **File → Settings → Certificates**
2. Перейдите на вкладку **Certificates**

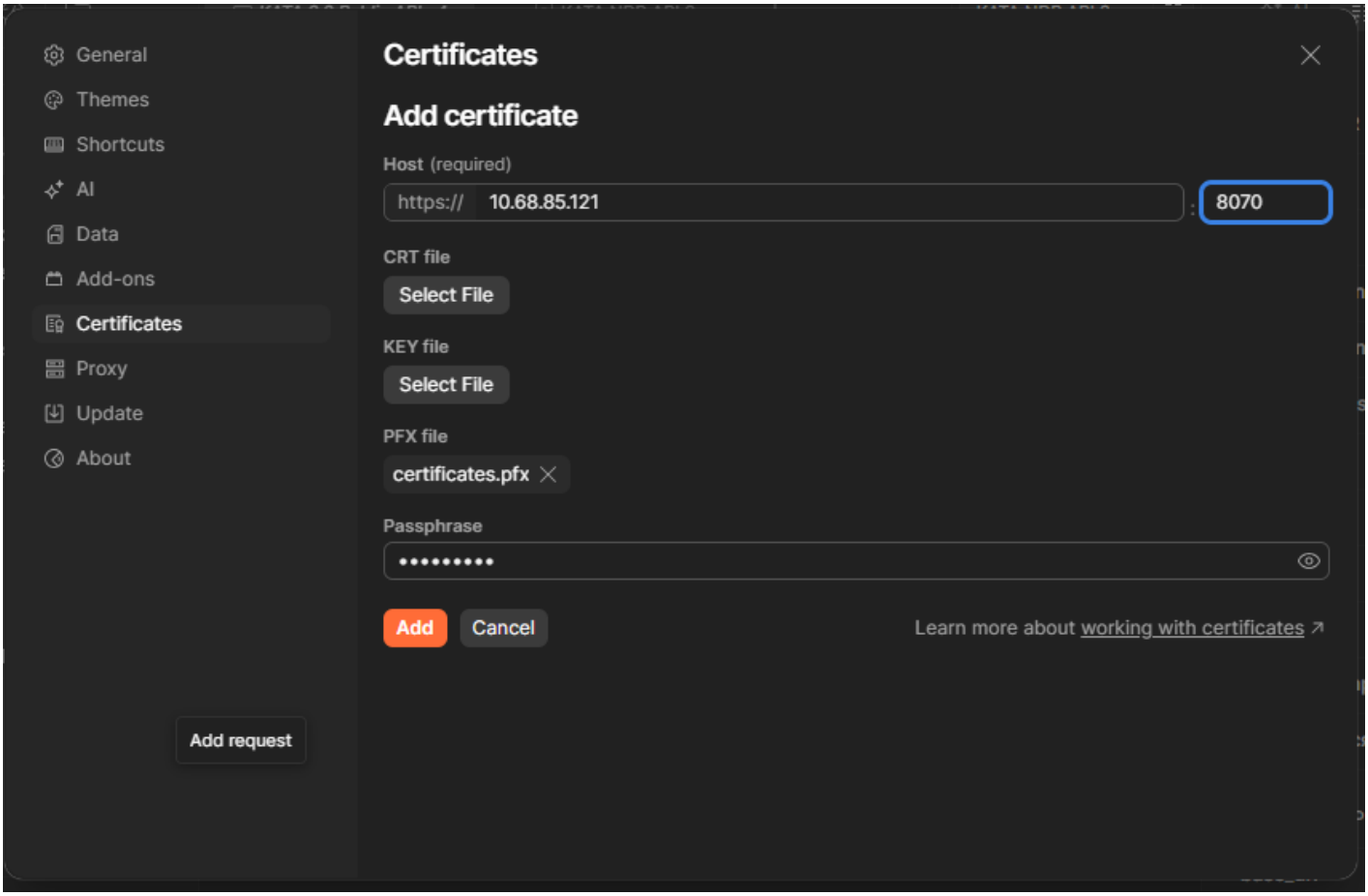


1. Нажмите **Add Certificate**

Заполните:

