

Принцип взаимодействия через API: http/https RESTFull
Сериализация: JSON

Данный документ описывает URN сервиса и способы взаимодействия с ресурсами. Для получения URL сервиса просьба обращаться отдельно.

Синтаксис описания URN:

- { } – Обязательные параметры
- [] – Не обязательные параметры

Для доступа к методам API после прохождения аутентификации выдаётся временный токен доступа. Токен доступа необходимо передавать вместе с запросом к методам одним из следующих способов (перечисляются в порядке поиска):

- 1 в URN – только если это указано в описании метода
- 2 в параметре URN с именем accessToken
- 3 в Header с именем Authorization и значением: Bearer {token}
- 4 в Cookies с именем Access-Token и значением: {token}

Если токен передан одновременно несколькими способами, то используется только тот который был найден первым, остальные игнорируются)

Терминология

API - Application Programming Interface (интерфейс программирования приложений, интерфейс прикладного программирования). Набор готовых классов, процедур, функций, структур, констант или ресурсов, предоставляемых приложением (библиотекой, сервисом) для использования во внешних программных продуктах. В контексте данного документа API описывает ресурсы сервиса, методы и способы взаимодействия с ресурсами и наборы входящих и выходящих данных требуемых для обращения к ресурсу.

URI – Uniform Resource Identifier (унифицированный идентификатор ресурса). Обозначает протокол, имя, порт, адрес ресурса в сети. Фактически является суммой Url плюс Urn:

URI = URL + URN

Пример: **https://www.example.com/path/to/resource.html**

URL – Uniform Resource Locator (унифицированный определитель местонахождения ресурса). Описывает схему(протокол), логин, пароль, хост(доменное имя или ip адрес), порт сервера на котором может находится ресурс, описывает только сервер, без ресурса.

Пример: **https://www.example.com**

URN – Uniform Resource Name (унифицированное имя ресурса). Описывает ресурс, без указания на каком URL находится ресурс. В контексте документа мы не будем придерживаться описанной в RFC модели URN ($URN = [urn:] + NID + ':' + NSS$), так как она сильно усложнит документ и его восприятие.

В документе качестве URN названы и описаны ресурсы являющиеся частью URI

Пример: **/path/to/resource.html**

Возвращаемые результаты и коды ошибок

[оглавление](#)
[описание](#)
[шаблоны и html создаваемый сервером](#)

При запросе методов API всегда возвращается HTTP code, в зависимости от HTTP кода ответа сервера необходимо по разному разбирать переданные данные.
Ниже представлена таблица описывающая зависимость возвращаемых данных от HTTP кода.

HTTP CODE от	HTTP CODE до	Описание
200	299	API ответит объектом описанным для каждого метода в документе ниже
300	399	Редирект, поведение описывается в каждом методе отдельно
400	999	API ответит объектом error

Объект error

Имя	Тип	Описание				
error	Объект	Объект error состоящий из трёх полей описанных ниже				
		Имя	Тип	Описание		
		code	Число	Уникальный код ошибки из справочника ошибок		
		message	Строка	Описание ошибки в локализации текущей сессии, сообщение готовое для вывода в интерфейсе пользователя		
		errors	Массив	Массив объектов с описанием имён полей и ошибок в них		
				Имя	Тип	Описание
				field	Строка	Название поля
				fieldValue	Строка	Переданное значение поля field
				message	Строка	Описание ошибки

Пример:
{error: {"errors":null,"code":1,"message":"Ошибка авторизации"}}

Коды HTTP ответов сервера

HTTP CODE	Описание
Информационные коды сервера	
100	Continue. Частично переданные на сервер данные проверены и сервер удовлетворён начальными данными. Клиент может продолжать отправку заголовков и данных
101	Switching Protocols. Сервер предлагает сменить протокол, перейти на более подходящий для указанного ресурса протокол. Список предлагаемых протоколов передаётся в заголовке Update Если клиент готов сменить протокол то новый запрос ожидается с указанием другого протокола
102	Processing (WebDAV). Запрос принят, на обработку запроса потребуется длительное время. Клиент, при получении такого ответа, должен сбросить таймер ожидания и ожидать следующего ответа в обычном режиме
105	Name Not Resolved. Ошибка при разрешении доменного имени, в связи с не верным или отсутствующих ip адресом DNS сервера

