

**Игровой кластер**

**Документация по SDK Игры Ростелеком**

**Версия 0.5**

2025  
Москва

## Аннотация

Настоящий документ содержит сводные данные по SDK Игры Ростелеком, применяемого в процессе разработки информационной системы «Игровой кластер».

В настоящем документе приведены сведения о предусловиях и процедуре перехода на SDK Игры Ростелеком, а также сведения о клиентском API.

# Содержание

ПЕРЕЧЕНЬ ТЕРМИНОВ, СОКРАЩЕНИЙ И ОПРЕДЕЛЕНИЙ	4
1. ПРЕДУСЛОВИЯ	5
2. ПЕРЕХОД НА SDK ИГРЫ РОСТЕЛЕКОМ	6
3. КЛИЕНТСКИЙ API	8
4. ОТДЕЛЬНЫЕ ЗАМЕЧАНИЯ	10
4.1. Уточнение по функции SteamAPI_RestartAppIfNecessary	11
5. ПОДГОТОВИТЕЛЬНЫЕ ДЕЙСТВИЯ ИЗДАТЕЛЕЙ К ЗАГРУЗКЕ ИГР	12
5.1. Клиентский API	15

## Перечень терминов, сокращений и определений

Термин / сокращение	Определение / пояснение
API	Application Programming Interface. Набор компонент, с помощью которых одна система взаимодействует с другой
SDK	(Software Development Kit) Набор инструментов, библиотек, примеров кода, документации и других ресурсов, который помогает разработчикам создавать программное обеспечение для определённой платформы, операционной системы или сервиса.
Игровой кластер, Игры Ростелеком	Онлайн-сервис, предназначенный для цифрового распространения компьютерных игр, игрового контента и ключей к играм.
Издатель	Юридическое лицо, осуществляющее материально-техническое обеспечение издания (подготовку, производство и выпуск) компьютерных игр.
ОС	Операционная система.
УЗ	Учетная запись

## 1. Предусловия

SDK Игры Ростелеком реализует отдельные элементы функционала Steam SDK. Переход на SDK Игры Ростелеком реализуется через подмену нативной динамической библиотеки `steam_api64.dll`, поставляемой Steam, на аналогичную библиотеку `steam_api64.dll`, предоставляемую Rostelecom. Примечание: здесь и далее по тексту под «нативной» подразумевается библиотека, предоставляемая Steam.

В данный момент поддерживаются следующие версии Steam SDK: **1.38a, 1.50, 1.53, 1.62**. Список поддерживаемых версий будет расширяться по мере роста количества клиентов. Добавление поддержки еще одной версии Steam SDK в настоящее время занимает в среднем три рабочих дня. Если используемая вами версия Steam SDK отсутствует в списке поддерживаемых версий (например, вы используете **1.60**), свяжитесь с нами.

---

**Примечание:** В настоящее время поддерживается только 64-битная версия библиотеки и только ОС семейства Microsoft Windows. Также планируется поддержка 64-битных ОС семейства Linux.

---

## 2. Переход на SDK Игры Ростелеком

Для перехода в SDK Игры Ростелеком необходимо выполнить следующие действия:

1. Скачать архив с версиями SDK Игры Ростелеком. Ссылка для скачивания архива:

[https://cdn.igrovoi.rt.ru/public/downloads/rtgames-sdk/rtgames\\_libs.zip](https://cdn.igrovoi.rt.ru/public/downloads/rtgames-sdk/rtgames_libs.zip)

**Примечание:** Ссылка может быть изменена позднее.

 3.62.82.82	14.10.2025 11:32	Папка с файлами
 6.6.99.59	14.10.2025 11:32	Папка с файлами
 6.91.21.57	14.10.2025 11:32	Папка с файлами
 9.60.44.10	14.10.2025 11:32	Папка с файлами
 date.txt	14.10.2025 11:32	Текстовый докум...

Рисунок 1 – Пример содержимого архива после распаковки

Архив содержит несколько папок и файл date.txt. В файле date.txt хранится информация о времени сборки библиотек. В папках вида 3.62.82.82 и 9.60.44.10 хранятся отладочные (debug) и релизные (release) версии библиотеки steam\_api64.dll.

2. Определить версию используемой нативной динамической библиотеки steam\_api64.dll.

Для этого необходимо найти нативную библиотеку steam\_api64.dll, вызвать контекстное меню правой кнопкой мыши, перейти в «Свойства» → «Подробно». В пункте «Версия файла» таблицы указана версия файла.

