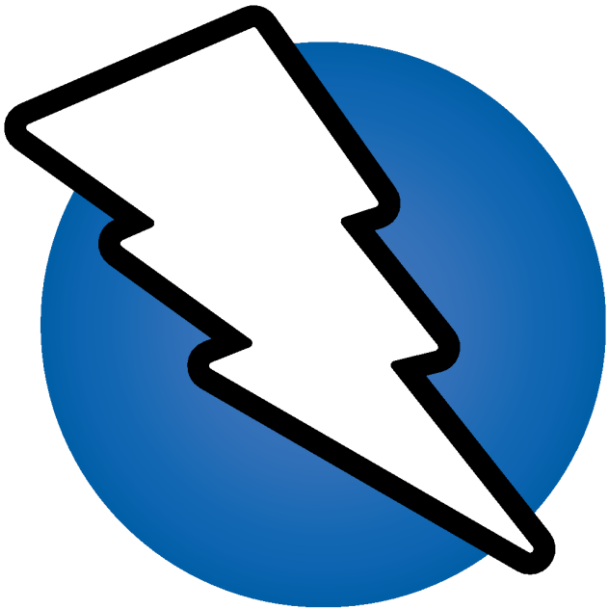
ZAP API Guide

Version 1.0



Contents

1. Overview

Welcome to the ZAP API.

ZAP provides a REST Application Programming Interface (API) which allows you to interact with ZAP programmatically.

The REST API can be accessed directly or via one of the client implementations detailed below.

It is documented briefly in the ZAP user guide

1.1. API Client Generation

The ZAP API clients are created via code generation - this makes them much easier to maintain.

Language	Download Links	Notes
Java	GitHub	Official API
Python	PyPI	Official API
Node.js	NPM	In process of becoming an official API
PHP	GitHub Packagist	In process of becoming an official API
Ruby	GitHub	

2. Getting Started

In order to be able to use the API when using the ZAP UI you have to first enable it.

1. Select **Tools > Options** to open the **Options** window
2. From the list on the left, select **API**
3. Check the **Enabled** box
4. Check the **UI Enabled** box
5. Copy the **API Key** for use later

> *If you run ZAP in 'headless' mode via the command line or 'daemon' mode using the -daemon flag then the API will be automatically enabled.*

2.1. Configure Zap To Run Locally

Open your preferred browser and set up the proxy. The browser and ZAP need to have the same proxy settings.

2.1.1. BROWSER CONFIGURATION:

In Chrome to do the following:

1. Open the Chrome menu and select '**Settings**'

2. Select '**Advanced**'

3. Scroll down to '**System**' and select **'Open proxy settings'**

4. Ensure the '**Connections**' tab is selected and click **'LAN Settings'**

5. Select **'Use a proxy server for your LAN'**

6. Enter the '**Address**' e.g. localhost 127.0.0.1 and **'Port**' number e.g. 8080

8. Click '**OK**' to close the **'LAN Settings'** dialog box

9. Click '**OK'** to close the Internet Properties dialog box

2.1.2. CONFIGURING ZAP

1. Start ZAP and select **Tools** > **Options** > **Local Proxy**

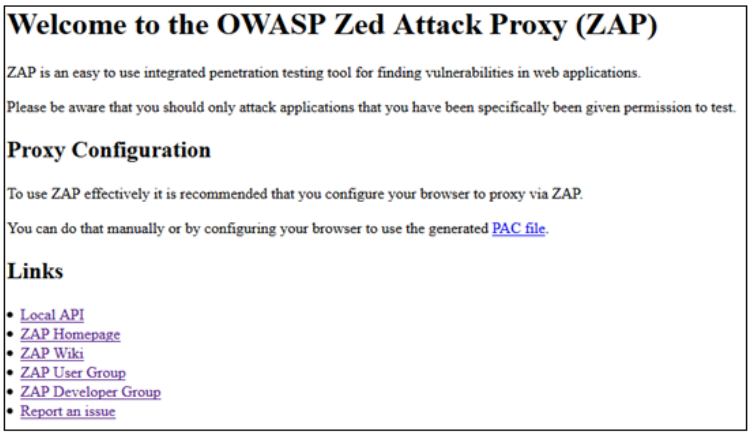
Make sure the port is set to 8080 (or the port you have configured in your browser)

2. Open any website using SSL in your browser and make sure the site shows up in the **sites list**

If ZAP runs on localhost port 8080

Go to your browser and open http://localhost:8080

You should see the following page:



Welcome to the OWASP Zed Attack Proxy (ZAP)

ZAP is an easy to use integrated penetration testing tool for finding vulnerabilities in web applications.

Please be aware that you should only attack applications that you have been specifically been given permission to test.

Proxy Configuration

To use ZAP effectively it is recommended that you configure your browser to proxy via ZAP.

You can do that manually or by configuring your browser to use the generated PAC file.

Links

- Local API
- ZAP Homepage
- ZAP Wiki
- ZAP User Group
- ZAP Developer Group
- Report an issue

3. The Zap API UI

From the '**Welcome** page', select '**Local API'** to view a list of all the functionalities exposed via the ZAP API.

These are known as **Components.**

Selecting a component reveals API related '**Views**' and '**Actions**'.

VIEWS : return information

ACTIONS : control ZAP

The API is available in **JSON**, **XML** and **HTML** formats.

The ZAP API UI is one way to interact with the ZAP API.

3.1. Standard API URLS

It is useful to copy API URLS into a text editor for later testing.

