

---

Интеграция со SPACES

---

---

## ТЕРМИНОЛОГИЯ

**Виджет** - кусок HTML и, возможно, JavaScript-кода, который автор игры вставляет на определенные страницы (или на базовую, например в footer), и который отобразит ссылку на vmmo.ru и/или рекламного партнера (в зависимости от того, по какой ссылке пришли на игру), а также любую другую информацию, которую будет нужно там разместить.

**Игра (Game)** - любая игра без привязки к любому компоненту системы. Идентифицируется по appId.

**Игровая система (GameSystem)** - конкретный экземпляр игры запущенный на определённом сервере. Идентифицируется по gameId или парой (домен, appId).

**Секретный ключ приложения (AppSecretKey)** - строка используемая для генерации запросов к приложениям. Каждое приложение знает собственный ключ.

**Сервер API (API Server)** - сервер API Playtox (далее по тексту - VMMO API). URL запросов API имеют вид <http://apiserver.vmmo.ru/api/>\*

**Паспорт (Playtox Passport)** - приложение обеспечивающие авторизацию пользователей в играх и приложениях Платформы Playtox посредством единой учетной записи. Также реализует авторизацию через внешние системы, в т.ч. соц.сети.

**Персонаж (Customer)** - игровой персонаж в любой из игровых систем. В игре идентифицируется по customerId, в системе идентифицируется парой (gameSystemId, customerId).

**Платеж (Payment)** - Реальная финансовая транзакция с использованием любой системы биллингов. Любая транзакция имеет уникальный идентификатор.

**Платежный Шлюз (ПШ)** - Приложение Платформы, предоставляющее сервисы по приему платежей от клиентов Игровых систем с использованием различных реальных платежных систем (коротких номеров SMS, Яндекс деньги, Qiwi и др.) и привязке персонажей к партнёрам.

Игровая Система и Платежный Шлюз взаимодействуют по протоколам HTTP/HTTPS.

**Платформа** - совокупность приложений Playtox обеспечивающие функционирование игр и взаимодействие с внешними системами.

**Пользователь Playtox (User)** - человек имеющий Playtox паспорт, идентифицируется playtox\_id.

---

---

# ОПИСАНИЕ ИНТЕГРАЦИИ

## Этапы реализации подключения

1. Реализовать требования принудительного входа в приложение, виджетов и осуществить подключение к платежному шлюзу.
2. Реализовать все 3 юзкейза, описанных ниже. Выполнить все условия, необходимые для прохождения QA, описанные в разделе Требования.
3. Предоставить вашему менеджеру данные для заведения игры в Spaces.
4. По окончании интеграции игра должна пройти QA Spaces.
5. После успешного прохождения игра встает на трафик.

## Данные для заведения игры

- Название игры
- Краткое описание игры
- Иконки (50x50 и 16x16, gif)
- Ссылка на биллинг игры
- Ссылка на вход в игру
- Ник обитателя Spaces, для получения доступа к sandbox
- IP адрес сервера

## Юзкейзы

В рамках задачи интеграции со spaces существует 3 юзкейса:

1. Вход в игру нового пользователя
2. Повторный вход в игру
3. Платёж персонажем, созданным из Spaces

## Алгоритмы реализации юзкейсов

1. Вход в игру нового или существующего пользователя
    - При входе в игру, Spaces отправляет пользователя в Паспорт
    - Паспорт, создав при необходимости нового пользователя Playtox, перенаправляет его на URL принудительной авторизации с соответствующим passport\_id.
    - Игра сама решает что делать с этим пользователем - создать нового персонажа, использовать существующего, предложить выбрать из списка существующих
    - После регистрации игра отправляет в Платформу нотификации, по результатам обработки которых, созданный персонаж привязывается к партнёру и Пользователю Playtox.
  2. Предполагается, что к моменту первого платежа все необходимые нотификации уже произведены и все привязки выполнены. Осуществление платежа происходит в рамках обычного взаимодействия между Игровой системой и ПШ, описанного ниже.
-

---

# ТРЕБОВАНИЯ SPACES

## Требования

1. полная, работающая, протестированная версия
2. достаточные серверные ресурсы (мин. 7000 одновременно играющих пользователей, добавление 15 000 новых в час, более 250 000 запросов картинки 128x128 одновременно)
3. служба поддержки пользователей
4. если игра/приложение осуществляет какое-либо действие от имени пользователя (например установка статуса), необходимо каждый раз перед совершением действия спрашивать разрешение пользователя на стороне приложения.
5. не содержит элементов, перечисленных в разделе "Ограничения"

## Требования к иконкам игры (приложения)

Необходимые параметры для основных картинок игры (16x16px и 50x50px). Картинка должна быть:

1. в формате GIF,
2. на прозрачном фоне,
3. Без теней,
4. Без эмблем "New", "Exclusive" и подобных.