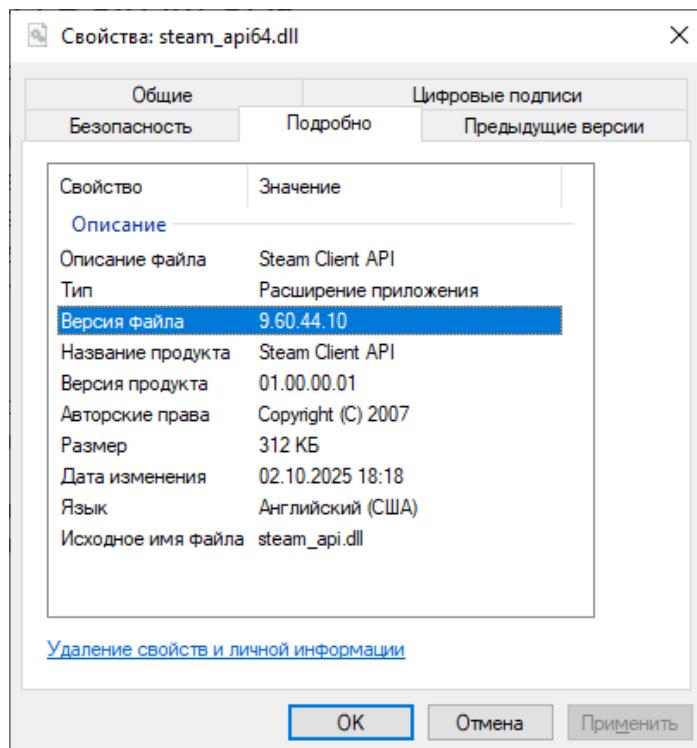


Рисунок 2 – Пример отображения версии файла steam\_api64.dll

В данном примере видно, что версией нативной steam\_api64.dll является 9.60.44.10 (что соответствует Steam API версии 1.62).

3. Выполнить подмену нативной библиотеки steam\_api64.dll на библиотеку, предоставленную RT.

На шаге 2 мы выяснили, что используется версия 9.60.44.10 нативной динамической библиотеки. Значит необходимо перейти в папку 9.60.44.10 разархивированной rtgames\_libs. Далее в зависимости от типа сборки (отладочная или релизная) перейти в папку debug или release, и скопировать оттуда с заменой steam\_api64.dll вместо нативной steam\_api64.dll. Этого должно быть достаточно для перехода на SDK Игры Ростелеком, выполнять пересборку проекта целиком не надо.

Если же возникли проблемы и необходима пересборка проекта, то необходимо также скопировать с заменой steam\_api64.lib и пересобрать проект.

---

**Примечание:** В настоящее время отладочная и релизная версии библиотеки ничем принципиально по функционалу не различаются. В отладочной версии, в отличие от релизной, присутствуют отладочные символы. В будущем планируется, что в отладочной версии будет больше вывода в log-файлы.

---

### 3. Клиентский API

В SDK Игры Ростелеком реализованы следующие элементы API Steam:

*SteamAPI\_RestartAppIfNecessary()*  
*SteamInternal\_SteamAPI\_Init()*  
*SteamAPI\_Shutdown()*  
*SteamAPI\_IsSteamRunning()*  
*SteamInternal\_ContextInit()*  
*SteamInternal\_FindOrCreateUserInterface()*  
*SteamInternal\_CreateInterface()*  
*SteamInternal\_FindOrCreateGameServerInterface()*  
*SteamAPI\_GetHSteamPipe()*  
*SteamAPI\_GetHSteamUser()*  
*SteamAPI\_GetSteamInstallPath()*  
*SteamAPI\_RegisterCallback()*  
*SteamAPI\_UnregisterCallback()*  
*SteamAPI\_RegisterCallResult()*  
*SteamAPI\_UnregisterCallResult()*  
*SteamAPI\_RunCallbacks()*  
*SteamAPI\_ReleaseCurrentThreadMemory()*  
*SteamAPI\_ManualDispatch\_Init()*  
*SteamAPI\_ManualDispatch\_GetNextCallback()*  
*SteamAPI\_ManualDispatch\_FreeLastCallback()*  
*SteamAPI\_ManualDispatch\_GetAPICallResult()*  
*SteamApps()*  
*SteamApps()→BIsSubscribedApp()*  
*SteamApps()→BIsAppInstalled()*  
*SteamApps()→GetDLCCount()*  
*SteamApps()→BIsDLCInstalled()*  
*SteamApps()→GetDLCDATAByIndex()*  
*SteamApps()→GetAppInstallDir()*  
*SteamUser()*  
*SteamUser()→BLoggedOn()*

---

Для остальных методов и функций реализованы заглушки. То есть если, например, будет вызвана функция *SteamInput()*, она вернет указатель на mock-экземпляр класса *ISteamInput* соответствующей версии. Однако методы этого класса не будут делать ничего и будут возвращать пустые значения.