The API URLs are of the form:

http://zap/<format>/<component>/<operation>/<operation name>[/?<parameters>]

The format can be 'JSON', 'XML' or 'HTML'

example:
http://localhost:8080/JSON/spider/action/scan/?zapapiformat=JSON&Method=GET&url=http%3A%2F%2Fwebscantest.com&maxChildren=1&recurse=&contextName=&subtreeOnly=

In the above example, the following applies:

Format	JSON
Component	spider
Operation	Action
Action	Scan
Parameters	url (http://www.webscantest.com)
	maxChildren (1)
	recurse context (not set)
	Name (not set)
	subtreeOnly (not set)

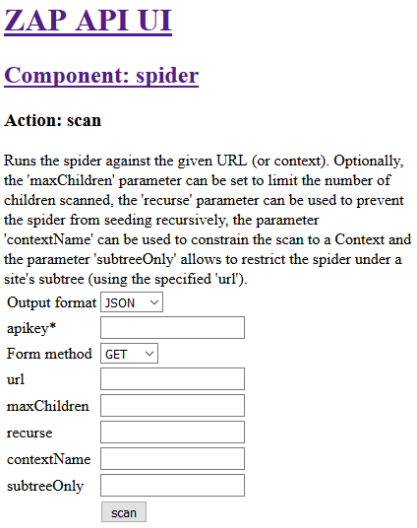
4. How to Perform Tasks

The following are examples of tasks that can be performed via the ZAP API UI.

4.1. Perform A Spider Scan

To perform a spider on an application that is running locally, do the following:

1. Open the ZAP API UI here: http://localhost:8080
2. Select **'Local API'** to view the list of components
3. Select **'spider'**
4. From the list of **'Actions'**, select '**scan** (url maxChildren recurse contextName subtreeOnly)' to open the following dialog:



5. Paste the **apikey** that you copied earlier. (If you no longer have it go to ZAP > Tools > Options and select API on the left)
6. Enter the URL required e.g. http://webscantest.com
7. Assign values for the other fields as required.
8. Click **'scan'** to perform the scan, you will see a confirmation such as "scan=1"

4.2. View The Status Of A Spider Scan

1. From the **spider** component, scroll down the list of **'Views'**
2. Select **status(scanID)**
3. Paste the **apikey**. Enter a scan ID if required. You will see the following message if the scan is completed: **"status":"100"**

4.3. View The Results Of A Spider Scan

1. From the **spider** component, scroll down the list of **'Views'**
2. Select **results(scanID)**
3. Paste the **apikey**. Enter a scan ID if required. You will see the results of your scan.

5. ZAP API Functions

All components of ZAP can be interacted with as required using the basic ingredients of **Components, Views and Actions.**

5.1. Components

The following components are able to be interacted with via the ZAP API.

Component	Description
acsrf	anti CSRF tokens to protect against-Cross Site Request Forgery (CSRF) attacks.
ajaxSpider	AJAX Spider uses crawljax to crawl AJAX rich sites.
alertFilter	Alert Filters allow you to automatically override the risk levels of any alerts raised by the active and passive scan rules within a context.
ascan	Active scanning attempts to find potential vulnerabilities by using known attacks against the selected targets.
authentication	Handles multiple types of authentication for websites / webapps
authorization	authorization detection method set for a context.
autoupdate	Automatic updates
break	Manages breakpoints during interception
context	Manages urls to include or exclude
core	Core ZAP features
forcedUser	Forced Browsing user settings
httpSessions	keeps track of the existing HTTP Sessions
importurls	import a file of URLs
keyboard	configure keyboard shortcuts
localProxies	configure the addresses and ports on which ZAP accepts incoming connections.
params	parameters for a site or all sites
pnh	Plug-n-Hack allows you to monitor client (browser) events in order to help test HTML5 applications
pscan	Configure passive scanner
quickstartlaunch	Perform a quick scan
replacer	replace strings in requests and responses.
reveal	show hidden fields and enable disabled fields
ruleConfig	configure the behaviour of specific active and passive scan rules.
script	run scripts that can be embedded within ZAP and can access internal ZAP data structures.
search	search for regular expressions in all of the URLs, requests, responses, headers
selenium	set and view the paths to the required WebDrivers and binary.
sessionManagement	handles multiple types of session management (called Session Management Methods) that can be used for websites / webapps.
spider	automatically discover new resources (URLs) on a particular Site.

Component	Description
stats	access to the stats now maintained by ZAP.
users	representations of websites/webapps' users. They allow certain actions to be performed from the point of view of an user of the webapps
websocket	can be used by web applications or web sites to setup a bi-directional (two-way), full duplex communication channel over a single TCP connection.

5.2. Views & Actions

The following views and actions are possible for the above listed components.

Component	Name	Type	Parameters	Description
acsrf	optionTokensNames	view		Lists the names of all anti-CSRF tokens
acsrf	addOptionToken	action	String*	Adds an anti-CSRF token with the given name, enabled by default
acsrf	removeOptionToken	action	String*	Removes the anti-CSRF token with the given name
acsrf	genForm	other	hrefId*	Generate a form for testing lack of anti-CSRF tokens - typically invoked via ZAP
pscan	scanOnlyInScope	view		Tells whether or not the passive scan should be performed only on messages that are in scope.
pscan	recordsToScan	view		The number of records the passive scanner still has to scan
pscan	scanners	view		Lists all passive scanners with its ID, name, enabled state and alert threshold.
pscan	setEnabled	action	enabled*	Sets whether or not the passive scanning is enabled (Note: the enabled state is not persisted).
pscan	setScanOnlyInScope	action	onlyInScope*	Sets whether or not the passive scan should be performed only on messages that are in scope.
pscan	enableAllScanners	action		Enables all passive scanners
pscan	disableAllScanners	action		Disables all passive scanners
pscan	enableScanners	action	ids*	Enables all passive scanners with the given IDs (comma separated list of IDs)
pscan	disableScanners	action	ids*	Disables all passive scanners with the given IDs (comma separated list of IDs)