Коды информирующие об успешной обработки запроса

200	Ok. Успешный запрос. Результат запроса передаётся в заголовке или теле ответа
201	Created. Успешный запрос, в результате которого был создан новый ресурс. В ответе возвращается заголовок Location с указанием созданного ресурса. Так же возвращаются характеристики нового ресурса, Content-Type .
202	Accepted. Запрос принят на обработку, но обработка не завершена. Клиенту нет необходимости ожидать окончания обработки запроса, так как процесс может быть весьма длительным
203	Non-Authoritative Information. Успешный запрос, но передаваемая в ответе информация взята из кэша или не является гарантированно достоверной (могла устареть)
204	No Content. Успешный запрос, в ответе передаются только заголовки без тела сообщения. Клиент не должен обновлять тело документа/ресурса, но может применить к нему полученные данные
205	Reset Content. Сервер обязывает клиента сбросить введённые пользователем данные. Тела сообщения сервер клиенту не передаёт. Документ на стороне клиента обновлять не обязательно
206	Partial Content. Сервер удачно выполнил частичный GET запрос и возвращает только часть данных В заголовке Content-Range сервер указывает байтовые диапазоны содержимого
207	Multi-Status (WebDAV). Сервер возвращает результат выполнения сразу нескольких независимых операций. Результат ожидается в виде XML тела ответа с объектом multistatus
208	Already Reported (WebDAV). Код возвращается в случае запроса к ресурсу состоящему из коллекций данных повторяющихся между собой и фактически является указателем на то что данные ресурса А можно взять из ресурса Б так как они идентичны, либо данные ответа текущего запроса с итерацией совпадают с ответом предыдущих запросов с итерацией (ссылка). Так же сервер возвращает заголовок с ссылкой на данные на которые ссылается
226	IM Used. Заголовок A-IM от клиента был успешно принят и обработан. Сервер возвращает содержимое с учётом указанных параметров

Коды информирующие клиента о том что, для успешного выполнения запроса необходимо выполнить другой запрос

300	Multiple Choices. Существует несколько вариантов предоставления ресурса по типу MIME Сервер возвращает список альтернатив для выбора на стороне клиента
301	Moved Permanently. Запрошенный ресурс был окончательно перенесён на новый URI Новый URI возвращается в заголовке Location
302	Found. Moved Temporarily. Запрошенный ресурс временно доступен по другому URI возвращаемому в заголовке Location
303	See Other. Ресурс по запрашиваемому URI необходимо запросить по адресу передаваемому в поле Location и только методом GET несмотря на первоначальный запрос иным методом
304	Not Modified. Сервер отвечает кодом 304 если ресурс был запрошен методом GET с использованием заголовков If-Modified-Since или If-None-Match и документ не изменился с указанного момента. П этом ответ сервера не содержит тело ресурса
305	Use Proxy. Запрос к запрашиваемому ресурсу должен осуществляться через прокси URI которого указывается в заголовки ответа в Location

