Web Robots

Features

2016-10-14

Table of Contents

Web Robots Platform

The following is a general list of features of the Web Robots product. In general, Web Robots full product, technical documentation, tutorials, and reference guides are available at https://support.webrobots.io These guides are all indexable, bookmarkable, and searchable. In general, Web Robots software stack runs on cloud as a service (SaaS) and is accessible through Portal at http://portal.webrobots.io

Web Robots Chrome Extension

Web Robots Chrome Extension is the Web Robots application for creating and debugging robots. It is covered by the Developer license. Chrome Extension is an integrated development environment (IDE) for robots. This means that Chrome Extension is all you need for programming robots in an Javascript programming language. To support you in the construction of robots, Chrome Extension provides you with powerful programming features including an environment with syntax highlighting, full debugging capabilities, an overview of the program state.

Web Robots Portal

Web Robots Portal (http://portal.webrobots.io) is a website where Clients can view their robots, robot runs, download data, check running status, detailed run logs, create new robots, configure and schedule robots, check usage statistics.

- Robots can be scheduled to run with whatever frequency desired from a web-based scheduler
- Detailed information about the robots run (execution of every ste, success or failure, number of elements returned) can be logged into a database optionally
- The “ Run Logs” view in the Portal is available to allow users to trace all steps of robot execution
- The “Robots” view lists all robots belonging to a client, showing robot's latest run date, duration and status.
- The “Robot Details” view shows robot's run history – date, duration and status of all runs beloning to a robot.
- The “Robot Edit” view allows user to configure robot's running schedule, quality assurance parameters (acceptable running duration, acceptable amount of data), logging level and other parameters.
- With the Scheduler, Robot runs can be scheduled for any running repetition pattern.
- Robot execution and the status can be monitored on Portal as normal website (for human users) and JSON view (for API integration).
- The “Client” view provides detailed statistics about Cloud usage, active users, hours, etc.

Web Robots Cloud

Web Robots Cloud is a cloud enviroment for running robots as a service to clients. Web Robots Cloud is always listening for requests from clients, such as robot execution requests, and executes robots. A client communicating with Portal which is an integrated part of a system. Web Robots Cloud is built on multiple infrastructure cloud environment to assure resilience and fault tolerance.

- Clients (the API, etc) can put robots in Cloud execution que.
- Cloud Workers are autoscaled—instances are added and removed seamlesly.
- Cloud Workers can concurrently execute robots in multiple threads (many robots can be run at once).
- The same robot can be executed in parallel with different tasks (for example, performing same task on different websites).
- The same robot can be forked into many parallel running child-robots for a quicker run execution.
- Cloud Workers report all statuses of execution to Portal in real time.

Web Robots Functionality

Robot Data Extraction

Robots can be written/configured to extract data from a variety of sources and if desired to combine data from multiple sources.

- Robots can extract from the web (intranet, extranet, or internet)
- Robots can extract from sites that use JavaScript
- Robots can execute JavaScript whether this JavaScript has been written from scratch (by the robot builder) or exists within loaded web content
- Robots can extract from sites that use AJAX (event-driven JavaScript)
- Robots can extract from sites that use Flash (techniques to use may vary)
- Robots can extract from delimited text files (like CSV)
- Robots can extract from and work with popup windows
- Robots can use n numbers of variables (including variables that are global to an entire robot) of differing types to hold values for the life of a robot run
- Robots can extract from sites that use form-based authentication
- Robots can extract from sites with BASIC authentication or require client or server-side certificates for authentication
- Robots can extract from sites that use SSL or TLS encryption as easily and transparently as they do with sites that use no encryption
- Robots can call SOAP and/or REST-based web-services and consume the resultant JSON or HML
- Robots can create and extract from existing cookies
- Robots can extract data from websites in multiple locales including those with languages that use double-byte character sets
- Robots can turn JavaScript-only URLs into absolute URLs (by executing all code to get the result) automatically without actually traversing the link
- Robots can use a lot of different logic to find elements on a page including placeholders, tag patterns, tag hierarchies, regular expressions, tag attributes, etc. Robots can look for elements either based on how their text appears to a user on a web page or their underlying HTML. Elements can be found relative to other tags on a page in any relation (inside, outside, before, after, in between) in multiple combinations of relationship logic. (e.g. Look for this tag after placeholder “A” and in between tags “B” and “C”). Using this kind of logic, robot builders can make robots fairly resilient to changes on the underlying