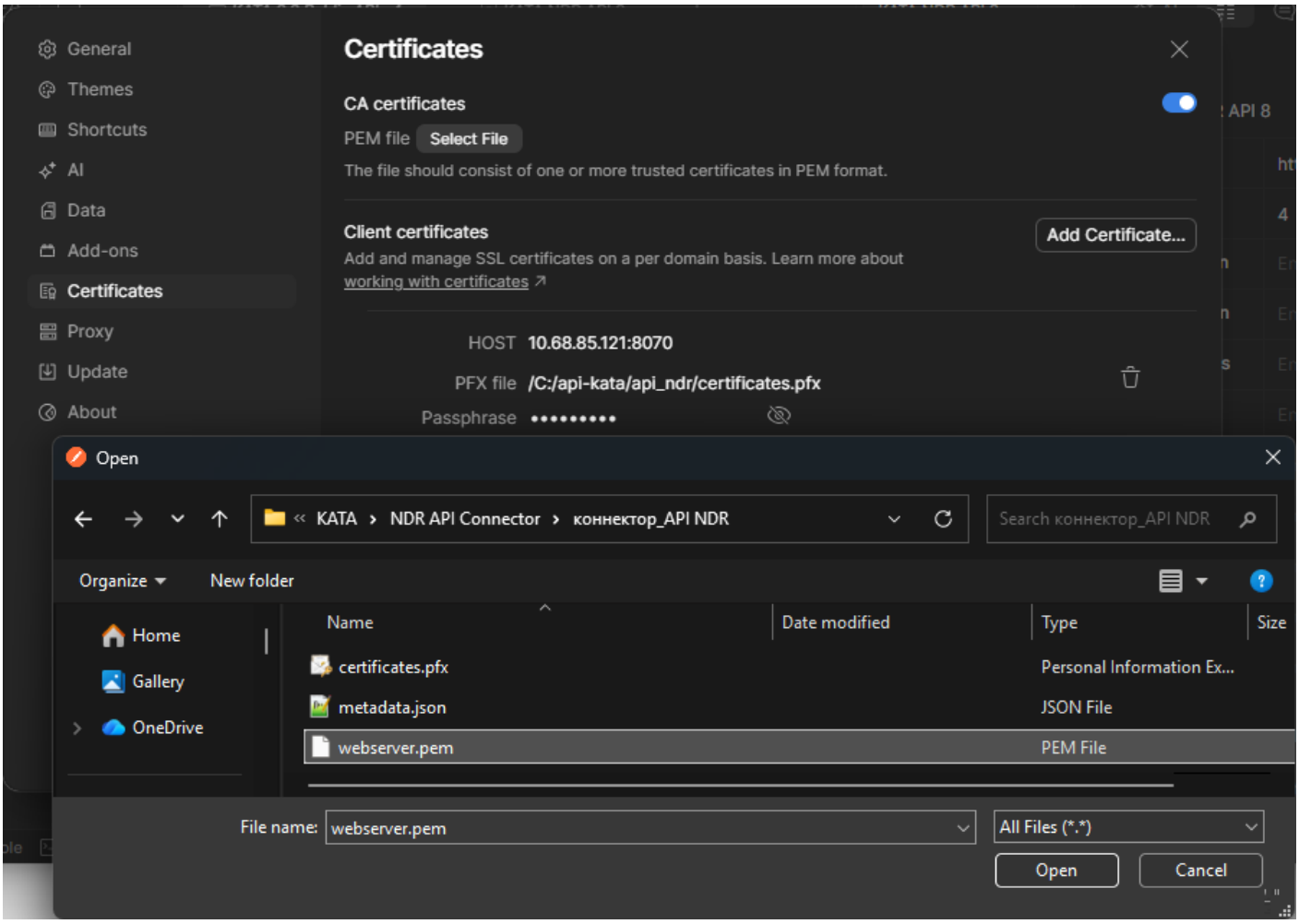
Поле	Значение	Пример
Host	Адрес сервера без протокола	kata-server.local:443 или *.kata.local
CRT file	Путь к клиентскому сертификату	C:\certs\connector.crt
Key file	Путь к приватному ключу	C:\certs\connector.key

PFX file	*(оставьте пустым, если используете CRT+KEY) либо добавьте выгруженный из файла свертки сертификат*	
Password	*(если ключ запаролен)*	your_password

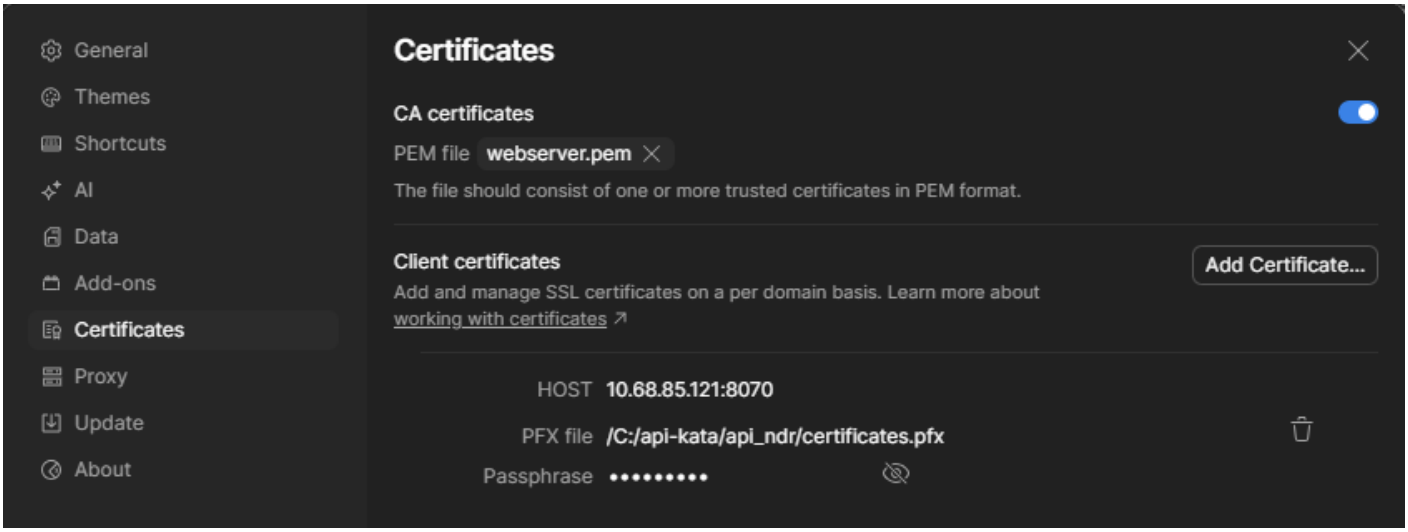


CA Certificate (для проверки сервера):

