## Требования к API. Описание структуры сообщений

### JSON Web Token

#### Общее описание

JSON Web Token (JWT) — открытый стандарт (RFC 7519) создания token-ов доступа, основанный на формате JSON.

При первом логине клиенту от сервера выдается token, образованный хеш-функцией от уникальных данных пользователя (id, логин, пароль). В базе данных регистрируется пара token — id. При следующих запросах для авторизации клиент передает token. Token имеет определенное время жизни.

Преимущества JWT:

Использование JWT не требует хранение дополнительных данных о выданных сессиях: сервер должен только проверить подпись.

JWT хранит дополнительную полезную информацию о пользователях.

JWT делает возможным предоставление одновременного доступа к различным доменам и сервисам.

#### Структура JWT

JWT состоит из трех основных частей: заголовка (header), полезной нагрузки (payload) и подписи (signature), которые разделены точками («.»). JWT в общем виде выглядит следующим образом:

xxxxxxx.yyyyyyy.zzzzzzz

Первые два элемента JWT — это JSON объекты определенной структуры, которые формируются отдельно. Третий элемент вычисляется на основе первых двух элементов и зависит от выбранного алгоритма хэширования подписи.

##### Заголовок

Заголовок (header) состоит из двух параметров: «typ» (тип token-a) и «alg» (наименование алгоритма хэширования подписи).

В случае подписанного JWT в заголовок могут быть добавлены дополнительные параметры, например:

**kid:** уникальный идентификатор используемого ключа, когда указывается набор ключей (Key ID).

Пример заголовка:

{

 "alg": "RS256",

 "typ": "JWT",

 "kid": "vetB22WKLuCj5x4rBc5VAgyxCX0b0n2hQeKEC4R-L88"

}

##### Полезная нагрузка

Payload - данные, которые передаются в token-е. Payload не шифруется при использовании token-a.

Стандарт предусматривает несколько зарезервированных параметров:

Таблица 3 Зарезервированные параметры

| **Наименование** | **Описание** |
| --- | --- |
| iss | Чувствительная к регистру строка или URI, которая является уникальным идентификатором стороны, генерирующей JWT (issuer) |
| sub | Чувствительная к регистру строка или URI, которая является уникальным идентификатором стороны, о которой содержится информация в JWT (subject). Значения с этим ключом уникальны в контексте стороны, генерирующей JWT |
| aud | Массив чувствительных к регистру строк или URI, являющийся списком получателей JWT. При получении JWT принимающая сторона проверяет наличие себя в получателях. Если нет в списке, то проигнорировать JWT (audience) |
| exp | Время в формате Unix Time, определяющее момент, когда JWT станет невалидным (expiration). |
| nbf | Время в формате Unix Time, определяющее момент, когда JWT станет валидным (not before). |
| jti | Строка, определяющая уникальный идентификатор JWT (JWT ID) |
| iat | Время в формате Unix Time, определяющее момент, когда JWT был создан (issued at). |
| sid | Строка, которая является уникальным идентификатором сессии |
| typ | Тип token-a |
| Azp | Чувствительная к регистру строка или URI, которая является уполномоченной стороной, для которой JWT был издан |
| acr | Authentication Context Class Reference |

##### Подпись

Подпись (signature) вычисляется на основе заголовка и полезной нагрузки. При попытке изменения данных в token-e подпись нельзя изменить, не зная приватного ключа. При аутентификации приватным ключом выступает пароль пользователя (или хеш от пароля).

Заголовок и полезная нагрузка приводятся к формату JSON, затем переводятся в base64:

Header:

eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9

Payload:

eyJpc3MiOiJDb2RleCBUZWFtIiwic3ViIjoiYXV0aCIsImV4cCI6MTUwNTQ2Nzc1Njg2OSwiaWF0IjoxNTA1NDY3MTUyMDY5LCJ1c2VyIjoxfQ

Затем две строки соединяются через точку («.») и хэшируются указанным в заголовке алгоритмом.

HS256('eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9' + '.' + 'eyJpc3MiOiJDb2RleCBUZWFtIiwic3ViIjoiYXV0aCIsImV4cCI6MTUwNTQ2Nzc1Njg2OSwiaWF0IjoxNTA1NDY3MTUyMDY5LCJ1c2VyIjoxfQ', 'password') = '0ynjTRZT9Uk77TnGy\_g9Mxi1decLBjKxQK6e2dVzDJo'

Результат работы алгоритма является подписью. Сгенерированный token выглядит следующим образом:

eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiOiJDb2RleCBUZWFtIiwic3ViIjoiYXV0aCIsImV4cCI6MTUwNTQ2Nzc1Njg2OSwiaWF0IjoxNTA1NDY3MTUyMDY5LCJ1c2VyIjoxfQ.0ynjTRZT9Uk77TnGy\_g9Mxi1decLBjKxQK6e2dVzDJo

### Общие требования к коммуникационному протоколу

Кодировка передачи сообщений – UTF-8.