Component	Name	Type	Parameters	Description
pscan	setScannerAlertThreshold	action	id* alertThreshold*	Sets the alert threshold of the passive scanner with the given ID, accepted values for alert threshold: OFF, DEFAULT, LOW, MEDIUM and HIGH
search	urlsByUrlRegex	view	regex* baseurl start count	
search	urlsByRequestRegex	view	regex* baseurl start count	
search	urlsByResponseRegex	view	regex* baseurl start count	
search	urlsByHeaderRegex	view	regex* baseurl start count	
search	messagesByUrlRegex	view	regex* baseurl start count	
search	messagesByRequestRegex	view	regex* baseurl start count	
search	messagesByResponseRegex	view	regex* baseurl start count	
search	messagesByHeaderRegex	view	regex* baseurl start count	
search	harByUrlRegex	other	regex* baseurl start count	
search	harByRequestRegex	other	regex* baseurl start count	
search	harByResponseRegex	other	regex* baseurl start count	
search	harByHeaderRegex	other	regex* baseurl start count	
autoupdate	latestVersionNumber	view		Returns the latest version number
autoupdate	isLatestVersion	view		Returns 'true' if ZAP is on the latest version
autoupdate	installedAddons	view		Return a list of all of the installed add-ons
autoupdate	newAddons	view		Return a list of any add-ons that have been added to the Marketplace since the last check for updates
autoupdate	updatedAddons	view		Return a list of any add-ons that have been changed in the Marketplace since the last check for updates
autoupdate	marketplaceAddons	view		Return a list of all of the add-ons on the ZAP Marketplace (this information is read once and then cached)
autoupdate	optionAddonDirectories	view		
autoupdate	optionDayLastChecked	view		
autoupdate	optionDayLastInstallWarned	view		
autoupdate	optionDayLastUpdateWarned	view		

Component	Name	Type	Parameters	Description
autoupdate	optionDownloadDirectory	view		
autoupdate	optionCheckAddonUpdates	view		
autoupdate	optionCheckOnStart	view		
autoupdate	optionDownloadNewRelease	view		
autoupdate	optionInstallAddonUpdates	view		
autoupdate	optionInstallScannerRules	view		
autoupdate	optionReportAlphaAddons	view		
autoupdate	optionReportBetaAddons	view		
autoupdate	optionReportReleaseAddons	view		
autoupdate	downloadLatestRelease	action		Downloads the latest release, if any
autoupdate	installAddon	action	id*	Installs or updates the specified add-on, returning when complete (ie not asynchronously)
autoupdate	uninstallAddon	action	id*	Uninstalls the specified add-on
autoupdate	setOptionCheckAddonUpdates	action	Boolean*	
autoupdate	setOptionCheckOnStart	action	Boolean*	
autoupdate	setOptionDownloadNewRelease	action	Boolean*	
autoupdate	setOptionInstallAddonUpdates	action	Boolean*	
autoupdate	setOptionInstallScannerRules	action	Boolean*	
autoupdate	setOptionReportAlphaAddons	action	Boolean*	
autoupdate	setOptionReportBetaAddons	action	Boolean*	
autoupdate	setOptionReportReleaseAddons	action	Boolean*	
spider	status	view	scanId	
spider	results	view	scanId	
spider	fullResults	view	scanId*	
spider	scans	view		
spider	excludedFromScan	view		Gets the regexes of URLs excluded from the spider scans.
spider	allUrls	view		Returns a list of unique URLs from the history table based on HTTP messages added by the Spider.
spider	addedNodes	view	scanId	Returns a list of the names of the nodes added to the Sites tree by the specified scan.

Component	Name	Type	Parameters	Description
spider	domainsAlwaysInScope	view		Gets all the domains that are always in scope. For each domain the following are shown: the index, the value (domain), if enabled, and if specified as a regex.
spider	optionDomainsAlwaysInScope	view		Use view domainsAlwaysInScope instead.
spider	optionDomainsAlwaysInScopeEnabled	view		Use view domainsAlwaysInScope instead.
spider	optionHandleParameters	view		
spider	optionMaxChildren	view		Gets the maximum number of child nodes (per node) that can be crawled, 0 means no limit.
spider	optionMaxDepth	view		
spider	optionMaxDuration	view		
spider	optionMaxParseSizeBytes	view		Gets the maximum size, in bytes, that a response might have to be parsed.
spider	optionMaxScansInUI	view		
spider	optionRequestWaitTime	view		
spider	optionScope	view		
spider	optionScopeText	view		
spider	optionSkipURLString	view		
spider	optionThreadCount	view		
spider	optionUserAgent	view		
spider	optionAcceptCookies	view		Gets whether or not a spider process should accept cookies while spidering.
spider	optionHandleODataParametersVisited	view		
spider	optionParseComments	view		
spider	optionParseGit	view		
spider	optionParseRobotsTxt	view		
spider	optionParseSVNEntries	view		
spider	optionParseSitemapXml	view		
spider	optionPostForm	view		
spider	optionProcessForm	view		
spider	optionSendRefererHeader	view		Gets whether or not the 'Referer' header should be sent while spidering.
spider	optionShowAdvancedDialog	view		