## 4. Отдельные замечания

SDK Игры Ростелеком несколько отличается по поведению от Steam SDK в части appID игры и ее дополнений (DLC).

Steam SDK подразумевает 2 механизма получения игрой ее appID: запрос у клиента Steam (в нашем случае – Launcher), или же appID можно указать в файле steam\_appid.txt, который должен лежать в той же папке, которая является рабочей папкой для вызова исполняемого файла с игрой. Во втором случае SDK берет appID из этого файла. И в Steam сборка одной и той же игры, но в тестовом или релизном окружении, имеют разные appID. У нас же ситуация другая. У игры есть один appID, но два типа окружения: релизное и тестовое. Соответственно, может отличаться содержимое файла steam\_appid.txt.

- Если в файле указано только число, это число берется за appID, окружение считается **тестовым**.
- Если в файле указано число/test, то окружение считается **тестовым**.
- Если в файле указано число/release, то окружение считается **релизным**.

Пример: у нас есть игра с appID 152 (у Steam таковой является игра Space War с appID 480).

Содержимое файла steam_appid.txt	appID	Тип окружения
152	152	Тестовое
152/test	152	Тестовое
152/release	152	Релизное

Если игра запущена, appID и тип окружения получены из текстового файла (допустим, окружение **тестовое**), то при вызове любых методов, которые используют appID (например ISteamApps::GetDLCCount()), SDK будет считать, что используется то окружение, которое указано в текстовом файле (в данном примере – **тестовое**).

## 4.1. Уточнение по функции SteamAPI\_RestartAppIfNecessary

Необходимо также сделать отдельное уточнение относительно реализации функции *SteamAPI\_RestartAppIfNecessary*. Данная функция принимает на вход всего один параметр: **uint32 unOwnAppID** – appID приложения, которое надо проверить на предмет запуска его клиентом SDK Игры Ростелеком. Если присутствует файл *steam\_appid.txt*, то параметр **uint32 unOwnAppID** игнорируется, а сама функция возвращает *false*.

Если же этого файла нет, то функция должна через клиента SDK Игры Ростелеком инициировать запуск игры клиентом SDK Игры Ростелеком, а сама функция вернуть *true*. И вот здесь необходимо уточнить, что будет запущена именно релизная версия игры с **appID unOwnAppID**.

### Замечания к уточнению

Как показала практика, при переходе на RTGames SDK возникают трудности, если в коде игры использована функция *SteamAPI\_RestartAppIfNecessary*. При регистрации в RTGames игра получает новый appID, который отличается от appID, полученного в Steam. Поэтому, если в игре есть вызов *SteamAPI\_RestartAppIfNecessary*, необходимо подставить туда новый appID и пересобрать игру. К сожалению, очень часто разработчики забывают это сделать, совершая одну из этих наиболее распространенных ошибок:

- Вызов *SteamAPI\_RestartAppIfNecessary(...)* с appID, полученным при регистрации игры в Steam (при публикации в RTGames у игры новый appID)
- Вызов *SteamAPI\_RestartAppIfNecessary(...)* с нулевым appID (это может быть константа **k\_uAppIdInvalid**, если код написан на C++; это может быть 0, если код написан на C# и Unity)
- Попытки обойти вызов *SteamAPI\_RestartAppIfNecessary* путем подкладывания файла *steam\_appid.txt*.

RTGames не позволяет публиковать игру вместе с *steam\_appid.txt*. Создание же этого файла в ходе исполнения кода игры, перед вызовом *SteamAPI\_RestartAppIfNecessary* и *SteamAPI\_Init()*, является плохой практикой, так как это нарушает логику проверки прав владения пользователем данной игрой.

---

**Примечание:** Уточнения по поводу расположения файла *steam\_appid.txt* – файл *steam\_appid.txt* (если он необходим), должен располагаться в корне рабочей папки игры (*working directory*).