307	Temporary Redirect. Запрошенный ресурс на короткое время доступен по другому URI указанному в заголовке ответа Location
308	Permanent Redirect (experiemental). Текущий запрос и все будущие запросы необходимо выполнить по другому URI ресурса. Новый адрес ресурса возвращается в заголовке Location ответа сервера
Коды информирующих клиента об ошибках со стороны клиента при отправке запроса	
На любой запрос сервер всегда возвращает тело ответа, с пояснением ошибки, за исключением запроса методом HEAD	
400	Bad Request. В запросе клиента присутствует синтаксическая ошибка или не указаны обязательные параметры запроса
401	Unauthorized. Для доступа у ресурсу необходима аутентификация клиента. Заголовок ответа сервера будет содержать поле WWW-Authenticate с перечнем условий аутентификации. Клиент может повторить запрос включив в новый запрос заголовок Authorization с требуемыми для аутентификации данными
402	Payment Required. Для доступа к ресурсу необходимо произвести оплату
403	Forbidden. Сервер отказывается выполнять запрос из за ограничений в доступе клиента к указанному ресурсу
404	Not Found. На сервере отсутствует запрашиваемый ресурс
405	Method Not Allowed. Указанный клиентом метод нельзя применять к запрошенному ресурсу. В ответе сервера будет заголовок Allow с перечисленными через запятую методами запроса к ресурсу
406	Not Acceptable. Запрошенный ресурс не удовлетворяет переданным в заголовке характеристикам. Если запрос был не HEAD, то сервер в ответе вернёт список доступных характеристик запрашиваемого ресурса
407	Proxy Authentication Required. Для доступа у ресурсу необходима аутентификация клиента на прокси сервере. Заголовок ответа сервера будет содержать поле WWW-Authenticate с перечнем условий аутентификации. Клиент может повторить запрос включив в новый запрос заголовок Authorization с требуемыми для аутентификации данными
408	Request Timeout. Время ожидания сервера окончания передачи данных клиентом истекло, запрос прерван
409	Conflict. Запрос не может быть выполнен из за конфликта обращения к ресурсу, например ресурс заблокирован другим клиентом
410	Gone. Ресурс ранее находившийся по указанному адресу удалён и не доступен, серверу не известно новый адрес ресурса
411	Length Required. Для выполнения запроса к указанному ресурсу клиент должен передать Content-Length в заголовке запроса
412	Precondition Failed. Сервер не смог распознать ни один заголовок запроса
413	Request Entity Too Large. Сервер отказывается выполнять запрос из за слишком большого тела запроса В случае если проблема временная, то возвращается заголовок Retry-After с указанием времени, по истечении которого можно повторить запрос
414	Request-URL Too Long. Сервер не может выполнить запрос к ресурсу из за слишком большого запрашиваемого URI
415	Unsupported Media Type. Сервер отказывается работать с указанным типом данных передаваемого контента данным методом запроса
416	Requested Range Not Satisfiable. Не корректный запрос с указанием Range . В поле Range заголовка запроса указан диапазон за пределами ресурса и отсутствует поле If-Range Если клиент передал байтовый диапазон, то сервер вернёт ресурса в заголовке Content-Range ответа
417	Expectation Failed. Сервер не может удовлетворить переданный от клиента заголовок запроса Expect (ждать)
418	I'm a teapot (RFC 2324). Я чайник. Первоапрельская шутка от IETF :)
420	Enhance Your Calm (Twitter). Ответ аналогичен ответу 429 – слишком много запросов, призван заставить клиента отправлять меньшее количество запросов к ресурсу. В ответе сервера возвращается заголовок Retry-After с указанием времени через которое можно повторить запрос к серверу
422	Unprocessable Entity (WebDAV). Не обрабатываемая сущность. Запрос к серверу корректный и верный, но в теле запроса имеется логическая ошибка из за которой не возможно выполнить запрос к ресурсу
423	Locked (WebDAV). В результате запроса ресурс успешно заблокирован
424	Failed Dependency (WebDAV). Успешность выполнения запроса зависит от не разрешимой зависимости (другого запроса) который ещё не выполнен
425	Unordered Collection. Запрос коллекции ресурсов в не корректном порядке или запрос к упорядоченному ресурсу в не корректном порядке
426	Upgrade Required. Клиенту необходимо обновить протокол запроса. В заголовке ответа сервера Upgrade и Connection возвращаются указания, которые должен выполнить клиент для успешности запроса
428	Precondition Required. Сервер требует от клиента выполнить запрос с указанием заголовков Range и If-Match
429	Too Many Requests. Слишком много запросов клиента к ресурсу. В ответе сервера возвращается заголовок Retry-After с указанием времени через которое можно повторить запрос к серверу
431	Request Header Fields Too Large. От клиента получено слишком много заголовков или длинна заголовка превысило допустимые размеры. Запрос прерван
434	Requested Host Unavailable. Запрашиваемый ресурс не доступен
444	No Response (Nginx). Сервер отказывается или не может вернуть результат запроса к ресурсу и незамедлительно закрывает соединение
449	Retry With (Microsoft). В запросе к ресурсу не достаточно информации для его успешного выполнения. В заголовке ответа сервера передаётся Ms-Echo-Request с указанием необходимых полей
450	Blocked by Parental Controls (Microsoft). Запрос заблокирован системой «родительский контроль»
451	Unavailable For Legal Reasons. Доступ к ресурсу закрыт по юридическим причинам или по требованиям органов государственной власти
456	Unrecoverable Error. Обработка запроса вызывает не обрабатываемые сбои в базе данных или её таблицах
499	Client Closed Request (Nginx). Код введён nginx для записи в логи. Указывает на то что клиент закрыл соединение не попытавшись получить от сервера ответ
Коды информирующие клиента об ошибках на стороне сервера	
На любой запрос сервер всегда возвращает тело ответа, с пояснением ошибки, за исключением запроса методом HEAD	
500	Internal Server Error. Любая не описанная ошибка на стороне сервера
501	Not Implemented. Сервер не имеет возможностей для удовлетворения доступа к ресурсу или реализация обработки запроса ещё не закончена
502	Bad Gateway. Сервер, выступающий в роли шлюза или прокси, получил не корректный ответ от вышестоящего сервера
503	Service Unavailable. По техническим причинам сервер не может выполнить запрос к ресурсу. В ответе сервер возвращает заголовок Retry-After с указанием времени через которое клиент может повторить запрос
504	Gateway Timeout. Сервер, выступающий в роли шлюза или прокси, не дождался ответа от вышестоящего сервера
505	HTTP Version Not Supported. Сервер отказывается поддерживать указанную в запросе версию HTTP протокола
506	Variant Also Negotiates. Ошибка на стороне сервера связанная с циклической или рекурсивной задачей которая не может завершиться