## Ограничения по доступу приложений

1. На всех страницах приложения должны присутствовать хедер и футер сайта [spaces.ru](https://spaces.ru) (реализуется через механизм виджетов VMMO API).
  2. На страницах приложения не должно быть ссылок, которые уведут пользователя на другие ресурсы.
  3. На страницах приложения не допускается размещение каких-либо видимых счетчиков. Исключением является счетчик [mobytop.ru](https://mobytop.ru)
  4. В приложении запрещается сбор каких-либо данных о пользователях. Все данные можно получать только через API. Например, не допускается предложение пользователю указать: пароль, E-mail, номер телефона или другие контактные данные.
  5. Доступ пользователя к приложению должен осуществляться только через авторизацию на сайте Spaces.ru. При потере авторизации, также должен закрываться доступ к аккаунту пользователя в приложении.
  6. Каждый конкретный случай рассматривается отдельно. Администрация сайта оставляет за собой право не допускать приложение и по другим причинам.
-

---

## ПОДПИСЬ ЗАПРОСОВ К VMMO API

Для использования VMMO API, приложение должно быть зарегистрировано на стороне VMMO, при этом будет выдан идентификатор приложения и секретный ключ для формирования подписи. Каждый запрос через VMMO API снабжается следующими параметрами:

Параметр запроса	Тип	Обязательный	Описание
vmmo.app	String	Да	Идентификатор клиентского приложения, выдается при регистрации приложения. Вместе с идентификатором выдается секретный ключ, участвующий в формировании подписи запросов.
vmmo.sign	String	Да	Цифровая подпись параметров запроса, включены параметры начинающиеся с "vmmo.", за исключением "vmmo.sign". Алгоритм описан ниже.
vmmo.debug	String	Нет	Флаг тестового запроса, API сервер должен отдавать ответ такой же как был бы при отсутствии флага, но не совершая никаких реальных действий по изменению и т.п.

### Алгоритм подписи

Параметр vmmo.sign строится на основе параметров запроса, начинающихся с префикса "vmmo." путем их склейки. Параметры запроса сортируются в алфавитном порядке по имени параметра. Параметры и значения разделяются символом "=". В случае, если запрос содержит несколько значений с одним и тем же параметром - значения склеиваются так же в отсортированном порядке. В конец полученной строки добавляется секретный ключ. От полученной строки берется sha256 хэш и приводится в HEX формат.

### Пример

Параметры запроса:	<b>vmmo.app</b> : sampleapp, <b>vmmo.domain</b> : somehost.mobi, <b>vmmo.debug</b> : 1. <b>SecretKey</b> : 908475hf(*)
Строка для формирования подписи:	vmmo.app=sampleappvmmo.debug=1vmmo.domain=somehost.mobi 908475hf(*)
Значение подписи:	f9e4e071e3df3291d7e2da032e30bf64a82f1b79be2786b97a78fe557 d2c3d8

---

---

Итоговый пример  
запроса:

```
http://apiserver.vmmo.ru/api/authUser?  
vmmo.app=sampleapp&vmmo.domain=somehost.mobi&vmmo.debug  
=1&vmmo.sign=f9e4e071e3df3291d7e2da032e30bf64a82f1b79be27  
86b97a78fe557d2c3d8
```

## Реализация

1. Для PHP может использоваться метод `hash("sha256", $stringToSign);`
2. Java

```
public static String getSha256Sign(String st) {  
  
    try {  
        MessageDigest md = MessageDigest.getInstance("SHA-256");  
  
        md.update(st.getBytes(Charsets.UTF_8));  
  
        byte byteData[] = md.digest();  
  
        //convert the byte to hex format method 1  
        StringBuilder sb = new StringBuilder();  
        for (byte b : byteData) {  
            sb.append(Integer.toString((b & 0xff) + 0x100, 16).substring(1));  
        }  
        return sb.toString();  
  
    } catch (NoSuchAlgorithmException e) {  
        throw new RuntimeException(e);  
    }  
}
```