Component	Name	Type	Parameters	Description
spider	scan	action	url maxChildren recurse contextName subtreeOnly	Runs the spider against the given URL (or context). Optionally, the 'maxChildren' parameter can be set to limit the number of children scanned, the 'recurse' parameter can be used to prevent the spider from seeding recursively, the parameter 'contextName' can be used to constrain the scan to a Context and the parameter 'subtreeOnly' allows to restrict the spider under a site's subtree (using the specified 'url').
spider	scanAsUser	action	contextId* userId* url maxChildren recurse subtreeOnly	Runs the spider from the perspective of a User, obtained using the given Context ID and User ID. See 'scan' action for more details.
spider	pause	action	scanId*	
spider	resume	action	scanId*	
spider	stop	action	scanId	
spider	removeScan	action	scanId*	
spider	pauseAllScans	action		
spider	resumeAllScans	action		
spider	stopAllScans	action		
spider	removeAllScans	action		
spider	clearExcludedFromScan	action		Clears the regexes of URLs excluded from the spider scans.
spider	excludeFromScan	action	regex*	Adds a regex of URLs that should be excluded from the spider scans.
spider	addDomainAlwaysInScope	action	value* isRegex isEnabled	Adds a new domain that's always in scope, using the specified value. Optionally sets if the new entry is enabled (default, true) and whether or not the new value is specified as a regex (default, false).

Component	Name	Type	Parameters	Description
spider	modifyDomainAlwaysInScope	action	idx* value isRegex isEnabled	Modifies a domain that's always in scope. Allows to modify the value, if enabled or if a regex. The domain is selected with its index, which can be obtained with the view domainsAlwaysInScope.
spider	removeDomainAlwaysInScope	action	idx*	Removes a domain that's always in scope, with the given index. The index can be obtained with the view domainsAlwaysInScope.
spider	enableAllDomainsAlwaysInScope	action		Enables all domains that are always in scope.
spider	disableAllDomainsAlwaysInScope	action		Disables all domains that are always in scope.
spider	setOptionHandleParameters	action	String*	
spider	setOptionScopeString	action	String*	Use actions [add
spider	setOptionSkipURLString	action	String*	
spider	setOptionUserAgent	action	String*	
spider	setOptionAcceptCookies	action	Boolean*	Sets whether or not a spider process should accept cookies while spidering.
spider	setOptionHandleODataParametersVisited	action	Boolean*	
spider	setOptionMaxChildren	action	Integer*	Sets the maximum number of child nodes (per node) that can be crawled, 0 means no limit.
spider	setOptionMaxDepth	action	Integer*	
spider	setOptionMaxDuration	action	Integer*	
spider	setOptionMaxParseSizeBytes	action	Integer*	Sets the maximum size, in bytes, that a response might have to be parsed. This allows the spider to skip big responses/files.
spider	setOptionMaxScansInUI	action	Integer*	
spider	setOptionParseComments	action	Boolean*	
spider	setOptionParseGit	action	Boolean*	
spider	setOptionParseRobotsTxt	action	Boolean*	
spider	setOptionParseSVNEntries	action	Boolean*	
spider	setOptionParseSitemapXml	action	Boolean*	
spider	setOptionPostForm	action	Boolean*	
spider	setOptionProcessForm	action	Boolean*	
spider	setOptionRequestWaitTime	action	Integer*	

Component	Name	Type	Parameters	Description
spider	setOptionSendRefererHeader	action	Boolean*	Sets whether or not the 'Referer' header should be sent while spidering.
spider	setOptionShowAdvancedDialog	action	Boolean*	
spider	setOptionThreadCount	action	Integer*	
core	alert	view	id*	Gets the alert with the given ID, the corresponding HTTP message can be obtained with the 'messageId' field and 'message' API method
core	alerts	view	baseurl start count riskId	Gets the alerts raised by ZAP, optionally filtering by URL or riskId, and paginating with 'start' position and 'count' of alerts
core	alertsSummary	view	baseurl	Gets number of alerts grouped by each risk level, optionally filtering by URL
core	numberOfAlerts	view	baseurl riskId	Gets the number of alerts, optionally filtering by URL or riskId
core	hosts	view		Gets the name of the hosts accessed through/by ZAP
core	sites	view		Gets the sites accessed through/by ZAP (scheme and domain)
core	urls	view	baseurl	Gets the URLs accessed through/by ZAP, optionally filtering by (base) URL.
core	message	view	id*	Gets the HTTP message with the given ID. Returns the ID, request/response headers and bodies, cookies, note, type, RTT, and timestamp.
core	messages	view	baseurl start count	Gets the HTTP messages sent by ZAP, request and response, optionally filtered by URL and paginated with 'start' position and 'count' of messages
core	messagesById	view	ids*	Gets the HTTP messages with the given IDs.
core	numberOfMessages	view	baseurl	Gets the number of messages, optionally filtering by URL
core	mode	view		Gets the mode
core	version	view		Gets ZAP version