- **Settings → Certificates → CA Certificates**
- Включите **Use custom CA certificate**
- Укажите путь к корневому сертификату: C:\certs\ca.crt или сертификату kata.crt или сертификату из файла свертки webserver.pem



Результат:



??????? B: ????????? ? ?????????? ???????

1. Откройте запрос **Get Access Token (certificate)**
2. Перейдите на вкладку **Settings** (рядом с Authorization)
3. В разделе **Client certificate** укажите:
   - **Cert**: путь к connector.crt

- **Key**: путь к connector.key

Сертификаты коннектора создаются в KATA Console: **Параметры → Серверы подключений → Коннекторы → Создать → Скачать сертификат**

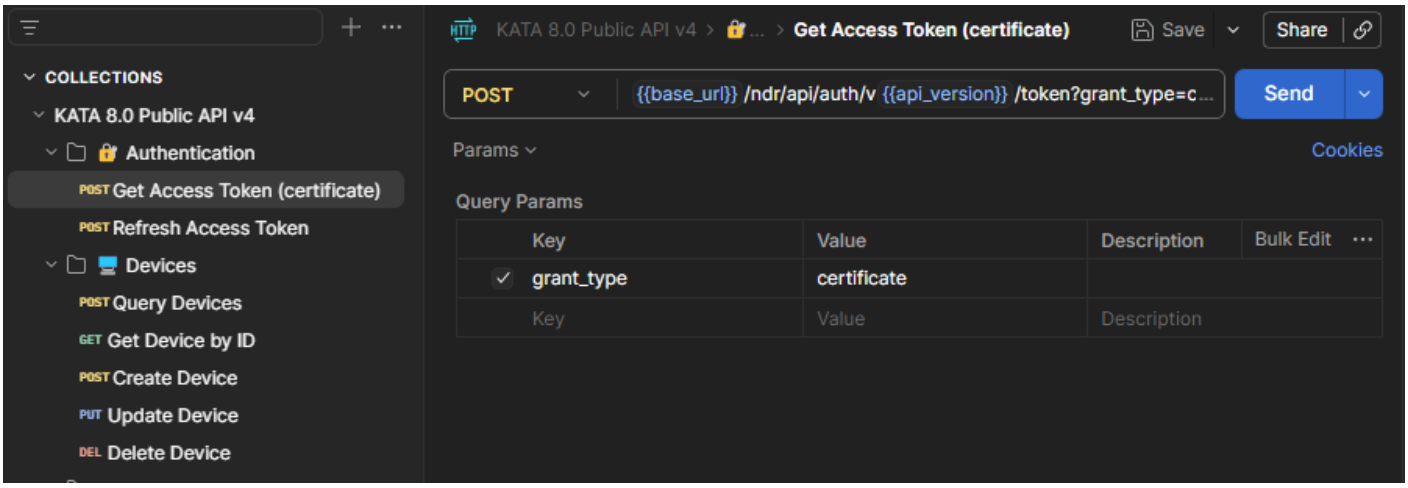
????????? SSL Verification

Если используете самоподписанные сертификаты:

1. **File → Settings → General**
2. **Отключите** опцию **SSL certificate verification** (только для тестов!)
3. Для продакшена: обязательно настройте **CA Certificate** как в Шаге 3

????????? ?????? (?????? ??????)

1. Выберите окружение в правом верхнем углу
2. В коллекции раскройте **Authentication**
3. Запустите запрос **Get Access Token (certificate)**
4. При успешном ответе:
   - access_token автоматически сохранится в окружение
   - В консоль (Postman Console: **View → Show Postman Console**) выведется время истечения



????????????? ????????

Выберите любой запрос, например: **Devices → Query Devices**

1. Убедитесь, что в заголовках есть:
   Authorization: Bearer {{access_token}}
   Content-Type: application/json
2. Нажмите **Send**

3. Проверьте ответ во вкладке **Body → JSON**

?????????? ?????? (????????????? ??? ???????)

????????????? (????? ??????):

- Коллекция содержит проверку в **Pre-request Script**
- Если токен истекает < 5 минут — в консоль выводится предупреждение

???????:

1. Запустите запрос **Refresh Access Token**
2. Или повторно выполните **Get Access Token (certificate)**

Revision #11
Created 29 December 2025 09:27:08 by Владислав
Updated 6 April 2026 22:44:22 by Владислав