507	Insufficient Storage (WebDAV). Не достаточно места или ресурсов пользователя для выполнения запросов. Код может возвращаться как по причинам физической не хватки ресурсов так и по причине текущих лимитов пользователя
508	Loop Detected (WebDAV). Бесконечный цикл. Запрос завершился в результате вызванного на сервере бесконечного цикла который не мог закончиться
509	Bandwidth Limit Exceeded (Apache). Запрос прерван в связи с превышением со стороны клиента ограничений на скорость доступа к ресурсам
510	Not Extended . На сервере отсутствует расширение которое необходимо для успешного выполнения запроса к ресурсу
511	Network Authentication Required . Запрос был прерван сервером посредником, прокси или шлюзом из за необходимости авторизации клиента на сервере посреднике до начала выполнения запросов
598	Network Read Timeout Error . Ответ возвращается сервером который является прокси или шлюзом перед вышестоящим сервером и говорит о том что сервер не смог считать ответ на запрос к вышестоящему серверу
599	Network Connect Timeout Error . Ответ возвращается сервером который является прокси или шлюзом перед вышестоящим сервером и говорит о том что сервер не смог установить связь с вышестоящим сервером (подключиться к серверу)

Внутренние коды ошибок возвращаемые в структуре error

code	message		
Внутренние коды ошибок, возвращаются вместе с HTTP кодом 500-599			
Код	Возвращаемое описание ошибки в message		Внутренняя ошибка для разработчиков или администраторов
-1	Внутренняя ошибка сервера		Любая не конкретизированная ошибка
-2	Внутренняя ошибка сервера		Ошибка определения IP адреса
-3	Внутренняя ошибка сервера		Ошибка получения DNS имени
-4	Внутренняя ошибка сервера		Ошибка разбора json в структуру
-5	Внутренняя ошибка сервера		Ошибка работы с jpeg библиотекой
-6	Внутренняя ошибка сервера		Ошибка работы с базой данных
-7	Внутренняя ошибка сервера		Ошибка сохранения файла в хранилище файлов
-8	Внутренняя ошибка сервера		Отказ како-то компонента стороннего разработчика (vendor) вызвавшего панику. Паника была восстановлена, но что конкретно произошло не ясно.
-9	Внутренняя ошибка сервера		Ошибка получение файла по идентификатору файла из хранилища
-10	Внутренняя ошибка сервера		Ошибка копирование структуры данных
-11	Внутренняя ошибка сервера		Ошибка получения файла из хранилища в виде io.ReadSeekerCloser()
-12	Внутренняя ошибка сервера		Ошибка удаления файла в файловом хранилище
Публичные коды ошибок, возвращаются с кодами HTTP входящими в диапазон 100-499			
1	Ошибка, логин или пароль указан не верно		
2	Ошибка, указанный токен доступа не верный или просрочен		
3	Выбран не поддерживаемый язык		
4	Переданы неверные данные. Данный код ответа сопровождается HTTP кодом ответа 400		
	Структура объекта errors – массив объектов:		
	Имя	Тип	Описание
	field	Строка	Название поля
	fieldValue	Строка	Переданное значение поля field
	message	Строка	Описание ошибки
5	Не переданы данные в формате JSON или XML		