Component	Name	Type	Parameters	Description
core	excludedFromProxy	view		Gets the regular expressions, applied to URLs, to exclude from the local proxies.
core	homeDirectory	view		
core	sessionLocation	view		Gets the location of the current session file
core	proxyChainExcludedDomains	view		Gets all the domains that are excluded from the outgoing proxy. For each domain the following are shown: the index, the value (domain), if enabled, and if specified as a regex.
core	optionProxyChainSkipName	view		Use view proxyChainExcludedDomains instead.
core	optionProxyExcludedDomains	view		Use view proxyChainExcludedDomains instead.
core	optionProxyExcludedDomainsEnabled	view		Use view proxyChainExcludedDomains instead.
core	zapHomePath	view		Gets the path to ZAP's home directory.
core	optionMaximumAlertInstances	view		Gets the maximum number of alert instances to include in a report.
core	optionMergeRelatedAlerts	view		Gets whether or not related alerts will be merged in any reports generated.
core	optionAlertOverridesFilePath	view		Gets the path to the file with alert overrides.
core	optionDefaultUserAgent	view		Gets the user agent that ZAP should use when creating HTTP messages (for example, spider messages or CONNECT requests to outgoing proxy).
core	optionDnsTtlSuccessfulQueries	view		Gets the TTL (in seconds) of successful DNS queries.
core	optionHttpState	view		
core	optionProxyChainName	view		
core	optionProxyChainPassword	view		
core	optionProxyChainPort	view		
core	optionProxyChainRealm	view		
core	optionProxyChainUserName	view		
core	optionTimeoutInSecs	view		
core	optionHttpStateEnabled	view		

Component	Name	Type	Parameters	Description
core	optionProxyChainPrompt	view		
core	optionSingleCookieRequest Header	view		
core	optionUseProxyChain	view		
core	optionUseProxyChainAuth	view		
core	accessUrl	action	url* followRedirects	Convenient and simple action to access a URL, optionally following redirections. Returns the request sent and response received and followed redirections, if any. Other actions are available which offer more control on what is sent, like, 'sendRequest' or 'sendHarRequest'.
core	shutdown	action		Shuts down ZAP
core	newSession	action	name overwrite	Creates a new session, optionally overwriting existing files. If a relative path is specified it will be resolved against the "session" directory in ZAP "home" dir.
core	loadSession	action	name*	Loads the session with the given name. If a relative path is specified it will be resolved against the "session" directory in ZAP "home" dir.
core	saveSession	action	name* overwrite	Saves the session with the name supplied, optionally overwriting existing files. If a relative path is specified it will be resolved against the "session" directory in ZAP "home" dir.
core	snapshotSession	action		
core	clearExcludedFromProxy	action		Clears the regexes of URLs excluded from the local proxies.
core	excludeFromProxy	action	regex*	Adds a regex of URLs that should be excluded from the local proxies.
core	setHomeDirectory	action	dir*	
core	setMode	action	mode*	Sets the mode, which may be one of [safe, protect, standard, attack]
core	generateRootCA	action		Generates a new Root CA certificate for the local proxies.

Component	Name	Type	Parameters	Description
core	sendRequest	action	request* followRedirects	Sends the HTTP request, optionally following redirections. Returns the request sent and response received and followed redirections, if any. The Mode is enforced when sending the request (and following redirections), custom manual requests are not allowed in 'Safe' mode nor in 'Protected' mode if out of scope.
core	deleteAllAlerts	action		Deletes all alerts of the current session.
core	deleteAlert	action	id*	Deletes the alert with the given ID.
core	runGarbageCollection	action		
core	deleteSiteNode	action	url* method postData	Deletes the site node found in the Sites Tree on the basis of the URL, HTTP method, and post data (if applicable and specified).
core	addProxyChainExcludedDomain	action	value* isRegex isEnabled	Adds a domain to be excluded from the outgoing proxy, using the specified value. Optionally sets if the new entry is enabled (default, true) and whether or not the new value is specified as a regex (default, false).
core	modifyProxyChainExcludedDomain	action	idx* value isRegex isEnabled	Modifies a domain excluded from the outgoing proxy. Allows to modify the value, if enabled or if a regex. The domain is selected with its index, which can be obtained with the view proxyChainExcludedDomains.
core	removeProxyChainExcludedDomain	action	idx*	Removes a domain excluded from the outgoing proxy, with the given index. The index can be obtained with the view proxyChainExcludedDomains.
core	enableAllProxyChainExcludedDomains	action		Enables all domains excluded from the outgoing proxy.
core	disableAllProxyChainExcludedDomains	action		Disables all domains excluded from the outgoing proxy.

Component	Name	Type	Parameters	Description
core	setOptionMaximumAlertInstances	action	numberOfInstances*	Sets the maximum number of alert instances to include in a report. A value of zero is treated as unlimited.
core	setOptionMergeRelatedAlerts	action	enabled*	Sets whether or not related alerts will be merged in any reports generated.
core	setOptionAlertOverridesFilePath	action	filePath	Sets (or clears, if empty) the path to the file with alert overrides.
core	setOptionDefaultUserAgent	action	String*	Sets the user agent that ZAP should use when creating HTTP messages (for example, spider messages or CONNECT requests to outgoing proxy).
core	setOptionProxyChainName	action	String*	
core	setOptionProxyChainPassword	action	String*	
core	setOptionProxyChainRealm	action	String*	
core	setOptionProxyChainSkipName	action	String*	Use actions [add
core	setOptionProxyChainUserName	action	String*	
core	setOptionDnsTtlSuccessfulQueries	action	Integer*	Sets the TTL (in seconds) of successful DNS queries (applies after ZAP restart).
core	setOptionHttpStateEnabled	action	Boolean*	
core	setOptionProxyChainPort	action	Integer*	
core	setOptionProxyChainPrompt	action	Boolean*	
core	setOptionSingleCookieRequestHeader	action	Boolean*	
core	setOptionTimeoutInSecs	action	Integer*	
core	setOptionUseProxyChain	action	Boolean*	Sets whether or not the outgoing proxy should be used. The address/hostname of the outgoing proxy must be set to enable this option.
core	setOptionUseProxyChainAuth	action	Boolean*	
core	proxy.pac	other		
core	rootcert	other		Gets the Root CA certificate used by the local proxies.
core	setproxy	other	proxy*	
core	xmlreport	other		Generates a report in XML format
core	htmlreport	other		Generates a report in HTML format
core	jsonreport	other		Generates a report in JSON format