website

- Robots can send additional HTTP headers and store and receive headers and status codes
- Robots can perform raw and dynamic HTTP requests (GET, HEAD, OPTIONS, POST) and consume the responses to these requests
- Robots can extract text from tags (HTML or XML), the tags themselves, or between ranges of tags (including the borders of the ranges or not)
- Robots can dynamically loop through lists and pages
- Robots can manipulate the HTML DOM in any way before the HTML page is “executed” or “processed”: this includes removing, changing, hiding (or un-hiding), or adding elements
- Robots can edit and manipulate JavaScript on a page before a page is “executed”
- Robots can use multiple proxy servers to come from multiple different IP addresses when extracting data; robots can shift to new IP addresses at any time (or this can be done between robot runs)
- Robots can crawl pages on the web recursively with many options to govern the crawl (including de-duping pages)
- Robots can extract binary files and images
- Robots can extract the current URL from where it is in a session
- Robots can transpose and manipulate tables on a web page
- Robots can loop through and extract from XML coming from any source such as flat files, the result of web-service calls, etc.
- Robots can consume RSS feeds
- Robots can loop through and extract from JSON coming from any source including flat files or the web
- Chrome Extension allows for the creation of output data models that can have multiple typed attributes and object relationships (parent/child, one to many, etc). Robots can extract data into these models and create relationships among objects

Robot Data Transformation

Robots can transform and manipulate data in most any way that one could do so in a Javascript programming language. To make data transofrmations even more porwerful robots can use included libraries like Underscore, Moment and others. The options below can be combined: for example, one can extract HTML from a page, apply a regular expression to that HTML, convert the HTML to nicely formatted text, then convert that text to lower case.

- String extraction, formatting, and manipulation (A “string” in this case could come from any source of data Web Robots supports and could be for example something like text on a web page, HTML on a web page, JavaScript, etc.).
- Regular expressions to transform text
- Mapping values to lists (for example the two-character representation of a state to the full state name)
- Text replacement, pattern replacement, case transformation (upper, lower, proper), remove spaces, remove non-printable characters, build strings, substring, indexing, inserting or extracting special characters, string operators
- Robots can manipulate strings in multiple locales including double-byte character sets
- Customized string conversion
- Numeric Extraction, transformation, and formatting
- Arithmetic (virtually all mathematic operators are supported), rounding, MAX/MIN
- Numeric format pattern application (for example the ability to specify decimal and thousand separators, number of decimal places)

- Date/Time Extraction, transformation, calculation, and formatting
- Moment.js library is included in robots by default which is the “go to” library when working with Date/Time. It's toolset covers Date/Time recognition, validation, manipulation and display.
- Transform dates from websites in double-byte character sets in multiple locales and date formats into the desired date/time format (including a standard date/time format which might be stored in a database as a DATETIME type or come back into a programming language data structure as a “Date” type)
- Extract any date/time element out of a date (for example get the year, day, or hour out of a date)
- Work with 12 or 24 hour time
- Work taking the time zone into account of date/times that are extracted
- Do date calculations (get the time between dates or modify dates using date arithmetic) easily
- Get the current date, time, weekday, etc. at any point in a robot
- Format date/times in any way desired
- Extract date/times automatically from websites that display it in the form “3 hours ago” or “2 days ago” into standard date/time types
- HTML Extraction and transformation
- Anything listed above with regard to “strings” in general also applies to HTML
- Remove all types of a specific tag from HTML given a string of HTML
- Count the number of a particular tags in a string of HTML
- Convert a particular string of HTML into formatted text with many automatic formatting options including whether to include URLs, whether to include text alternatives to images, whether to include URLs, etc.
- String encoding and decoding
- Ampersand, URL, and Base64 encode/decode
- Conditionally do all of the above transformations with many options on the conditions—unlimited if-then/else conditions (for example if a string contains “xyz” then extract from it using regular expression “A” else extract from it using regular expression “B”)
- Robots can convert types (strings to dates and back for example)
- Robots can generate random numbers
- Robots can compute the MD5 checksum of some input
- Robots can generate globally unique Ids (GUIDs)
- Robots can take a relative URL and transform it into an absolute URL
- Robots can perform XSL transforms on XML given an XSL template and some XML