---

## ПРИНУДИТЕЛЬНЫЙ ВХОД В ПРИЛОЖЕНИЕ

Для обеспечения входа в клиентские приложения пользователей Playtox, на стороне приложения должен быть реализован метод принудительной авторизации пользователя и предоставлен URL, который должен быть сконфигурирован для каждого приложения на стороне Платформы. При перенаправлении пользователя на адрес AuthenticateUserURL приложения, будут указаны следующие параметры, кроме тех, что участвуют в формировании подписи:

Параметр запроса	Тип	Описание
vmmo.ts	Long	Timestamp перехода, количество миллисекунд с 1 января 1970 00:00:00 UTC. Приложение должно ограничивать время жизни переходов (например, 5 минут) во избежание использования ссылок третьими лицами.
vmmo.from	String	Обратный адрес, на который следует перенаправить пользователя в случае невозможности авторизации, истечении времени жизни запроса и т.д. Добавляются обратные параметры vmmo.sign, vmmo.fail=code, vmmo.reason=text. Где code - ниже описанный код ошибки, text - URLEncoded описание причины ошибки авторизации. Данный параметр необходим для обработки ошибки, если он не указан или содержит значение "TBD", ошибку следует обрабатывать и отображать средствами игры.
vmmo.session_attributes	String	Атрибуты пользователя, которые необходимо сохранять на стороне приложения после авторизации. Представляет из себя Base64 encoded JSON (проверочный пример ниже). Эти атрибуты следует передавать в неизменном виде в методы VMMO API где они требуются, например VMMO Widgets API.
vmmo.passport_id	String	URLEncoded OpenID идентификатор пользователя.
vmmo.customer_id	Long	customer_id пользователя, указывается, при наличии у пользователя нескольких персонажей в клиентском приложении. Приложение должно проверять соответствие vmmo.customer_id и vmmo.passport_id.
vmmo.requested_url	String	URLEncoded строка на стороне приложения, куда следует отправить пользователя после успешной авторизации.
vmmo.display_name	String	Отображаемое имя пользователя (ник). Если не указан в запросе, то в игре совпадает с OpenID идентификатором vmmo.passport_id. Используется при формировании подписи только если присутствует в запросе.

### Коды ошибок

Код	Описание
1	Неверная подпись запроса.

---

---

Код	Описание
2	Неверные параметры запроса, отсутствуют vmmo.ts, vmmo.passport_id и т.п.
3	Неверная комбинация vmmo.passport_id и vmmo.customer_id.
4	Истекло время жизни запроса.
5	Прочее.

### Пример запроса

*http://[домен\_игры]/forceAuth?*

*vmmo.ts=1382330644520&vmmo.display\_name=USER&vmmo.customer\_id=3843119&vmmo.passport\_id=http%3A%2F%2Fpassport.vmmo.ru%2Fpages%2FIndexPage%2Fuserid%2FUSER&vmmo.from=vmmo&vmmo.sign=917470548afa2f798f38a8caef8e22fb35663135560c6a29afede3079d3d8c35*

---

---

# ОТСЛЕЖИВАНИЕ СЕССИЙ И АКТИВНЫХ ДЕЙСТВИЙ ИГРОКОВ

## Краткое описание

Для сбора и анализа значимых действий пользователя игры используется POST метод `/api/user-events`. В теле запроса содержится дамп пользовательских сессий за определенный интервал времени, дамп представляет из себя разбитый на части CSV файл в кодировке UTF-8. Части сжимаются с помощью стандартного архиватора ZIP. При вызове метода должны быть указаны следующие параметры, кроме тех, что участвуют в формировании подписи:

Параметр запроса	Тип	Обязательный	Описание
<code>vmmo.dump_id</code>	Long	Да	Порядковый номер дампа пользовательских действий.
<code>vmmo.dump_check_sum</code>	String	Да	Контрольная сумма части дампа, подсчитывается с помощью алгоритма MD5.
<code>vmmo.part_number</code>	Long	Нет	Номер части, для разбитого на куски дампа. Максимальный рекомендуемый размер части - 10 мегабайт.
<code>vmmo.total_parts</code>	Long	Нет	Общее число частей для разбитого дампа, в случае отсутствия считается что дамп состоит из одной части.
<code>vmmo.dump_version</code>	String	Нет	Версия формата дампов. По умолчанию считается, что дамп отправляется в формате текущей (последней) версии.

