1. **Friendship Token**

Эта API предназначена для генерации **токена дружбы**, который используется на платформе Steam для добавления пользователей в друзья.

**Основные функции:**

1. **Эндпоинт**:
   * POST /generateFriendToken — создает токен дружбы.
2. **Входные данные**:
   * Принимает JSON с обязательным параметром:
     + user\_id — уникальный идентификатор пользователя, создающего токен.
3. **Выходные данные**:
   * Успешный ответ (200):
     + success: указывает на успешность операции (1 — успешно, 0 — ошибка).
     + token: сам сгенерированный токен дружбы.
     + invite: объект с информацией о токене приглашения:
       - invite\_token: токен для приглашения.
       - invite\_limit: лимит использования токена.
       - invite\_duration: срок действия токена (в секундах).
       - time\_created: время создания токена (UNIX).
       - valid: указывает, действителен ли токен (1 — да, 0 — нет).
4. **Ошибки**:
   * 400: Некорректный запрос, возвращается описание ошибки.
   * 500: Ошибка сервера, возвращается описание проблемы.

**Пример использования:**

Пользователь отправляет запрос с user\_id, получает токен, который может быть передан другому человеку для добавления в друзья или приглашения.

1. **Friendship Tokens List API**

Эта API позволяет получить список всех сгенерированных **токенов дружбы** для конкретного пользователя.

**Основные функции:**

1. **Эндпоинт**:
   * GET /getFriendTokens — возвращает массив токенов дружбы.
2. **Входные параметры**:
   * user\_id (обязательный, в query): уникальный идентификатор пользователя, чьи токены нужно получить.
3. **Выходные данные**:
   * Успешный ответ (200):
     + success: успешность операции (1 — успешно, 0 — ошибка).
     + tokens: массив объектов токенов, каждый из которых содержит:
       - invite\_token: уникальный токен приглашения.
       - invite\_limit: лимит использования (если задан).
       - invite\_duration: срок действия токена (в секундах, если задан).
       - time\_created: время создания токена (UNIX).
       - valid: статус токена (1 — действителен, 0 — нет).
4. **Ошибки**:
   * 400: Некорректный запрос, возвращает описание ошибки.
   * 500: Ошибка сервера, возвращает описание проблемы.

**Пример использования:**

Клиент запрашивает список токенов для определенного пользователя, чтобы узнать их статус, срок действия или лимиты.

1. **Friend Search**

Эта API выполняет **поиск пользователей Steam** по заданному тексту имени или его части.

**Основные функции:**

1. **Эндпоинт**:
   * GET /friendsearch — ищет пользователей, соответствующих тексту поиска.
2. **Входные параметры**:
   * search\_text (обязательный): текст для поиска (например, имя или его часть).
   * search\_filter (необязательный): фильтр для ограничения поиска (например, только "users").
   * search\_page (необязательный): номер страницы для просмотра результатов.
3. **Выходные данные** (при успешном ответе 200):
   * success: успешность выполнения (1 — успешно).
   * search\_text: текст, использованный для поиска.
   * search\_result\_count: количество найденных пользователей.
   * search\_filter: применённый фильтр поиска.
   * search\_page: текущая страница результатов.
   * html: HTML-код отображения результата (например, сообщение об отсутствии совпадений).
4. **Ошибки**:
   * 400: Неправильные параметры запроса.
   * 500: Ошибка сервера.

**Пример использования:**

Пользователь вводит часть имени в поле поиска, API возвращает список соответствующих аккаунтов или сообщение о том, что совпадений не найдено.

1. **Community Search**

Эта API предназначена для **поиска сообществ (групп) Steam** по текстовому запросу.

**Основные функции:**

1. **Эндпоинт**:
   * GET /communitysearch — выполняет поиск групп, соответствующих указанному тексту.
2. **Входные параметры**:
   * search\_text (обязательный): текстовый запрос для поиска сообщества.
   * search\_filter (необязательный): фильтр поиска (например, "groups").
   * search\_page (необязательный): номер страницы для просмотра результатов.
3. **Выходные данные** (при успешном ответе 200):
   * success: успешность выполнения (1 — успешно).
   * search\_text: текст запроса.
   * search\_result\_count: количество найденных сообществ.
   * search\_filter: применённый фильтр.
   * search\_page: текущая страница результатов.
   * html: HTML-код с результатами поиска, включая аватар группы, ссылку на страницу сообщества, тип группы и дополнительную информацию (например, описание или количество участников).
4. **Ошибки**:
   * 400: Неправильные параметры запроса.
   * 500: Ошибка сервера.