Component	Name	Type	Parameters	Description
core	mdreport	other		Generates a report in Markdown format
core	messageHar	other	id*	Gets the message with the given ID in HAR format
core	messagesHar	other	baseurl start count	Gets the HTTP messages sent through/by ZAP, in HAR format, optionally filtered by URL and paginated with 'start' position and 'count' of messages
core	messagesHarById	other	ids*	Gets the HTTP messages with the given IDs, in HAR format.
core	sendHarRequest	other	request* followRedirects	Sends the first HAR request entry, optionally following redirections. Returns, in HAR format, the request sent and response received and followed redirections, if any. The Mode is enforced when sending the request (and following redirections), custom manual requests are not allowed in 'Safe' mode nor in 'Protected' mode if out of scope.
params	params	view	site	Shows the parameters for the specified site, or for all sites if the site is not specified
ascan	status	view	scanId	
ascan	scanProgress	view	scanId	
ascan	messagesIds	view	scanId*	Gets the IDs of the messages sent during the scan with the given ID. A message can be obtained with 'message' core view.
ascan	alertsIds	view	scanId*	Gets the IDs of the alerts raised during the scan with the given ID. An alert can be obtained with 'alert' core view.
ascan	scans	view		
ascan	scanPolicyNames	view		
ascan	excludedFromScan	view		Gets the regexes of URLs excluded from the active scans.
ascan	scanners	view	scanPolicyName policyId	
ascan	policies	view	scanPolicyName policyId	

Component	Name	Type	Parameters	Description
ascan	attackModeQueue	view		
ascan	excludedParams	view		Gets all the parameters that are excluded. For each parameter the following are shown: the name, the URL, and the parameter type.
ascan	optionExcludedParamList	view		Use view excludedParams instead.
ascan	excludedParamTypes	view		Gets all the types of excluded parameters. For each type the following are shown: the ID and the name.
ascan	optionAttackPolicy	view		
ascan	optionDefaultPolicy	view		
ascan	optionDelayInMs	view		
ascan	optionHandleAntiCSRFTokens	view		
ascan	optionHostPerScan	view		
ascan	optionMaxChartTimeInMins	view		
ascan	optionMaxResultsToList	view		
ascan	optionMaxRuleDurationInMins	view		
ascan	optionMaxScanDurationInMins	view		
ascan	optionMaxScansInUI	view		
ascan	optionTargetParamsEnabledRPC	view		
ascan	optionTargetParamsInjectable	view		
ascan	optionThreadPerHost	view		
ascan	optionAllowAttackOnStart	view		
ascan	optionInjectPluginIdInHeader	view		Tells whether or not the active scanner should inject the HTTP request header X-ZAP-Scan-ID, with the ID of the scanner that's sending the requests.
ascan	optionPromptInAttackMode	view		
ascan	optionPromptToClearFinishedScans	view		
ascan	optionRescanInAttackMode	view		
ascan	optionScanHeadersAllRequests	view		Tells whether or not the HTTP Headers of all requests should be scanned. Not just requests that send parameters, through the query or request body.

Component	Name	Type	Parameters	Description
ascan	optionShowAdvancedDialog	view		
ascan	scan	action	url recurse inScopeOnly scanPolicyName method postData contextId	Runs the active scanner against the given URL and/or Context. Optionally, the 'recurse' parameter can be used to scan URLs under the given URL, the parameter 'inScopeOnly' can be used to constrain the scan to URLs that are in scope (ignored if a Context is specified), the parameter 'scanPolicyName' allows to specify the scan policy (if none is given it uses the default scan policy), the parameters 'method' and 'postData' allow to select a given request in conjunction with the given URL.
ascan	scanAsUser	action	url contextId userId recurse scanPolicyName method postData	Active Scans from the perspective of a User, obtained using the given Context ID and User ID. See 'scan' action for more details.
ascan	pause	action	scanId*	
ascan	resume	action	scanId*	
ascan	stop	action	scanId*	
ascan	removeScan	action	scanId*	
ascan	pauseAllScans	action		
ascan	resumeAllScans	action		
ascan	stopAllScans	action		
ascan	removeAllScans	action		
ascan	clearExcludedFromScan	action		Clears the regexes of URLs excluded from the active scans.
ascan	excludeFromScan	action	regex*	Adds a regex of URLs that should be excluded from the active scans.
ascan	enableAllScanners	action	scanPolicyName	
ascan	disableAllScanners	action	scanPolicyName	
ascan	enableScanners	action	ids* scanPolicyName	
ascan	disableScanners	action	ids* scanPolicyName	
ascan	setEnabledPolicies	action	ids* scanPolicyName	