## Формат дампа

Текущая версия формата: 1.0.0.

Дамп представляет из себя CSV файл с разделителем `,` со следующими полями:

`customer_id`, `login`, `level`, `experience`, `playtime`, `game_money`, `event_timestamp`, `event_type`, `event_details`.

При наличии в значении поля CSV строки разделителя, поле помещается в двойные кавычки. Кавычки в значении при этом дублируются (см. [описание формата CSV](#)).

Внутри полей CSV файла не должно быть символов перевода строки, каждое событие должно быть записано в одну строку.

---

---

## Расшифровка полей

Поле	Тип	Описание
customer_id	Long	Идентификатор персонажа.
login	String	Игровое имя персонажа, получение персонажем логина означает прохождение регистрации.
level	Long	Уровень персонажа на момент события.
experience	Long	Опыт персонажа.
playtime	Long	Суммарный playtime персонажа на момент события, в миллисекундах.
game_money	Long	Количество игровой валюты у персонажа на момент события. Центы, серебро, алмазы и т.п.
event_timestamp	TimeStamp with TimeZone	Момент времени события, в формате "YYYY-MM-DDThh:mm:ssTZD", например "1999-07-01T19:30:45+10:00".
event_type	Long	Тип события, возможные варианты описаны ниже.
event_details	String	CSV строка с разделителем ";" описывающая суть события. При наличии в значении поля CSV строки разделителя, поле помещается в одинарные кавычки. Кавычки в значении при этом дублируются.

---

## Типы событий

Код	Название события	Формат детализации	Описание
0	Вход в игру	method;identity;ip;forwarded_for;user_agent;domain:url;http_referer;opera_ua;opera_phone;phone	<p>Событие вход персонажа в игру. Должно отправляться при входе персонажа в игру, в т.ч. для вновь зарегистрированного пользователя.</p> <p>method - метод HTTP запроса пользователя (GET/POST) identity - идентификатор пользователя (OpenID) ip - IP адрес пользователя forwarded_for - значение HTTP заголовка "X-Forwarded-For" запроса пользователя user_agent - значение HTTP заголовка "User-Agent" domain - значение HTTP заголовка "Host" url - URL запроса http_referer - значение HTTP заголовка "Referer" opera_ua - любое из значений, передаваемое в заголовках "Device-Stock-UA", "X-Original-User-Agent", "X-Device-User-Agent", "X-OperaMini-Phone-UA" opera_phone - значение HTTP заголовка "X-OperaMini-Phone" phone - модель телефона пользователя (возможно пустое)</p>
1	Смена логина	old_login	<p>Событие смены имени (логина) персонажа, новое имя берется из поля login записи дампа. old_login - старое имя пользователя, возможно пустое.</p>

---

## Ответ сервера

Сервер в ответ на запрос по регистрацию дампа отдает JSON с кодом статуса обработки и кратким описанием, например:

```
{"status":0,"message":"OK"}
```

## Коды ответов

Код	Описание
0	Запрос успешно обработан. Дамп принят в обработку.
1	Неверная подпись запроса.
2	Неверные параметры запроса, отсутствуют идентификатор дампа, неверная контрольная сумма и т.п.
3	Приложению запрещено использовать данный метод.
4	Дамп (часть) уже был получен от игры.
5	Внутренняя ошибка сервера

---

---

# ВИДЖЕТЫ

## Техническая реализация

Получение виджетов из внешних систем берет на себя VMMO, для игр предоставляется общий метод /api/widgets. При вызове метода должны быть указаны следующие параметры, кроме тех, что участвуют в формировании подписи:

Параметр запроса	Тип	Описание
vmmo.domain	String	Домен, на котором находится пользователь.
vmmo.session_attributes	String	Аттрибуты сессии, полученные при принудительном входе в приложение.
vmmo.customer_id	Long	Идентификатор персонажа.