**Пример использования:**

Если пользователь вводит текст, например, "kuren1e", API возвращает список групп с их аватарами, типом группы (открытая/закрытая), количеством участников, а также ссылкой на страницу сообщества. Если совпадений нет, отображается соответствующее сообщение.

1. **WebSocket Server Discovery API**

Эта API предоставляет **список доступных WebSocket серверов Steam** с детальной информацией о каждом сервере.

**Основные функции:**

1. **Эндпоинт**:
   * GET /v1/ — возвращает информацию о доступных WebSocket серверах.
2. **Входные параметры**:
   * cellid (обязательный): идентификатор ячейки сети Steam, который определяет конкретный сервер.
   * cmtype (обязательный): тип соединения, всегда websockets для WebSocket.
   * origin (обязательный): источник запроса (обычно URL-адрес Steam, например, "[https://steamcommunity.com](https://steamcommunity.com/)").
3. **Выходные данные** (при успешном ответе 200):
   * serverlist: массив серверов с характеристиками:
     + endpoint: основной адрес сервера.
     + legacy\_endpoint: альтернативный адрес сервера.
     + type: тип соединения (например, WebSocket).
     + dc: дата-центр, где размещён сервер.
     + realm: область сервера (например, "steamglobal").
     + load: текущая нагрузка на сервер (в процентах).
     + wtd\_load: усреднённая взвешенная нагрузка на сервер.
   * success: статус успешности выполнения запроса.
   * message: дополнительное сообщение (обычно пустое).
4. **Ошибки**:
   * 400: Неправильные параметры запроса.
   * 500: Ошибка сервера.

**Пример использования:**

Если запрос отправляется с параметрами cellid=0, cmtype=websockets, origin=https://steamcommunity.com, API возвращает список WebSocket серверов Steam, их адреса, нагрузку и дата-центр. Это может использоваться для балансировки нагрузки и выбора наиболее подходящего сервера.

1. **MiniProfile**

Эта API предоставляет **информацию о мини-профиле пользователя Steam**, включая уровень, аватар, любимый значок и фон профиля.

**Основные функции:**

1. **Эндпоинт**:
   * GET /miniprofile/{profile\_id}/json/ — возвращает данные мини-профиля для указанного пользователя.
2. **Входные параметры**:
   * profile\_id (обязательный, в path): уникальный идентификатор профиля пользователя.
   * origin (обязательный, в query): источник запроса (обычно URL-адрес Steam, например, "[https://steamcommunity.com](https://steamcommunity.com/)").
3. **Выходные данные** (при успешном ответе 200):
   * level: уровень пользователя.
   * level\_class: CSS-класс для визуального отображения уровня.
   * avatar\_url: URL аватара пользователя.
   * persona\_name: имя пользователя в Steam.
   * favorite\_badge: информация о любимом значке пользователя, включая:
     + название, уровень, количество опыта, описание, URL иконки.
   * profile\_background: информация о фоновом видео профиля (в форматах WebM и MP4).
4. **Ошибки**:
   * 400: Неправильные параметры запроса.
   * 404: Профиль с указанным ID не найден.
   * 500: Внутренняя ошибка сервера.

**Пример использования:**

При запросе /miniprofile/1276937048/json/?origin=https://steamcommunity.com API возвращает данные о профиле пользователя с ID 1276937048, такие как уровень, аватар и любимый значок. Эта информация может использоваться для отображения мини-профиля в интерфейсе Steam или на сторонних платформах.

1. **Chat Emoji API**

Этот API предоставляет информацию о смайликах, используемых в чате Steam. Через запросы можно получить данные о конкретном смайлике, включая его кодировку (например, изображение в формате Base64) и ссылку на изображение. В запросе необходимо указать:

* **appid**: идентификатор приложения (обязательный параметр).
* **app\_name**: название приложения, например, "Steam" (обязательный параметр).

Ответ API включает:

* **appid**: идентификатор приложения.
* **app\_name**: название приложения.
* **url**: данные смайлика (например, изображение в формате Base64 или ссылка на картинку).

API поддерживает ответы с кодами 200 (успех), 400 (неправильные параметры), 404 (не найдено) и 500 (внутренняя ошибка).

Базовый URL для запросов: https://steamcommunity.com.