

1. Accessing the Administration Panel

The administration panel allows you to manage the website-specific configurations (XPaths, scraping options, etc.) that the Telegram bot uses.

Login Details

- **URL:** http://86.104.73.3
- **Username:** admin
- **Password:** admin123

Key Functionality

1. **Dashboard:** Lists all currently configured websites. This is where you'll see a record for each website the bot knows how to scrape.
 2. **Add/Edit Configuration:**
 - Click "Add New" or "Edit" next to an existing configuration.
 - You'll need two main inputs:
 - **Website:** A unique substring of the URL (e.g., realestate.com). The bot will match any URL containing this string to use this configuration.
 - **Config JSON:** A JSON object containing all the scraping rules (XPaths for title, price, photos, next page, etc.). This is the **most crucial part** and requires technical knowledge of the target website's HTML structure.
-

2. Using the Telegram Bot

The Telegram bot is the user-facing interface for initiating the web scraping process.

Finding the Bot

You'll need to find the bot using its username on Telegram. Since the code uses a BOT_TOKEN from an environment variable, you must know the bot's username (e.g., @YourBotName) to interact with it.

Commands

Command	Description
/start	Greets you and prompts for a URL.
Sending a URL	Initiates the scraping process for the provided URL.

Scraping Process

1. Send the URL: Send a single URL in a message to the bot.

- The bot first checks if the URL matches a configuration in the admin panel. If not, it will reply with: ❌ Источник не подключён. Добавьте в панели. (Source is not connected. Add it in the panel.)
- If a config is found, the bot replies: ✅ Принято, собираю... (Accepted, collecting...)

2. Scraping Logic:

- The bot first attempts to scrape the URL as a **single property page** (detailed view).
- If the single-property scrape fails (e.g., the "Название" / Title field is not found or is 'ERROR'), it then attempts to scrape the URL as a **list/search results page**. The list page parser will follow pagination rules (either by clicking a **Next button** or by incrementing a **page query** in the URL) and scrape all property links it finds until it reaches the last page.

3. Result:

- If no properties are found, it replies: ❌ Недвижимость не найдена. (Property not found.)
- Upon completion, it replies with the result and a link to download the Excel file.
 - **Example Reply:** ✅ Готово: 5 объекта 1 с ошибками. Скачать Excel: 📁 http://86.104.73.3/output_file/2025-10-23_12-02-46_properties.xlsx
 - The link will point to the specific file on your hosted server.

3. Understanding the Scraped Data

The bot exports the data into an Excel file with a fixed, Russian-language column order.

Excel Column Order (Desired Order in Code)

1. Ссылка на объект (Object Link)
2. Название (Title)
3. Цена (Price)
4. Валюта (Currency)
5. Площадь (Area)
6. Площадь земли (Land Area)
7. Тип объекта (Object Type)
8. Год постройки (Year Built)
9. Количество комнат (Number of Rooms)
10. Описание (Description)
11. Инфраструктура (Infrastructure)
12. С/у (Bathrooms)
13. Этаж (Floor)
14. Локация (Location)
15. Координаты (Coordinates)
16. **Фото_ссылки** (Photo Links) - *Links to the downloaded images on your server (e.g., <http://86.104.73.3/images/...>).*
17. **Фото_уникальные_названия** (Unique Photo Names) - *The unique filenames for the downloaded images.*
18. Контактное лицо (Contact Person)
19. Телефон контактного лица (Contact Phone)
20. Компания (Company)
21. Телефон компании (Company Phone)

Error Indication

Any cell in the Excel file that couldn't be scraped will contain the text **ERROR** and will be **highlighted in red** (using a red_fill). The final bot message will also report the total number of errors.

4. Troubleshooting and Configuration Tips

General Errors

- **✗ Источник не подключён.:** The URL you provided doesn't match any website entry in the admin panel. Add a new configuration.
- **Empty Excel or ERROR in all fields:** The XPath expressions in your JSON configuration are likely incorrect or the website uses anti-bot measures that the bot's current settings aren't overcoming.

Configuration Tips

- **Website Field:** Use the root domain or a unique, short part of the URL. E.g., for `https://www.example.com/property/123`, use `example.com`.
- **XPath Accuracy:** XPaths must be exact. Use browser developer tools to find robust XPaths.
- **Selenium vs. Requests:** The bot defaults to **Selenium** (`get_rendered_html`) which handles dynamic, JavaScript-rendered content. If the website is static, you can use the requests fallback, but for modern sites, Selenium is necessary.
- **Photo Fields:** The `Фото_ссылки` field expects an XPath that returns a list of image URLs. The bot automatically downloads these images, stores them in the local images folder, and updates the Excel links to point to your hosted image path (`http://86.104.73.3/images/...`).
- **List Page Configuration:**
 - `"list_page_check"`: XPath for all individual property links on the list page.
 - **Pagination Method:**
 - If the page uses a **Next Button**, provide the XPath to the button in `"next_page_xpath"`.
 - If the page uses a **URL Query** (e.g., `?page=2`), provide the query parameter name (e.g., `page` or `p`) in `"page_query"`.
- **Transform Field:** The transform field in the JSON allows you to perform basic data cleaning/modification using Python's `eval` function on the scraped value. For example, to extract only numbers from a price string: `"transform": "re.sub(r'^0-9.', '', value)"`. **Be cautious** with this field as it executes arbitrary code.