## Пример запроса

<http://apiserver.vmmo.ru/api/widgets?>

`vmmo.app=sampleapp&vmmo.domain=vmmo.ru&vmmo.session_attributes=&vmmo.customer_id&vmmo.debug=1&vmmo.o.sign=264d54877fe3730c3ef49357b83d63e7fa2c831784f0c4ccb8997f51875ba2cc`

## Формат ответа

VMMO API метод на запрос виджетов возвращает JSON следующего формата:

```
{ "status": 0, "message": "OK", "ts": 1351761905044, "cache": 300, "widgets": [{"id": "header", "content": "BASE64 Content"}, ...]}
```

Поле	Тип	Описание
status	Long	Результат обработки запроса.
message	String	Краткое описание результата обработки.
ts	Long	Timestamp генерации виджета, количество миллисекунд с 1 января 1970 00:00:00 UTC.
cache	Long	Время в секундах на которое разрешается кэшировать результат запроса, отсчитывается от параметра ts.
widgets	Array	JSON массив виджетов.
id	String	Идентификатор виджета. <ul style="list-style-type: none"><li>• "header" - должен располагаться вверху страниц,</li><li>• "header-simple" - разрешен к использованию на критичных ко времени ответа страницах (онлайн-сражения и т.п.),</li><li>• "footer" - должен располагаться внизу страницы,</li><li>• "footer-simple" - разрешен к использованию только на критичных ко времени ответа страницах.</li></ul> При отсутствии "header-simple" и/или "footer-simple" должны использоваться "header" и "footer" соответственно.

---

---

Поле	Тип	Описание
content	String	Base64 encoded содержимое виджета, которое нужно вставить на странице в соответствии с id.

### Коды ответов

Код	Описание
0	Запрос успешно обработан.
1	Неверная подпись запроса.
2	Неверные параметры запроса, не найден пользователь, неверные сессионные атрибуты.
3	Приложению запрещено использовать данный метод.
4	Время жизни сессии пользователя окончено, его нужно направить на метод авторизации в портале.

---

---

# ИНТЕГРАЦИЯ С ПЛАТЕЖНЫМ ШЛЮЗОМ

## Профиль игровой системы для взаимодействия с платежным шлюзом

Параметр	Описание
CheckURL	URL игровой системы для проверки корректности платежа
PaymentAvisoURL	URL игровой системы для уведомления об успешном платеже
SuccessURL	URL игровой системы на который происходит переход после оплаты в платежной системе
FailURL	URL игровой системы на который происходит переход в случае ошибки и невозможности оплаты в платежной системе
gameSystemId	Идентификатор игровой системы (выдается платежным шлюзом)
PaymentGatewayURL	URL по которому игровая система передает запрос на проведение платежа (выдается платежным шлюзом)

## Сценарий взаимодействия

1. Игровая система формирует страницу для игрока для заполнения основных параметров платежа
2. Игрок нажимает «Оплатить/купить», игровая система отправляет платежную форму на URL платежного шлюза
3. Платежный шлюз проверяет корректность параметров платежа и отправляет запрос в игровую систему для подтверждения платежа на CheckURL (из профиля игровой системы)
4. В случае если игровая система подтверждает корректность полученной шлюзом информации платежный шлюз пересылает (redirect) запрос платежной системе
5. Пользователь (игрок) взаимодействует с платежной системой для подтверждения и осуществления платежа
6. После прохождения реального платежа в платежной системе платежный шлюз вызывает игровую систему для оповещения о статусе платежа с использованием PaymentAvisoURL
7. Происходит возврат в игровую систему с использованием SuccessURL либо FailURL.

## Примеры URL

PaymentGatewayURL: <http://vmmo.ru/Payment/onlinePayment>

Для других стран, тестового сервера и т.д. домен платежного шлюза может меняться.

При подключении Игровой системы к Платформе, разработчиком Игровой системы должны быть предоставлены CheckURL, PaymentAvisoURL, SuccessURL, FailURL.

## Уведомление о создании персонажа

При создании нового персонажа, платформа должна нотифицироваться о данном событии со стороны игры на **NewUserURL**.

