# **Project: Crawl Node API V1 (for Cheng)**

Updated: May 30, 2025

# Objective

Build a production-ready Application Program Interface that can fully replace using a proxy for browsing ***any*** website – using a real machine, real browser, and residential IP address in the background in order to work around popular bot detection blocking services.

# Background

Websites are getting increasingly difficult to crawl due to getting [blocked](https://passimage.in/i/90e68a3f3051e58eef7b.png) by sophisticated bot detection services like DataDome, CloudFlare, and Incapsula. Users of our proxy services – even when using residential IPs – may not use our proxies carefully and then conclude: “these proxies don’t work – I still got blocked”.

To counter this, we can carefully develop a system to automate – via an API – requests within a proper setup with a real computer, real browser, and real IP address, encapsulating our expertise around web crawling on top of our proxy network to help developers and businesses effectively crawl any website.

# Technical Requirements (V1)

1. An API for browsing any website with human-like interactions: https://api1.crawlnode.com/v1/

* Input #1: Navigate to a URL (/go)
  + session: session id (default: start new browser session)
  + url: destination URL
  + timeout: the maximum number of seconds to wait (default: 60 seconds)
  + user: the HTTP basic authentication username (optional)
  + pass: the HTTP basic authentication password (optional)
* Input #2: Click on element (/click)
  + id: session ID
  + element: element selector
  + wait: 10s / page\_load / none
  + timeout: the maximum number of seconds to wait (default: 60 seconds)
* Input #3: Type keystrokes into element (/type)
  + id: session ID
  + element: element selector (or none)
  + keystrokes: sequence of [keystrokes](https://www.autohotkey.com/docs/v1/KeyList.htm) to type (e.g., david@vinaudit.com{TAB}12346{ENTER} )
  + wait: 10s / page\_load / none
  + timeout: the maximum number of seconds to wait (default: 60 seconds)
* Input #4: Download response content (/download)
  + session: session ID
  + response: response ID

* Output:
  + session: Session ID
  + url: Main url requested
  + status\_code: Main page HTTP status code
  + response\_headers: List of response headers
  + html: Raw HTML content for page.
  + elements: Associative array of interactive elements on page:
    1. element\_id: Element ID
    2. name: Element Name
    3. class: Element Class
    4. value: Element Value
  + responses: List of all response URLs for sub-URLs requested (e.g., images, AJAX, etc.)
    1. response\_id
    2. url
    3. content\_type
    4. status\_code
    5. size

2. A dashboard (dashboard-cheng1.crawlnode.com) for configuring and monitoring a distributed system:

* Display status of each server + browser:
  + Server: n1.crawlnode.com (Online)
    - Browser #1: (Session ID# 5837393 | loading: https://google.com)
    - Browser #2: (Session ID# 4328932 | loading: https://www.amazon.com/dp/B002DYJ02U)
    - Browser #3: (ready)
    - Browser #4: (ready)
    - Browser #5: (ready)
  + Server: n2.crawlnode.com (Online)
* Support add, edit server (hostname, username, password, max\_browser\_count)
* Support viewing each logged request (URL, response time, status, code) from central dashboard

3. Agent for distributed crawling via real browsers with undetectable human-like interaction

* New crawl requested are distributed (via a distributed queue) to an available server and browser then linked via Session ID
* Server: (To be supplied)
  + OS: Windows-based Operating System
  + Browser: Real Chrome Browser (not Selenium or other detectable version of Chrome)
  + Support a configurable number of instances of the browser, each running an independent profile (Profile: Guest 1, Guest 2, etc.) with independent cookies.
  + Support for linking a specific browser via Session ID (for long-running sessions)
  + Support for configuring different proxy (to be supplied) for each session
  + Support for monitoring and downloading all URL content (including images and ajax requests)

# Milestones

* **Milestone V0.5**:
  + Crawl API can ***sequentially*** crawl all pages of 5 dealer websites using 1 browser instance on 1 physical server:
    - Website List: thruwaynissan.com, motorwerkshonda.com, capitalfordrockymount.com, cityautotrucks.com, kirklandhonda.com
* **Milestone V1.0**:
  + Crawl API can ***simultaneously*** crawl all pages of 15 dealer websites using 15 browser instances on 3 physical servers added and monitored via dashboard:
    - Website List: thruwaynissan.com, motorwerkshonda.com, capitalfordrockymount.com, cityautotrucks.com, kirklandhonda.com, 1autoliquidators.com, 0to60motorsportsct.com, 039autosale.com, 1autodmv.com, 1oakautos.com, 1ownerautosales.net, my1stopauto.com, 1800autoapproved.com, 1easyrideautosale.com, 1uniquemotorsllc.com

# Deliverables

* An approved design document outlining the technology stack, databases, data schemas, and data flow for this system.
* Cleaned, well-organized, object-oriented source code (to be checked into private Github repository) for: API Server, Dashboard Server, 3x Crawl Servers
* Server setup documentation with step-by-step instructions for setting the initial environment for each machine in this system.
* Test cases confirming robust crawling:
  + Test script for requesting top 100 domains: https://radar.cloudflare.com/domains
  + Test script for requesting 60,000 known dealer websites (to detect which websites are active)
  + Test script for crawling all known links on given dealer websites:
    - thruwaynissan.com, motorwerkshonda.com, capitalfordrockymount.com, cityautotrucks.com, kirklandhonda.com, 1autoliquidators.com, 0to60motorsportsct.com, 039autosale.com, 1autodmv.com, 1oakautos.com, 1ownerautosales.net, my1stopauto.com, 1800autoapproved.com, 1easyrideautosale.com, 1uniquemotorsllc.com

# Resources (to be provided)

|  |  |
| --- | --- |
| API Server | Server: api-cheng1.crawlnode.com OS: AlmaLinux 8 Disk: 200GB SSD RAM: 8 GB  Credentials: (to be provided) |
| Dashboard | Server: dashboard-cheng1.crawlnode.com OS: AlmaLinux 8 Disk: 100GB SSD RAM: 4 GB  Credentials: (to be provided) |
| 3x Crawl Servers | Servers: crawl-m1.crawlnode.com, crawl-m2.crawlnode.com, ... OS: Windows 11 Disk: 50GB SSD RAM: 12 GB  Credentials: (to be provided) |
| GitHub Repo | (to be provided) |