Оглавление

					описание
					коды ошибок
Core					
B	F	GET	/api/v1.0/version	Текущая версия приложения	
		GET	/api/v1.0/ping	Проверка доступности сервера	
		GET	/api/v1.0/panic	Проверка ошибки 5**	
		PUT	/api/v1.0/settings/time	Серверное время в UTC	
Какой-то функционала бизнес логики					
		OPTIONS	/api/v1.0/myitem	Получение состояния данных в БД	
		POST	/api/v1.0/myitem	Сохранение в БД данных	
		GET	/api/v1.0/myitem/{id}	Получение данных из БД по ID	
		DELETE	/api/v1.0/myitem/{id}	Удаление данных в БД по ID	

Текущая версия приложения

[оглавление](#)

Метод возвращает текущую версию приложения в семантике «Semantic Versioning 2.0.0» <http://semver.org/>

Адрес	/api/v1.0/version		
Метод	GET		
Входящие данные			
Заголовки	Нет		
Параметры	Нет		
Исходящие данные			
Заголовки	Имя	Значение	Описание
	Content-Type	application/json	Формат передаваемых данных
	Retry-After	{waitTimeSec}	Рекомендуемое время ожидания до выполнения следующего запроса
Параметры	Имя	Значение	Описание
	version	Строка	Текущая версия приложения
HTTP коды	Описание		
200	Ok. Успешный запрос. Результат запроса передаётся в заголовке или теле ответа Успешный запрос		
500	Internal Server Error. Любая не описанная ошибка на стороне сервера Сервер не в состоянии вернуть ответ		
Внутренние коды ответа			

Где:
{waitTimeSec} – Рекомендуемое время ожидания до выполнения следующего запроса

Проверка доступности сервера

[оглавление](#)

Метод предназначен для проверки доступности сервера, действительности токена доступа и своевременного закрытия интерфейса пользователя в случае не доступности сервера или вывода иных сообщений в UI

ВНИМАНИЕ! Данный метод не продлевает время жизни токена сессии

Адрес	/api/v1.0/ping?accessToken={token} /api/v1.0/ping		
Метод	GET		
Входящие данные			
Заголовки	Имя	Значение	Описание
	Content-Type*	application/json	Формат передаваемых данных
	Authorization*	Bearer {token}	Токен доступа
Параметры	Нет		
Исходящие данные			
Заголовки	Имя	Значение	Описание
	WWW-Authenticate	Token realm="api"	Указатель необходимости пройти авторизацию и получить токен доступа
Параметры	Нет		
HTTP коды	Описание		
204	No Content. Успешный запрос, в ответе передаются только заголовки без тела сообщения. Клиент не должен обновлять тело документа/ресурса, но может применить к нему полученные данные. Сервер доступен		
401	Unauthorized. Для доступа у ресурсу необходима аутентификация клиента. Заголовок ответа сервера будет содержать поле WWW-Authenticate с перечнем условий аутентификации. Клиент может повторить запрос включив в новый запрос заголовок Authorization с требуемыми для аутентификации данными Токен доступа не передан или передан пустой		
403	Forbidden. Сервер отказывается выполнять запрос из за ограничений в доступе клиента к указанному ресурсу Токен доступа передан, но является не верным		
Внутренние коды ответа			

Где:
{token} – Токен доступа пользователя

Проверка ошибки 5**

[оглавление](#)