---

**NewUserURL**

URL платежного шлюза для уведомления о регистрации игрока. <http://vmmo.ru/PaymentGateway/newUser>

**Передаваемые параметры**

Параметр запроса	Тип	Описание
gameSystemId	Long	Идентификатор игровой системы
customerId	Long	Идентификатор зарегистрированного игрока
domain	String	Домен игры на котором зарегистрировался игрок
pageUri	String	URL страницы на которой произошло событие регистрации
ip	String	IP адрес игрока
originatingIp	String	Заголовок «HTTP_X_FORWARDED_FOR» (опционально, если есть)
userAgent	String	User-agent игрока, заголовок «HTTP_USER_AGENT» http запроса игрока
operaMiniUserAgent	String	Заголовок «HTTP_X_OPERAMINI_PHONE_UA» http-запроса игрока (опционально, только если заголовок есть)
registrationDate	Timestamp with Timezone	Время регистрации игрока в формате "YYYY-MM-DDThh:mm:ssTZD", например "1999-07-01T19:30:45+10:00" (опционально)
login	String	Логин зарегистрированного пользователя (если пользователь зарегистрирован).

**Форма запроса на проведение платежа**

Запрос отправляется методом GET или POST на PaymentGatewayURL (из профиля игровой системы), параметры запроса:

paymentSystemId	идентификатор платежной системы, например yandex (список выдается платежным шлюзом)
gameSystemId	идентификатор игровой системы (из профиля)

scid	идентификатор витрины (из профиля)
sum	сумма платежа в рублях (разделитель — десятичная точка, разделители тысяч недопустимы)
customerId	идентификатор покупателя (уникальный и специфичный для игровой системы)
additionalInfo	Дополнительная информация (произвольное содержимое но не более 2048 символов) — необязательное поле
failureUrl	Url encoded строка передаваемая игровой системой. Используется, если FailURL из профиля системы нежелателен (другой домен, и т. п.). - рекомендуемое поле. При отсутствии — используется FailURL.
successUrl	Url encoded строка передаваемая игровой системой. Используется, если SuccessURL из профиля системы нежелателен (другой домен, и т. п.). - рекомендуемое поле. При отсутствии — используется SuccessURL.
любые другие параметры, специфические для магазина, отличные от вышеуказанных	Могут использоваться для идентификации сессии и будут возвращены при переходе по SuccessURL или FailURL.

### Формат запроса на подтверждение (проверку) платежа игровой системой

Запрос отправляется методом GET или POST на CheckURL игровой системы

Параметр	Описание
requestDateTime	Время отправки запроса платежным шлюзом в формате YYYY-MM-DDThh:mm:ssTZD, например 2010-11-20T10:20:06+03:00
action	Check
paymentSystemId	Идентификатор платежной системы
gameSystemId	Идентификатор игровой системы
sum	Сумма
internalInvoiceId	идентификатор платежа в платежном шлюзе (генерируется платежным шлюзом, уникален в рамках системы)
customerId	идентификатор покупателя (из исходного запроса)
additionalInfo	Дополнительная информация (переданная в исходном запросе)

---

## Формат запроса с уведомлением об осуществлении платежа к игровой системе

Отправляется методом GET или POST на PaymentAvisoURL игровой системы

Параметр	Описание
requestDateTime	время отправки запроса платежным шлюзом
action	PaymentSuccess
paymentSystemId	идентификатор платежной системы
gameSystemId	идентификатор игровой системы
internalInvoiceId	внутренний идентификатор платежа в платежном шлюзе
externalInvoiceId	внешний идентификатор платежа в реальной платежной системе
customerId	идентификатор покупателя
orderCreatedDateTime	момент регистрации заказа в платежной системе
orderSumAmount	сумма платежа покупателя
gameSystemSumAmount	сумма которая поступит на р/с игровой системы (в общем случае соответствует сумме orderSumAmount за вычетом комиссии платежной системы и платежного шлюза)
orderIsPaid	содержит 1 если платеж оплачен
additionalInfo	дополнительная информация переданная в исходном запросе

---

---

# РЕАЛИЗАЦИЯ ЮЗКЕЙСОВ

## Вход в игру нового пользователя

Вход в игру осуществляется через URL принудительной авторизации [http://\[домен игры\].mobi/forceAuth](http://[домен игры].mobi/forceAuth)