---

**Таблица соответствия версии SDK Steam версии нативной динамической библиотеки steam\_api64.dll**

<b>Версия Steam SDK</b>	<b>Версия нативной библиотеки steam_api64.dll</b>
1.11	0.91.58.20
1.12	0.96.33.48
1.13	1.6.99.61
1.14	1.8.87.63
1.15	1.10.1.46
1.16	1.14.66.23
1.17	1.23.45.93
1.18	1.30.50.46
1.19	1.33.85.19
1.31	2.37.91.26
1.35	3.4.27.90
1.35a	3.22.75.80
1.36	3.27.76.74
1.37	3.42.61.66
1.38a	3.62.82.82
1.39	3.75.32.7
1.40	3.92.72.58
1.41	4.4.91.85
1.42	4.28.51.7

<b>Версия Steam SDK</b>	<b>Версия нативной библиотеки steamapi64.dll</b>
1.43	4.95.20.30
1.44	4.95.20.30
1.45	5.19.38.62
1.46	5.25.65.21
1.47	5.53.33.78
1.48	5.69.73.98
1.48a	5.69.73.98
1.49	5.92.36.75
1.50	6.6.99.59
1.51	6.28.18.86
1.52	6.75.97.18
1.53	6.91.21.57
1.53a	6.91.21.57
1.54	7.30.20.28
1.55	7.40.51.27
1.56	7.96.0.44
1.57	8.2.21.95
1.58	8.33.9.23
1.59	8.63.11.84
1.60	8.97.99.70

<b>Версия Steam SDK</b>	<b>Версия нативной библиотеки steamapi64.dll</b>
1.61	9.31.86.4
1.62	9.60.44.10

## 5. Подготовительные действия Издателей к загрузке игр

Для публикации Игры в магазин Игры Ростелекома необходимо иметь УЗ в магазине и партнерском сайте.

Для создания УЗ в магазине достаточно иметь номер телефона.

Для создания УЗ и ее верификации необходимо заполнить всю финансовую и юридическую информацию по компании.

---

**Примечание:** В настоящее время Игры Ростелекома поддерживают игры только ОС семейства Microsoft Windows 64-битная версии. Также планируется поддержка 64-битных ОС семейства Linux.

---

Игры могут содержать SDK Игры Ростелеком либо загружаться без SDK.

SDK Игры Ростелеком реализует отдельные элементы функционала Steam SDK.

Переход на SDK Игры Ростелеком реализуется через подмену нативной динамической библиотеки *steam\_api64.dll*, поставляемой Steam, на аналогичную библиотеку *steam\_api64.dll*, предоставляемую Rostelecom.

### 5.1. Клиентский API

В SDK Игры Ростелеком реализованы следующие элементы API Steam:

*SteamAPI\_RestartAppIfNecessary()*  
*SteamInternal\_SteamAPI\_Init()*  
*SteamAPI\_Shutdown()*  
*SteamAPI\_IsSteamRunning()*  
*SteamInternal\_ContextInit()*  
*SteamInternal\_FindOrCreateUserInterface()*  
*SteamInternal\_CreateInterface()*  
*SteamInternal\_FindOrCreateGameServerInterface()*  
*SteamAPI\_GetHSteamPipe()*  
*SteamAPI\_GetHSteamUser()*  
*SteamAPI\_GetSteamInstallPath()*  
*SteamAPI\_RegisterCallback()*

---

*SteamAPI\_UnregisterCallback()*  
*SteamAPI\_RegisterCallResult()*  
*SteamAPI\_UnregisterCallResult()*  
*SteamAPI\_RunCallbacks()*  
*SteamAPI\_ReleaseCurrentThreadMemory()*  
*SteamAPI\_ManualDispatch\_Init()*  
*SteamAPI\_ManualDispatch\_GetNextCallback()*  
*SteamAPI\_ManualDispatch\_FreeLastCallback()*  
*SteamAPI\_ManualDispatch\_GetAPICallResult()*  
*SteamApps()*  
*SteamApps()→BIsSubscribedApp()*  
*SteamApps()→BIsAppInstalled()*  
*SteamApps()→GetDLCCount()*  
*SteamApps()→BIsDLCInstalled()*  
*SteamApps()→GetDLCDataByIndex()*  
*SteamApps()→GetAppInstallDir()*  
*SteamUser()*  
*SteamUser()→BLoggedOn()*

Для остальных методов и функций реализованы заглушки. То есть если, например, будет вызвана функция *SteamInput()*, она вернет указатель на mock-экземпляр класса *ISteamInput* соответствующей версии. Однако методы этого класса не будут делать ничего и будут возвращать пустые значения.

---