Метод предназначен для проверки как сервер реагирует на ошибки в коде. Сервер вызывает исключение которое сам же должен пере-хватить и выдать корректный ответ с кодом ошибки 500-599, при этом в лог должно быть записано максимум информации о возникшей ошиб-ке. Соединение не должно быть сброшено и пользователю должно вернуться либо "no content", либо информация о том что произошла ошибка. При запросе с заголовком «application/json» возвращается ответ в JSON
При запросе без заголовка или с иным другим заголовком ответ возвращается в HTML

ВНИМАНИЕ! Данный метод вызывается без авторизации

Адрес	/api/v1.0/panic		
Метод	GET		
Входящие данные			
Заголовки	Имя	Значение	Описание
	Content-Type	application/json	Формат передаваемых данных
Параметры	Нет		
Исходящие данные			
Заголовки	Имя	Значение	Описание
	Content-Type	application/json	Формат передаваемых данных
Параметры	Описание и стек трэйс паники в формате html или json		
HTTP коды	Описание		
500	Internal Server Error. Любая не описанная ошибка на стороне сервера Сервер доступен, но на сервере произошла ошибка вызвавшее прерывание выполнения кода В случае запроса с content-type json, ответ дополняется структурой error с описанием ошибки		
Внутренние коды ответа			
-1	Внутренняя ошибка сервера		

Серверное время в UTC

[оглавление](#)

Передача на сервер локального времени агента в UTC таймзоне, для определения корректности времени агента и вычислении delta которую необходимо добавлять к меткам времени при передаче метрик на сервер.

После получения ответа на запрос агент вычисляет время затраченное на запрос, делит его пополам и на основе серверного и локально-го времени получает дельту, если разница между дельта полученной с сервера и вычисленной локально больше 1 секунды, то берётся среднее между дельтами.

ВНИМАНИЕ! Данный метод вызывается без авторизации

Адрес	/api/v1.0/settings/time		
Метод	PUT		
Входящие данные			
Заголовки	Имя	Значение	Описание
	Content-Type	application/json	Формат передаваемых данных
Параметры	Имя	Значение	Описание
	time	Строка	Дата и время локального времени агента приведённая в UTC таймзону Дата и время в стандарте ISO 8601
Исходящие данные			
Заголовки	Имя	Значение	Описание
	Content-Type	application/json	Формат передаваемых данных
Параметры	Имя	Значение	Описание
	time	Строка	Дата и время серверного времени приведённая в UTC таймзону Дата и время в стандарте ISO 8601
	delta	Число	Посчитанная разница между локальным временем клиента и серверным временем в наносекундах. Отрицательное или положительное число В результате сложения дельты и локального времени клиента в UTC должно получиться серверное время в UTC
HTTP коды	Описание		
200	Ok. Успешный запрос. Результат запроса передаётся в заголовке или теле ответа В теле ответа запрошенные данные		
400	Bad Request. В запросе клиента присутствует синтаксическая ошибка или не указаны обязательные параметры запроса Передан не верный запрос, не верная структура данных или иная ошибка не позволившая корректно разобрать входящие данные в формате json		
Внутренние коды ответа			
4	Переданы неверные данные		

Получение состояния данных в БД

[оглавление](#)

Метод возвращает структуру описывающую состояние БД для сущности myitem

ВНИМАНИЕ! Данный метод вызывается без авторизации

Адрес	/api/v1.0/myitem		
Метод	OPTIONS		
Входящие данные			
Заголовки	Нет		
Параметры	Нет		
Исходящие данные			
Заголовки	Имя	Значение	Описание
	Content-Type	application/json	Формат передаваемых данных
	Retry-After	{waitTimeSec}	Рекомендуемое время ожидания до выполнения следующего запроса
Параметры	Имя	Значение	Описание
	count	Число	Количество записей в БД
	ids	Массив	Массив идентификаторов сущностей базы данных
HTTP коды	Описание		
204	No Content. Успешный запрос, в ответе передаются только заголовки без тела сообщения. Клиент не должен обновлять тело документа/ресурса, но может применить к нему полученные данные Успешное выполнение запроса		
500	Internal Server Error. Любая не описанная ошибка на стороне сервера Внутренняя ошибка сервера		
Внутренние коды ответа			

