Interakt

API музыкальной платформы

Техническая документация: интерфейс RPC

Интерактив

Татарский Павел
07.02.2012

Оглавление

Описание

API музыкальной платформы позволяет получать списки контента на витрине, а также производить покупки выбранного контента с последующей возможностью скачать и прослушать. Все методы АПИ доступны как через GET, так и через POST запросы. Каждая витрина обладает своим собственным идентификатором (storefront_id), который передается при каждом запросе к API. Каждый пользователь API обладает своим уникальным кодом API (api_key), который также передает при каждом запросе. Каждый пользователь API может использовать API только с определенного диапазона IP-адресов. После совершения покупки пользователь API получает возможность формировать специальные ссылки для загрузки купленного контента конечными пользователями. Пользователь API обязан следить за тем, чтобы сгенерированные им ссылки получали только конечные пользователи оплатившие покупку. Пользователь API не должен разглашать свой api_key и storefront_id конечному пользователю.

Авторизация

Авторизация пользователя производится при помощи метода «auth». При аутентификации пользователя создается сессия, в которой фиксируются все параметры клиента, необходимые для защищённой работы клиента с АПИ платформы (api_key, storefront_id, и т.п.). Авторизованный пользователь может обращаться непосредственно к API без указания полного набора параметров для аутентификации, указывая лишь минимальный набор обязательных параметров, необходимых для распознавания пользователя.

В следующих разделах, при описании сигнатуры методов АПИ, параметры, необходимые для авторизации, будут указываться условным обозначением [AUTH_ARGS] без описания назначения.

Запрос:

```
auth/auth.json?domain=partner&api_key={api_key}&storefront_id={storefront_id}&user_id={user_id}&csrf_token={csrf_token}&auth_key={auth_key}&time={time}
```

```
user_id – пользователь, которого необходимо авторизовать
```

```
csrf_token – уникальный идентификатор сессии
```

```
time – текущее время в формате YYYYMMDDHHMMSS
```

```
auth_key – шифрованная подпись. Этот ключ формируется как hash-функция (md5) от строки вида:
```

```
{имя параметра1}={значение параметра1}{имя параметра2}={значение параметра2}…{кодовое слово}
```

```
Все параметры должны быть отсортированы по имени параметра.
```

```
При проверке подписи используются все переданные параметры.
```

Ответ:

```
Если пользователь авторизовался, то возвращается идентификатор сессии, если нет, то текст ошибки.
```

Если авторизация пользователя прошла успешно, при дальнейшей работе с методами АПИ, необходимо и достаточно указывать только 2 обязательных параметра: csrf_token и PHPSESSID (в значение этого параметра подставляется идентификатор сессии, отданный при успешном прохождении авторизации). Т.е. в описании сигнатур методов в запросах к АПИ под условным обозначением [AUTH_ARGS] имеется в виду наличие этих обязательных параметров.

Разделы витрины

Контент в витрине разбит по разделам, как правило соответствующих музыкальным направлениям (жанрам).

Список разделов

Возвращает список разделов сортированных по названию.

Запрос:

content/sections.json?[AUTH_ARGS]

Ответ:

[{"name": <string>, "id": <int>}, …]

name – название раздела

id – идентификатор раздела

Здесь и далее:

<string> - строка, пример: "Джаз" (кавычки обязательны)

<int> - целое число, пример: 57

<binary> - двоичные данные

Список контента в разделе

Возвращает список контента в разделе, отсортированный по названию.

Запрос:

content/content.json?[AUTH_ARGS]
&section_id={section_id}&page_num={page_num}&page_size={page_size}&q={q}

section_id – идентификатор раздела (необязательный параметр, в случае если передан параметр «q»)

page_num – номер страницы (по-умолчанию 1)

page_size – количество элементов на странице (по-умолчанию 10)

q – строка поискового запроса (необязательный параметр)

Ответ:

[{"name": <string>, "artists": {<string>: <string>, …}, "content_id": <string>, "preview_picture_url": <string>, "preview_melody_url": <string>, "content_url": <string>, "artists": {<string>: <string>, …}, "duration": <int>}, …]

name – название контента

artists – массив исполнителей (главный идёт первым)

content_id – идентификатор контента

preview_picture_url – ссылка на изображение (обложка альбома, либо фото главного исполнителя)

preview_melody_url – ссылка на контент, урезанный по длительности

content_url – опционально, ссылка на контент (если контент не куплен, то ссылка на контент урезанный по длительности)

duration – длительность (сек), для альбома длительность всего альбома

Если нет контента удовлетворяющего условиям поиска, то сервис возвращает []

Покупки

При покупке API может опционально вызывать сервер приложения для дополнительной валидации покупки. В таком случае только провалидированные покупки будут зарегистрированы. Подобный механизм способен существенным образом сократить возможность вмешательства третьих лиц в процесс формирования покупки. Для того чтобы воспользоваться этой возможностью, обратитесь к разработчику.

Покупка контента

Производит покупку контента. Данное действие тарифицируется.

Запрос:

purchase/buy.json?[AUTH_ARGS]
&content_id={content_id}&user_id={user_id}&tariff={tariff}

content_id – идентификатор контента (необязательный параметр при подписках)

user_id – идентификатор пользователя

tariff – идентификатор тарифа (определяет способ покупки и стоимость). Необязательный параметр, если для партнера определен только 1-н тариф по умолчанию.

Ответ:

null

Список покупок

Отображает список покупок пользователя. Если покупки просматривает пользователь, которому их подарили или у которого есть подписка, то он получает ссылку на контент, иначе – ссылки на предпрослушку (укороченная версия мелодии).

Запрос:

purchase/purchases.json?[AUTH_ARGS]
&user_id={user_id}&page_num={page_num}&page_size={page_size}

user_id – идентификатор пользователя покупки которого мы смотрим

page_num – номер страницы (по-умолчанию 1)

page_size – количество элементов на странице (по-умолчанию 10)

Ответ:

```
[{"purchase_id": <int>, "content_id": <string>, "buyer_id": <string>, "name":
<string>, "purchase_date": <string>, "artists": {<string>: <string>, …},
"preview_picture_url": <string>, "preview_melody_url": <string>, "content_url":
<string>}, …]
```

purchase_id – идентификатор покупки

content_id – идентификатор контента

buyer_id – идентификатор пользователя, совершившего покупку

name – название трека

purchase_date – дата свершения покупки в формате timestamp without time zone (пример: "2012-07-31 15:28:12.594556")

artists – массив артистов в формате идентификатор артиста => навзание

preview_picture_url – ссылка на изображение (обложка альбома, либо фото исполнителя)

preview_melody_url – ссылка на трек, урезанный по длительности

content_url – ссылка на полный трек (появляется в случае если user_id = viewer_id)

Подписка

Версии: 2+

Пользователь может подключить подписку на определенный срок, что позволяет ему в течении этого срока прослушивать полные треки. По истечению срока действия подписки пользователь вновь имеет доступ только к предпрослушке. При повторной покупке подписки срок текущей подписки увеличивается на время приобретенной.

Подключение подписки

Версии: 2+

Подключает подписку на определенный срок. Если у юзера уже есть подписка – её срок действия продлевается на указанный период. При получении/продлении подписки, пользователь получает соответствующее уведомление.

Запрос:

subscription/subscribe.json?[AUTH_ARGS]&user_id={user_id}&duration={duration}

user_id – идентификатор пользователя, которому продлевают подписку

duration – длительность действия подписки в днях, начиная с текущего момента. Допустимый набор значений этого параметра предопределен для каждого партнера.

Ответ:

```
<string>|null – дата окончания подписки в формате «timestamp without time zone»
                (пример: '2012-12-17 23:36:26.484594')
```

Ошибки:

Wrong duration – некорректный формат продолжительности

Gift subscriptions is not implemented – дарение подписок ещё не реализовано

Проверка подписки

Версии: 2+

Проверяет наличие действующей подписки по идентификатору пользователя

Запрос:

```
subscription/subscription.json?[AUTH ARGS]&user_id={user_id}
```

```
user_id – идентификатор пользователя
```

Ответ:

```
<string>|null – дата окончания подписки в формате «timestamp without time zone»
                (пример: '2012-12-17 23:36:26.484594')
```

Получение контента

Купленный контент можно получить следующими путями: получить по ссылке, сформированной особым образом, получить список покупок или при получении списка контента (для всего ранее купленного контента будут возвращены ссылки на полную версию трека).