Robot Output

The following shows options for where robots can put data using Web Robots in addition to database and API’s:

- Robots can fill out forms in web applications as a human would do (except in an automated way) working with page input elements dynamically
- Robots can dynamically iterate through drop-downs and radio-button sets
- Robots can upload data from/to websites
- Robots can write data to Amazon S3 storage
- Default output formats are JSON and CSV which are always generated

Robot Configuration

These values are changeable globally within a robot or can be set differently at the “step” level.

- Robots can “spoof” their user-agent attributes including values for the browser used, locale, screen size, flash version, referred from URL, etc.
- Robots can be set to retry actions on a website multiple times if the page is timing out and be set to wait for prescribed periods of time for the page to come and in between attempts
- Robots can selectively ignore any errors encountered when loading a page (for example missing cripts or invalid JavaScript
- Robots can start using, change/rotate, stop using proxies
- Robots can work according robots.txt guidelines or override them
- Robots can track visited URLs and skip them if directed to crawl again
- Robots can be configured to split into parallel running robots at any “step” during run

Client Library

Client Library is a special library that is available to all robots owned by a customer. This library is used to store code, functions, building blocks which is reused across many robots.

- Client Library can be edited like a regular robot
- Library components can be shared within or between robots
- Changes to a library will have effect in all robots using the library

Robot Execution

The following shows execution features of robots in the Web Robots platform.

- Robots can be debugged within Chrome Extension including the use of single-stepping, breakpoints, etc.
- Errors can be handled and the robot can follow different logic paths depending on errors (for example if a site that is trying to be loaded is down or a tag on a page cannot be found using specified logic)
- Robots can follow different logic paths based on any condition one could imagine creating in a programming language including what is going on with any data that has been extracted from any source, the current state of a web page, etc. Multiple “AND” and “OR” conditions can be constructed to create complex decision points
- Robots created by Chrome Extension can be executed in Chrome Extension with full functionality
- Robots created by Chrome Extension can be executed on Web Robots Cloud without any additional adaptations
- Robots can pause for prescribed or random periods of time at any point in their execution to appear more humanlike

Web Robots API

- Robots API is built on Amazon's SQS, SNS and S3 services.
- Robots can be called from any programming environment. Amazon provides SDKs for all major programming languages and environments:

    - JavaScript
    - Java
    - .NET
    - Node.js
    - PHP
    - Python
    - Ruby
    - Go
    - C++
    - Android
    - iOS
- Robots automatically write data brought back from a robot execution to a database and generates JSON and CSV data files on S3.
- Robots can be passed input objects and attributes (of any number and type including binary types) and act on them in a dynamic way
- Robots can be executed synchronously or asynchronously
- Robots can return data with multiple objects, object attributes of various types, and object relationships of varying complexity
- Robots return data in JSON and CSV formats automatically, other formats can be custom configured

Web Robots High Availability and Load Distribution

Web Robots platforms ensures high availability in the production environment.

- Web Robots Cloud will automatically apply workload distribution by distributing robot runs to cloud workers
- Web Robots Cloud will automatically add cloud resources when workload is waiting in queue and automatically shut down resources that are idle.
- Workloads are distributed on at least two industry leading infrastructure cloud providers
- Cloud workers have self-healing features to recover from temporary failures and unexpected states
- All robots are backed-up daily to a 3rd party backup solution.
- Robot backups are versioned and versions from any date can be restored.