Где:
{waitTimeSec} – Рекомендуемое время ожидания до выполнения следующего запроса

Сохранение в БД данных

[оглавление](#)

Метод сохраняет в базе данных новую сущность и возвращает её ID
ВНИМАНИЕ! Данный метод вызывается без авторизации

Адрес	/api/v1.0/myitem		
Метод	POST		
Входящие данные			
Заголовки	Имя	Значение	Описание
	Content-Type*	application/json	Формат передаваемых данных
Параметры	Имя	Значение	Описание
	date	Строка	Любая дата и время в стандарте ISO 8601, не может быть пустым
	number	Число	Любое положительное или отрицательное число кроме 0 и меньше 1024
	text	Строка	Любая не пустая строка
Исходящие данные			
Заголовки	Имя	Значение	Описание
	Content-Type	application/json	Формат передаваемых данных
	Retry-After	{waitTimeSec}	Рекомендуемое время ожидания до выполнения следующего запроса
Параметры	Имя	Значение	Описание
	id	Число	Уникальный идентификатор сохраненной сущности
HTTP коды	Описание		
200	Ok. Успешный запрос. Результат запроса передаётся в заголовке или теле ответа Успешное выполнение запроса		
400	Bad Request. В запросе клиента присутствует синтаксическая ошибка или не указаны обязательные параметры запроса Не верные входящие данные, подробности ошибки в структуре ответа		
500	Internal Server Error. Любая не описанная ошибка на стороне сервера Сервер доступен, но не в состоянии выполнить запрос		
Внутренние коды ответа			

Где:
{waitTimeSec} – Рекомендуемое время ожидания до выполнения следующего запроса

Получение данных из БД по ID

[оглавление](#)

Метод возвращает данные из БД по ID

ВНИМАНИЕ! Данный метод вызывается без авторизации

Адрес	/api/v1.0/myitem/{id}		
Метод	DELETE		
Входящие данные			
Заголовки	Нет		
Параметры	Нет		
Исходящие данные			
Заголовки	Имя	Значение	Описание
	Content-Type	application/json	Формат передаваемых данных
	Retry-After	{waitTimeSec}	Рекомендуемое время ожидания до выполнения следующего запроса
Параметры	Имя	Значение	Описание
	id	Число	Уникальный идентификатор сущности
	date	Строка	Любая дата и время
	number	Число	Любое число
	text	Строка	Любая строка
HTTP коды	Описание		
200	Ok. Успешный запрос. Результат запроса передаётся в заголовке или теле ответа Успешное выполнение запроса		
500	Internal Server Error. Любая не описанная ошибка на стороне сервера Сервер доступен, но не в состоянии выполнить запрос		
Внутренние коды ответа			

Где:
{waitTimeSec} – Рекомендуемое время ожидания до выполнения следующего запроса
{id} — Уникальный идентификатор сущности

Удаление данных в БД по ID

[оглавление](#)

Метод удаляет сущность из БД по её ID

ВНИМАНИЕ! Данный метод вызывается без авторизации

Адрес	/api/v1.0/myitem/{id}		
Метод	DELETE		
Входящие данные			
Заголовки	Нет		
Параметры	Нет		
Исходящие данные			
Заголовки	Имя	Значение	Описание
	Content-Type	application/json	Формат передаваемых данных
	Retry-After	{waitTimeSec}	Рекомендуемое время ожидания до выполнения следующего запроса
Параметры	Нет		
HTTP коды	Описание		
204	No Content. Успешный запрос, в ответе передаются только заголовки без тела сообщения. Клиент не должен обновлять тело документа/ресурса, но может применить к нему полученные данные Успешное выполнение запроса		
404	Not Found. На сервере отсутствует запрашиваемый ресурс Сущность не существует		
500	Internal Server Error. Любая не описанная ошибка на стороне сервера Сервер доступен, но не в состоянии выполнить запрос		
Внутренние коды ответа			

Где:
{waitTimeSec} – Рекомендуемое время ожидания до выполнения следующего запроса
{id} — Уникальный идентификатор