

# Puppeteer Web Scraping AliExpress Products

## Introduction

This documentation provides a guide on how to use the Puppeteer web scraping project to extract product information from AliExpress. The project utilizes Puppeteer, Puppeteer Extra, and Stealth Plugin to navigate through AliExpress search results, scrape product details from multiple pages, and handle pagination dynamically.

## Installation

To use this project, you need to have Node.js installed on your machine. You can install the required dependencies using npm or yarn:

```
npm install puppeteer puppeteer-extra puppeteer-extra-plugin-stealth
```

## Usage

To use the web scraping functionality, simply call the **scrapeAliExpress()** function:

```
scrapeAliExpress();
```

## Features

- Automatically handles navigation and pagination on AliExpress search results.
- Scrapes product names from multiple pages.
- Utilizes Puppeteer Extra with the Stealth Plugin for stealthy scraping.

## Code Overview

The code consists of the following main components:

1. **Initialization:** Initializes Puppeteer, Puppeteer Extra, and Stealth Plugin.
2. **Navigating to AliExpress:** Opens the AliExpress search URL and starts scraping.

3. **Scraping Products:** Extracts product names from the current page.
4. **Handling Pagination:** Navigates to the next page and repeats the scraping process until no more pages are available.
5. **Auto Scrolling:** Scrolls the page to load all products dynamically.

## Implementation Details

### Initialization

```
const puppeteerExtra = require("puppeteer-extra");
const stealthPlugin = require("puppeteer-extra-plugin-stealth");

puppeteerExtra.use(stealthPlugin());
```

This section initializes Puppeteer Extra with the Stealth Plugin to ensure that the web scraping activities are undetectable.

### Navigating to AliExpress

```
const browser = await puppeteerExtra.launch({
  headless: false,
  executablePath: "", // Add your path here
});

const page = await browser.newPage();

const aliExpressSearchURL = "https://fr.aliexpress.com/w/wholesale-Machines-
Café.html?spm=a2g0o.best.search_ranking.7.4e2b142dNKaRF1";
```

Launches a headless browser instance, opens a new page, and defines the AliExpress search URL.

### Scraping Products

```
async function scrapePage(page) {
  // Assuming the product names are within a certain class, adjust the selector accordingly
  const productNamesSelector = '.multi--titleText--nXeOvyr';

  await page.waitForSelector(productNamesSelector);
```

```
const productNames = await page.evaluate((selector) => {
  const productNodes = document.querySelectorAll(selector);
  return Array.from(productNodes, (node) => node.textContent.trim());
}, productNamesSelector);

console.log("Number of products:", productNames.length);
console.log("Product Names:", productNames);
}
```

Scrapes product names from the current page by selecting specific elements and extracting their text content.

## Handling Pagination

```
async function goToNextPage(page) {
  const nextButtonSelector = '.comet-pagination-next .comet-pagination-item-link';
  const nextPageButton = await page.$(nextButtonSelector);

  if (nextPageButton) {
    await Promise.all([
      nextPageButton.click(),
      page.waitForNavigation({ waitUntil: 'domcontentloaded', timeout: 60000 }), // Increased
      // timeout to 60 seconds
      page.waitForTimeout(5000) // Add a delay after clicking next button
    ]);
    return true;
  } else {
    console.log("No next button found. Reached the last page.");
    return false;
  }
}
```

Navigates to the next page of search results by clicking the next button and waits for the page to load.

## Auto Scrolling

```
async function autoScroll(page) {
  // Function definition
}
```

Defines a function to automatically scroll the page to load all products dynamically.

## **Conclusion**

The Puppeteer web scraping project for AliExpress offers a convenient way to extract product information from search results. By following the provided documentation, developers can easily integrate this functionality into their projects and customize it as needed.