Сообщения передаются в формате JSON, в заголовке Content-Type указывается application/json. Параметры запроса и ответа на запрос являются JSON-объектами, передаются в теле запроса и ответа на запрос соответственно, если не указано иное.

После выполнения аутентификации (ввода логина и пароля), пользователь получает JWT токен. Все запросы направляются с заголовком Authentication: Bearer token.

В качестве способа взаимодействия с сервером используется REST API. Данный способ взаимодействия основывается на протоколе передачи гипертекста HTTP (Hypertext Transfer Protocol). Это стандартный протокол интернета, созданный для передачи гипертекста. Каждый объект на сервере в HTTP имеет свой уникальный URL-адрес в строго последовательном формате.

В REST API есть пять основных методов HTTP, которые используют для действий с объектами на серверах:

GET (получение информации о данных или списка объектов);

DELETE (удаление данных);

POST (добавление новых данных);

PUT (добавление новых данных или редактирование имеющихся данных).

PATCH (используется для частичного изменения ресурса)

В целях повышения безопасности при использовании REST API для взаимодействия WEB-приложения с внешним сервером используется HTTPS. HTTPS является расширением протокола HTTP. Данные в протоколе HTTPS передаются поверх криптографических протоколов TLS.

Таблица 4 Типы параметров, используемых в запросах:

| **Тип параметра** | **Описание** |
| --- | --- |
| Number (N) | Числовой тип. В качестве разделителя используется точка. Если дополнительно не указана размерность данных тогда допустимо использование любой длины и точности. В качестве разделителя используется точка.Длина и точность указываются через запятую – например: N,12,2 |
| String (S) | Строковый тип. Последовательность символьных значений. Допустимо использование любых печатных символов. Если дополнительно не указана размерность - тогда возможно использование любой длины, но не более 3000 байт. Незначащие пробелы с обеих сторон строки запрещены.  |
| Date (D) | Тип даты. Формат YYYY-MM-DDThh:mm:ss±hh:mm. |
| Hash (X) | Хэш-код. Цифры и буквы латинского алфавита.  |

Формат даты и времени: YYYY-MM-DDThh:mm:ss±hh:mm, где

* YYYY – год,
* MM – месяц,
* DD – число,
* префикс T – разделитель даты и времени,
* hh – часы (24-х часовой формат),
* mm – минуты,
* ss – секунды,
* ±hh:mm - разница между местным временем и UTC.
* Формат даты: YYYY-MM-DD, где YYYY – год, MM – месяц, DD – число.

 **Обозначения, используемые для указания обязательности элемента**:

* "0-1" — элемент не обязателен. Может принимать только одно значение.
* "1-1" — элемент обязателен. Может принимать только одно значение.
* "0-\*" — элемент не обязателен. Может принимать список значений.
* "1-\*" — элемент обязателен. Может принимать список значений.

В случае, когда тип элемента не указан, элемент может принимать список значений.

По умолчанию все элементы являются не обязательными и имеют тип "String", если специально не оговорено другое.

В протоколе, в сообщения - ответы могут быть добавлены новые элементы. Такие изменения не должны влиять на корректность обработки сообщений-ответов на стороне фронт-офисных решений.

#### Структура запросов

Header:

Заголовок включает в себя параметры авторизации или другие параметры, которые являются общими для участников взаимодействия.

Path:

Параметры в Path являются частью URI, отделяются знаком «/» от параметров строки запроса. В документе эти параметры выделяются фигурными скобками. Параметры пути используются для идентификации конкретного ресурса или ресурсов.

Query:

Параметры Query указываются после знака вопроса «?» в URI. Символ «?», параметры, которые следуют за «?», и их значения называются «строкой запроса». В строке запроса параметры перечисляются один за другим, разделяются амперсандом («&»). Порядок параметров строки запроса не имеет значения. Параметры запроса используются для фильтрации ресурсов.

Body:

Параметры в Body передаются в теле HTTP запроса. Формат передачи параметров указывается в заголовке Content-Type.

#### Коды ответов на запросы

Запросы, результат которых подразумевает создание объекта, в случае успешной обработки запроса возвращают в ответе пустое тело и код 201 Created.

Запросы, результат которых подразумевает изменение объекта (добавление ролей пользователю и т.д.), в случае успешной обработки запроса возвращают в ответе пустое тело и код 204 No Content.

Когда сервер авторизации не может авторизовать действие от клиента или пользователя (например, истек token доступа), сервер возвращает в ответе код 401 Unauthorized.

Когда владелец token-a доступа не имеет права на выполнение определенного действия, север авторизации возвращает код ошибки 403 Forbidden.

Если во время обработки запроса возникает логическая ошибка, сервер возвращает код ошибки 400-499. Коды состояния определены в секции 10 описания спецификации протокола HTTP/1.1 (RFC 2616).

Таблица 5 Описание ошибки в JSON формате

| **Название** | **Описание** | **Тип данных** | **Обязательность присутствия в ответе** |
| --- | --- | --- | --- |
| error | Код ошибки. При отсутствии ошибок передается 0 | String | Обязательный |
| error\_description | Описание ошибки | String | Опциональный |