### Параметры запроса

Название	Значение
vmmo.from	
vmmo.passport_id	openID пользователя в формате  <a href="http://passport.vmmo.ru/pages/IndexPage/userid/123456@spaces">http://passport.vmmo.ru/pages/IndexPage/userid/123456@spaces</a>  где 123456 - уникальный (внутренний) идентификатор пользователя Spaces (он же передается в vmmo.session_attributes как login).
vmmo.session_attributes	Base64-кодированная строка с параметрами app_sid и login в формате Json. Данные параметры потребуются в дальнейшем для запроса виджетов и платежей, их нужно сохранить в сессии.
vmmo.ts	Время запроса в миллисекундах.
vmmo.customer_id	Если передан, предполагается, что нужно залогинить персонажа с данным customer_id.

### Платежная страница

На платежной странице игры (см. пример [http://\[домен игры\].spaces.ru/billing/info](http://[домен игры].spaces.ru/billing/info)) предполагается наличие ссылки "Купить"

Ссылка ведет на URL онлайн платежа в ПШ:

<http://vmmo.ru/Payment/onlinePayment?>

[spaces\\_koef=2&customerId=...&spaces\\_desc=...&spaces\\_app\\_sid=...&gameSystemId=...&sum=100&spaces\\_user\\_id=...&additionalInfo=&paymentSystemId=spaces&currency=Coins](http://vmmo.ru/Payment/onlinePayment?spaces_koef=2&customerId=...&spaces_desc=...&spaces_app_sid=...&gameSystemId=...&sum=100&spaces_user_id=...&additionalInfo=&paymentSystemId=spaces&currency=Coins)

### Передаваемые параметры специфичные для Spaces

Параметр	Описание
spaces_koef	коэффициент конвертации игровой валюты в Spaces Coins, если за 1 ед. игровой валюты пользователь должен заплатить 2 монеты, значит значение spaces_koef=2 и т.д.
spaces_desc	текстовое описание покупки (например "Покупка игровой валюты игры Игра").
session_attributes	строка vmmo.session_attributes

---

## Общие параметры для всех платежей

Параметр	Описание
customerId	customerId пользователя
gameSystemId	идентификатор игры в ПШ
sum=100	не используется, но должно быть задано значение
additionalInfo	пустое значение
paymentSystemId=spaces	константа
currency=Coins	константа

## Проверка платежа

После поступления запроса на создание платежа Spaces ПШ делает запрос в игру на check\_url аналогично другим онлайн системам.

Далее пользователь перенаправляется на страницу Spaces и совершает платеж.

## Завершение успешного платежа

После поступления платежа, ПШ нотифицирует игру через aviso\_url аналогично ОК и другим онлайн платежным системам. В параметре paymentSystemId передается значение spaces. В параметре orderSumAmount передается количество купленных Spaces Coins.

## Выход из игры

При логaute необходимо перенаправить пользователя на API метод паспорта /api/authUser с параметром logout=true.

## Виджеты

Виджеты запрашиваются через VMMO API по стандартной схеме. В запросе нужно передавать в запросе параметр vmmo.session\_attributes полученный при логине в из Spaces.

Также при получении клиентским приложением кода ошибки "Время жизни сессии пользователя окончено" (например, код 4 из Widgets API) приложение должно отправить пользователя на /api/authUser, используя следующие параметры, кроме тех, что участвуют в формировании подписи:

Параметр запроса	Тип	Обязательный	Описание
vmmo.domain	String	Да	Домен, на котором произошел инцидент.
vmmo.session_attributes	String	Нет	Переданные при принудительнов входе сессионные атрибуты пользователя.
vmmo.passport_id	String	Нет	OpenID идентификатор пользователя.
vmmo.customer_id	Long	Да	customer_id пользователя

---

Параметр запроса	Тип	Обязательный	Описание
vmmo.logout	Boolean	Нет	В случае указания - обрабатывается выход из клиентского приложения.

### Виджеты

Виджеты запрашиваются через VMMO API по стандартной схеме. В запросе нужно передавать в запросе параметр `vmmo.session_attributes` полученный при логине в из Spaces.

---

## КОНТАКТНАЯ ИНФОРМАЦИЯ

При возникновении любых вопросов, связанных с интеграцией, пишите на [ik@playtox.com](mailto:ik@playtox.com) или обращайтесь к вашему менеджеру.

---