Получение по ссылке

Получение контента по ссылке можно реализовать путем генерации ссылки определенного вида, по которой конечный пользователь сможет скачать контент в течение ограниченного времени. Ссылка подписывается шифрованной подписью при помощи секретного слова. При загрузке контента проверяется подлинность пользователя по ip-адреса и наличие у него подписки. Так же есть возможность проверки по идентификатору сессии. Если у юзера нет подписки, или она истекла – отдаётся только превью контента.

Запрос:

```
d/{data'}/{filename}
```

```
data – это сформированная специальным образом строка в сжатом виде, эта строка в
исходном виде имеет следующий вид:
api_key={api_key}&storefront_id={storefront_id}&content_id={content_id}&purchase_id={
purchase_id}&time={time}&user_id={user_id}&sell_type={sell_type}&hash={hash}
```

```
data' = urlencode(base64_encode(gzcompress(data, 9)))
```

```
hash – шифрованная подпись
```

hash = md5(salt + api_key + storefront_id + content_id + purchase_id + time + user_id + sell_type_id + ip_address + session_id)

salt – оговоренное заранее секретное слово

ip_address – IP-адрес конечного пользователя

session_id – идентификатор сессии из cookie session_id (если данная cookie не передается при загрузке контента, то в hash её включать не надо)

time – время генерации ссылки (формат YYYYMMDDHHMMSS, пример: 20121028145363)

purchase_id – идентификатор покупки

sell_type_id – идентификатор типа продажи

filename – имя файла, которое получит абонент, может быть любым

Ответ:

<binary> - контент

Если ссылка некорректна, то возвращается текст “Bad link”

Сгенерированная ссылка будет действовать в течение получаса.

Получение превью

Следует помнить, что в редких случаях превью может отсутствовать.

Получение музыкального превью

pm/{data'}/{file_name}

data – это сформированная специальным образом строка в сжатом виде, эта строка в исходном виде имеет следующий вид:
api_key={api_key}&storefront_id={storefront_id}&content_id={content_id}&hash={hash}

data' = urlencode(base64_encode(gzcompress(data, 9)))

hash – шифрованная подпись

hash = md5(salt + api_key + storefront_id + content_id)

file_name – имя файла, может быть любым

Ответ:

<binary> - контент

Если ссылка некорректна, то возвращается текст “Bad link”

Получение картиночного превью

pp/{data'}/{file_name}

data – это сформированная специальным образом строка в сжатом виде, эта строка в исходном виде имеет следующий вид:
api_key={api_key}&storefront_id={storefront_id}&content_id={content_id}&template_id={template_id}&hash={hash}

data' = urlencode(base64_encode(gzcompress(data, 9)))

hash – шифрованная подпись

hash = md5(salt + api_key + storefront_id + content_id + template_id)

template_id – идентификатор шаблона

file_name – имя файла, может быть любым

Ответ:

<binary> - контент

Если ссылка некорректна, то возвращается текст “Bad link”

Справочники

Шаблоны картиночного превью

Название	template_id
24x24.gif	40
33x33.gif	38
30x40.gif	24
39x52.gif	25
50x50.gif	6
51x68.gif	26
60x60.gif	3
85x85.png	36
75x100.gif	27
90x90.gif	16
110x80.gif	9
100x100.gif	7
100x100.jpg	39
100x118.gif	19
96x128.gif	17
128x128.gif	5
128x160.gif	18
176x144.gif	22
176x208.gif	20
220x170.jpg	11
220x170.gif	23
200x200.png	33
200x200.gif	8
208x208.gif	21
240x240.gif	28
320x240.jpg	30
450x280.jpg	10
567x425.gif	32

Приложения

Обработка ошибок

В случае ошибки вывод каждого из методов возвращает:

```
{"error_code": <int>, "error_message": <string>}
```

error_code – код ошибки

error_message – текст ошибки