Component	Name	Type	Parameters	Description
ascan	setPolicyAttackStrength	action	id* attackStrength* scanPolicyName	
ascan	setPolicyAlertThreshold	action	id* alertThreshold* scanPolicyName	
ascan	setScannerAttackStrength	action	id* attackStrength* scanPolicyName	
ascan	setScannerAlertThreshold	action	id* alertThreshold* scanPolicyName	
ascan	addScanPolicy	action	scanPolicyName* alertThreshold attackStrength	
ascan	removeScanPolicy	action	scanPolicyName*	
ascan	updateScanPolicy	action	scanPolicyName* alertThreshold attackStrength	
ascan	importScanPolicy	action	path*	Imports a Scan Policy using the given file system path.
ascan	addExcludedParam	action	name* type url	Adds a new parameter excluded from the scan, using the specified name. Optionally sets if the new entry applies to a specific URL (default, all URLs) and sets the ID of the type of the parameter (default, ID of any type). The type IDs can be obtained with the view excludedParamTypes.
ascan	modifyExcludedParam	action	idx* name type url	Modifies a parameter excluded from the scan. Allows to modify the name, the URL and the type of parameter. The parameter is selected with its index, which can be obtained with the view excludedParams.
ascan	removeExcludedParam	action	idx*	Removes a parameter excluded from the scan, with the given index. The index can be obtained with the view excludedParams.
ascan	skipScanner	action	scanId* scannerId*	Skips the scanner using the given IDs of the scan and the scanner.
ascan	setOptionAttackPolicy	action	String*	
ascan	setOptionDefaultPolicy	action	String*	
ascan	setOptionAllowAttackOnStart	action	Boolean*	
ascan	setOptionDelayInMs	action	Integer*	

Component	Name	Type	Parameters	Description
ascan	setOptionHandleAntiCSRFTokens	action	Boolean*	
ascan	setOptionHostPerScan	action	Integer*	
ascan	setOptionInjectPluginIdInHeader	action	Boolean*	Sets whether or not the active scanner should inject the HTTP request header X-ZAP-Scan-ID, with the ID of the scanner that's sending the requests.
ascan	setOptionMaxChartTimeInMins	action	Integer*	
ascan	setOptionMaxResultsToList	action	Integer*	
ascan	setOptionMaxRuleDurationInMins	action	Integer*	
ascan	setOptionMaxScanDurationInMins	action	Integer*	
ascan	setOptionMaxScansInUI	action	Integer*	
ascan	setOptionPromptInAttackMode	action	Boolean*	
ascan	setOptionPromptToClearFinishedScans	action	Boolean*	
ascan	setOptionRescanInAttackMode	action	Boolean*	
ascan	setOptionScanHeadersAllRequests	action	Boolean*	Sets whether or not the HTTP Headers of all requests should be scanned. Not just requests that send parameters, through the query or request body.
ascan	setOptionShowAdvancedDialog	action	Boolean*	
ascan	setOptionTargetParamsEnabledRPC	action	Integer*	
ascan	setOptionTargetParamsInjectable	action	Integer*	
ascan	setOptionThreadPerHost	action	Integer*	
context	contextList	view		List context names of current session
context	excludeRegexs	view	contextName*	List excluded regexs for context
context	includeRegexs	view	contextName*	List included regexs for context
context	context	view	contextName*	List the information about the named context
context	technologyList	view		Lists the names of all built in technologies
context	includedTechnologyList	view	contextName*	Lists the names of all technologies included in a context
context	excludedTechnologyList	view	contextName*	Lists the names of all technologies excluded from a context

Component	Name	Type	Parameters	Description
context	excludeFromContext	action	contextName* regex*	Add exclude regex to context
context	includeInContext	action	contextName* regex*	Add include regex to context
context	newContext	action	contextName*	Creates a new context with the given name in the current session
context	removeContext	action	contextName*	Removes a context in the current session
context	exportContext	action	contextName* contextFile*	Exports the context with the given name to a file. If a relative file path is specified it will be resolved against the "contexts" directory in ZAP "home" dir.
context	importContext	action	contextFile*	Imports a context from a file. If a relative file path is specified it will be resolved against the "contexts" directory in ZAP "home" dir.
context	includeContextTechnologies	action	contextName* technologyNames*	Includes technologies with the given names, separated by a comma, to a context
context	includeAllContextTechnologies	action	contextName*	Includes all built in technologies in to a context
context	excludeContextTechnologies	action	contextName* technologyNames*	Excludes technologies with the given names, separated by a comma, from a context
context	excludeAllContextTechnologies	action	contextName*	Excludes all built in technologies from a context
context	setContextInScope	action	contextName* booleanInScope*	Sets a context to in scope (contexts are in scope by default)
httpSessions	sites	view		Gets all of the sites that have sessions.
httpSessions	sessions	view	site* session	Gets the sessions for the given site. Optionally returning just the session with the given name.
httpSessions	activeSession	view	site*	Gets the name of the active session for the given site.
httpSessions	sessionTokens	view	site*	Gets the names of the session tokens for the given site.
httpSessions	createEmptySession	action	site* session	Creates an empty session for the given site. Optionally with the given name.
httpSessions	removeSession	action	site* session*	Removes the session from the given site.
httpSessions	setActiveSession	action	site* session*	Sets the given session as active for the given site.

