**Процесс работы со Steam API для будущих разработчиков**

**1. Получение данных о цене игры**

**Метод:** get\_steam\_game\_price(app\_id: str) -> Optional[Dict]

**Что делает:**

Этот метод отправляет запрос к Steam API, чтобы получить информацию о цене и скидках для указанного app\_id (идентификатора игры в Steam).

**Как работает:**

1) Проверяет, что app\_id корректен (состоит из цифр).

2) Соблюдает лимит запросов к Steam API (не чаще чем раз в STEAM\_REQUEST\_DELAY секунд, чтобы избежать блокировки).

3) Отправляет GET-запрос по URL вида: https://store.steampowered.com/api/appdetails?appids={app\_id}&cc=ru

\*Параметр cc=ru указывает регион (Россия), чтобы получить цены в рублях

4) Обрабатывает ответ:

Если API возвращает success=False, логирует ошибку и возвращает None.

Если у игры нет данных о цене (например, бесплатная игра), возвращает None.

В случае успеха возвращает словарь с данными:

{

"original": 1999.0, # Исходная цена (в рублях)

"final": 999.0, # Цена со скидкой (в рублях)

"discount": 50, # Процент скидки

"currency": "RUB" # Валюта

}

**Ошибки:**

Логирует сетевые ошибки или проблемы с API.

Возвращает None при любых ошибках, чтобы бот мог продолжить работу.

**2. Извлечение app\_id из URL игры**

**Метод:** extract\_app\_id(game\_url: str) -> Optional[str]

**Что делает:**

Извлекает идентификатор игры (app\_id) из её URL в Steam.

**Как работает:**

Ищет в URL подстроку app/ и берёт следующий за ней идентификатор.

Например, для URL https://store.steampowered.com/app/570/Dota\_2/ вернёт "570".

Если app\_id не найден или не состоит из цифр, возвращает None.

**3. Получение названия игры из URL**

**Метод:** get\_game\_name\_from\_url(game\_url: str) -> str

**Что делает:**

Извлекает название игры из её URL, чтобы использовать его в уведомлениях.

**Как работает:**

Берёт последнюю часть URL (например, Dota\_2 из <https://store.steampowered.com/app/570/Dota_2/>).

Заменяет символы подчёркивания на пробелы (Dota 2).

Удаляет цифры в конце (например, Half Life\_2 → Half Life).

Если URL некорректный, возвращает "Неизвестная игра".

**4. Интеграция с ботом**

**Где используется:**

1) Подписка на игру:

Пользователь отправляет боту ссылку на игру.

Бот извлекает app\_id и название, сохраняет их в базу данных.

2) Проверка скидок:

Каждый час бот проверяет скидки для всех игр из базы данных, используя get\_steam\_game\_price.

Если скидка достигла порога, установленного пользователем, отправляет уведомление.

3) Команда /mysubs:

Показывает список подписок пользователя с текущими ценами и скидками.

**5. Важные моменты**

**Лимиты запросов:**

Steam API имеет ограничения на частоту запросов, поэтому, чтобы избежать блокировки:

Между запросами добавляется задержка (STEAM\_REQUEST\_DELAY = 1.1 секунды).

Используется глобальная переменная LAST\_STEAM\_REQUEST\_TIME для отслеживания времени последнего запроса.

**Обработка ошибок:**

Все ошибки (сетевые, некорректные данные) логируются, и бот продолжает работу.

Если API не возвращает данные, бот не паникует — просто пропускает игру до следующей проверки.

**6. Пример запроса Steam API**

import requests

response = requests.get("https://store.steampowered.com/api/appdetails?appids=570&cc=ru")

data = response.json()

print(data["570"]["data"]["price\_overview"])

**Вывод:**

{

"currency": "RUB",

"initial": 1999,

"final": 999,

"discount\_percent": 50,

"initial\_formatted": "1999 руб.",

"final\_formatted": "999 руб."

}