Component	Name	Type	Parameters	Description
httpSessions	unsetActiveSession	action	site*	Unsets the active session of the given site.
httpSessions	addSessionToken	action	site* sessionToken*	Adds the session token to the given site.
httpSessions	removeSessionToken	action	site* sessionToken*	Removes the session token from the given site.
httpSessions	setSessionTokenValue	action	site* session* sessionToken* tokenValue*	Sets the value of the session token of the given session for the given site.
httpSessions	renameSession	action	site* oldSessionName* newSessionName*	Renames the session of the given site.
break	isBreakAll	view		Returns True if ZAP will break on both requests and responses
break	isBreakRequest	view		Returns True if ZAP will break on requests
break	isBreakResponse	view		Returns True if ZAP will break on responses
break	httpMessage	view		Returns the HTTP message currently intercepted (if any)
break	break	action	type* state* scope	Controls the global break functionality. The type may be one of: http-all, http-request or http-response. The state may be true (for turning break on for the specified type) or false (for turning break off). Scope is not currently used.
break	setHttpMessage	action	httpHeader* httpBody	Overwrites the currently intercepted message with the data provided
break	continue	action		Submits the currently intercepted message and unsets the global request/response break points
break	step	action		Submits the currently intercepted message, the next request or response will automatically be intercepted
break	drop	action		Drops the currently intercepted message

Component	Name	Type	Parameters	Description
break	addHttpBreakpoint	action	string* location* match* inverse* ignorecase*	Adds a custom HTTP breakpont. The string is the string to match. Location may be one of: url, request_header, request_body, response_header or response_body. Match may be: contains or regex. Inverse (match) may be true or false. Lastly, ignorecase (when matching the string) may be true or false.
break	removeHttpBreakpoint	action	string* location* match* inverse* ignorecase*	Removes the specified break point
authentication	getSupportedAuthentication Methods	view		
authentication	getAuthenticationMethodCo nfigParams	view	authMethodNam e*	
authentication	getAuthenticationMethod	view	contextId*	
authentication	getLoggedInIndicator	view	contextId*	
authentication	getLoggedOutIndicator	view	contextId*	
authentication	setAuthenticationMethod	action	contextId* authMethodNam e* authMethodConfi gParams	
authentication	setLoggedInIndicator	action	contextId* loggedInIndicator Regex*	
authentication	setLoggedOutIndicator	action	contextId* loggedOutIndicat orRegex*	
authorization	getAuthorizationDetectionM ethod	view	contextId*	Obtains all the configuration of the authorization detection method that is currently set for a context.
authorization	setBasicAuthorizationDetect ionMethod	action	contextId* headerRegex bodyRegex statusCode logicalOperator	Sets the authorization detection method for a context as one that identifies un-authorized messages based on: the message's status code or a regex pattern in the response's header or body. Also, whether all conditions must match or just some can be specified via the logicalOperator parameter, which accepts two values: "AND" (default), "OR".

Component	Name	Type	Parameters	Description
sessionManagement	getSupportedSessionManagementMethods	view		
sessionManagement	getSessionManagementMethodConfigParams	view	methodName*	
sessionManagement	getSessionManagementMethod	view	contextId*	
sessionManagement	setSessionManagementMethod	action	contextId* methodName* methodConfigParams	
users	usersList	view	contextId	
users	getUserById	view	contextId userId	
users	getAuthenticationCredentialsConfigParams	view	contextId*	
users	getAuthenticationCredentials	view	contextId* userId*	
users	newUser	action	contextId* name*	
users	removeUser	action	contextId* userId*	
users	setUserEnabled	action	contextId* userId* enabled*	
users	setUserName	action	contextId* userId* name*	
users	setAuthenticationCredentials	action	contextId* userId* authCredentialsConfigParams	
forcedUser	isForcedUserModeEnabled	view		Returns 'true' if 'forced user' mode is enabled, 'false' otherwise
forcedUser	getForcedUser	view	contextId*	Gets the user (ID) set as 'forced user' for the given context (ID)
forcedUser	setForcedUser	action	contextId* userId*	Sets the user (ID) that should be used in 'forced user' mode for the given context (ID)
forcedUser	setForcedUserModeEnabled	action	boolean*	Sets if 'forced user' mode should be enabled or not
script	listEngines	view		Lists the script engines available
script	listScripts	view		Lists the scripts available, with its engine, name, description, type and error state.
script	enable	action	scriptName*	Enables the script with the given name
script	disable	action	scriptName*	Disables the script with the given name

Component	Name	Type	Parameters	Description
script	load	action	scriptName* scriptType* scriptEngine* fileName* scriptDescription charset	Loads a script into ZAP from the given local file, with the given name, type and engine, optionally with a description, and a charset name to read the script (the charset name is required if the script is not in UTF-8, for example, in ISO-8859-1).
script	remove	action	scriptName*	Removes the script with the given name
script	runStandAloneScript	action	scriptName*	Runs the stand alone script with the give name
stats	stats	view	keyPrefix	Statistics
stats	allSitesStats	view	keyPrefix	Gets all of the site based statistics, optionally filtered by a key prefix
stats	siteStats	view	site* keyPrefix	Gets all of the global statistics, optionally filtered by a key prefix
stats	optionStatsdHost	view		Gets the Statsd service hostname
stats	optionStatsdPort	view		Gets the Statsd service port
stats	optionStatsdPrefix	view		Gets the prefix to be applied to all stats sent to the configured Statsd service
stats	optionInMemoryEnabled	view		Returns 'true' if in memory statistics are enabled, otherwise returns 'false'
stats	optionStatsdEnabled	view		Returns 'true' if a Statsd server has been correctly configured, otherwise returns 'false'
stats	clearStats	action	keyPrefix	Clears all of the statistics
stats	setOptionStatsdHost	action	String*	Sets the Statsd service hostname, supply an empty string to stop using a Statsd service
stats	setOptionStatsdPrefix	action	String*	Sets the prefix to be applied to all stats sent to the configured Statsd service
stats	setOptionInMemoryEnabled	action	Boolean*	Sets whether in memory statistics are enabled
stats	setOptionStatsdPort	action	Integer*	Sets the Statsd service port

* Starred parameters are mandatory.