

Sidebar

Programming · VS Code REST Client Cheat Sheet

Sending HTTP requests from a text file inside VS Code: the request format, variables, environments, authentication, commands and settings. Checked against REST Client 0.25.1 installed here. Sheet 21 in the VS Code series. Handy for trying out a FastAPI or other web API without leaving the editor.

GETTING STARTED & SHORTCUTS

Extension ID	humao.rest-client (version 0.25.1 here)
File type	Save requests in a <code>.http</code> or <code>.rest</code> file (language mode <i>HTTP</i>)
Send Request (link)	A clickable link above each request in HTTP mode
Ctrl+Alt+R	Send the request at the cursor, or the selected text
Ctrl+Alt+L	Rerun the last request (also from the response panel)
Ctrl+Alt+K	Cancel a request that is still waiting
Ctrl+Alt+H	View request history
Ctrl+Alt+E	Switch environment
Ctrl+Alt+C	Generate a code snippet for the request
###	A line of three or more <code>#</code> separates requests in one file
Hover the status bar	Hover the time to see the DNS, TCP, first-byte and download breakdown; hover the size for headers and body

VARIABLES: KINDS AND SYNTAX

{{name}}	Use a custom variable; system variables use <code>{{\${name}}}</code>
Environment variable	Defined in settings under <code>rest-client.environmentVariables</code> ; changes with the chosen environment
@name = value	File variable: a whole line anywhere in the file; can use other variables
# @prompt name Description	Prompt variable: you are asked for the value each time you send
# @name login	Names a request so later requests can use its response
{{login.response.body.\$.token}}	Use a part of a named request's response: JSONPath in the body
{{login.response.headers.X-AuthToken}}	Use a response header
{{x.response.body.//reply[1]/@id}}	XPath works for XML bodies
Unresolved reference	The plain text is sent instead, and a diagnostic appears

AUTHENTICATION

Authorization: Basic user:passwd	Basic auth; the extension encodes it. Also accepted: <code>Basic user passwd</code> or an already base64-encoded value
Authorization: Digest user passwd	Digest auth
rest-client.certificates	Client certificates per host, e.g. { "localhost:8080": { "pfx": "./client.pfx", "passphrase": "..." } } (default { })
Azure AD / AWS Signature v4	Supported; see the extension's README for the exact syntax
Keep secrets out of the file	Use <code>{{\${processEnv ...}}}</code> , <code>{{\${dotenv ...}}}</code> or a <i>user</i> setting, never a token typed into a committed <code>.http</code> file

WRITING A REQUEST

GET https://example.com/items	Method, URL
Query string over several lines	Put <code>?page</code> request line
Header: value	Header lines follow the request line
(blank line) body	Everything after the first blank line
< ./body.json	Use a file as the body (path absolute)
<@ ./body.json	Same, but substitute <code>{{variable}}</code> <code><@latin1</code>
multipart/form-data	Write boundary-separated
x-www-form-urlencoded	One <code>name=value</code> per line
X-REQUEST-TYPE: GraphQL	Header that makes the request GraphQL (blank line)
curl command	Paste a <code>curl</code> command and send
# comment or // comment	Comments

SYSTEM VARIABLES

{{\${guid}}}	A new random UUID
{{\${randomInt 5 200}}}	A random whole number between 5 and 200
{{\${timestamp}}}	Now as a Unix timestamp; <code>{{\${timestamp 1000}}}</code> is 1000 ms ago
{{\${datetime iso8601}}}	Now in UTC as <code>iso8601</code> <code>'YYYY-MM-DD'</code>
{{\${localDatetime 'YYYY-MM-DD'}}	The same in local time
{{\${processEnv NAME}}}	A value from your operating system
{{\${processEnv %secretKey}}}	With <code>%</code> , look up <code>secretKey</code> in the REST Client environment
{{\${dotenv NAME}}}	A value from a <code>.env</code> file near the file
Offset units	<code>y</code> , <code>M</code> , <code>w</code> , <code>d</code> , <code>h</code> , <code>m</code> , <code>s</code> , <code>ms</code> , <code>ns</code>
{{\${aadToken}}}	Azure Active Directory token
{{\${aadV2Token}}}	Azure Active Directory token (scopes: <code>scopes: https://graph.microsoft.com/.default</code>)

ENVIRONMENTS

rest-client.environmentVariables	{ "\${shared}" }
\$shared	Variables available in every environment; <code>rest-client.environmentVariables</code> is the default
Rest Client: Switch Environment	Command to switch environment in status bar
{{\${shared name}}}	Refer to a shared variable from <code>rest-client.environmentVariables</code>
No Environment	Choose it to use only file and request variables

BEHAVIOUR SETTINGS (REST-CLIENT.*)

- rest-client.followRedirectFollow 3xx responses (default `true`); per request: `# @no-redirect`
- rest-client.defaultHeadersAdded when you omit them (default `{ "User-Agent": "vscode-restclient" }`)
- rest-client.timeoutInMillisecondsRequest timeout; `0` (default) means wait forever
- rest-client.rememberCookiesForSubsequentRequestsKeep cookies between requests (default `true`); per request: `# @no-cookie-jar`
- rest-client.excludeHostsForProxyHosts that skip the VS Code proxy settings (default `[]`)
- rest-client.formParamEncodingStrategy`automatic` (default), `never`, `always`: URL-encode form values
- rest-client.decodeEscapedUnicodeCharactersShow `\u00e9`-style escapes as real characters (default `false`)
- rest-client.addRequestBodyLineIndentationAroundBracketsAuto-indent inside `{ }` when you press Enter (default `true`)
- rest-client.enableTelemetryAnonymous usage data (default `true`)
- rest-client.logLevel`error` (default), `warn`, `info`, `verbose` in the REST output panel
- rest-client.enableSendRequestCodeLensThe Send Request link (default `true`)
- rest-client.enableCustomVariableReferencesCodeLensA references count above variables (default `true`)

RESPONSE PREVIEW

- rest-client.previewOption`full` (default), `code`
- rest-client.previewColumn`beside` (default), `below`
- rest-client.previewResponsePanelTakeFocusWhen the response preview is active
- rest-client.showResponseInDifferentTabAlways
- rest-client.requestNameAsResponseTabTitleUse request name as response tab title
- rest-client.previewResponseInUntitledDocumentOpen response preview in untitled document
- rest-client.fontSize / fontFamily / fontWeightFont settings for response preview
- rest-client.largeResponseBodySizeLimitInMBLimit response body size in MB
- rest-client.disableHighlightResonseBodyForLargeResponsesDisable highlighting response body for large responses
- rest-client.disableAddingHrefLinkForLargeResponsesDisable adding href link for large responses
- rest-client.suppressResponseBodyContentTypeTextOnlySuppress response body content type text only
- rest-client.mimeAndFileExtensionMappingMapping for mime and file extension
- rest-client.useContentDispositionFilenameUse content disposition filename

COMMANDS (REST CLIENT: ...)

- Send Request / Rerun Last Request / Cancel RequestRun, repeat or stop a request
- View Request History / Clear Request HistoryEvery sent request is stored; ask for it again from the history list
- Switch EnvironmentPick the active environment
- Save Full Response / Save Response BodySave the whole response, or just the body, to a file
- Copy Response BodyCopy the body to the clipboard
- Generate Code Snippet / Copy Code SnippetTurn the request into code in many languages and libraries
- Copy Request As cURLGet an equivalent `curl` command (also on the right-click menu)
- Fold Response / Unfold ResponseCollapse or expand the response body
- Clear Azure AD Token CacheForget cached Azure tokens
- # @notePer-request: ask for confirmation before sending (for risky requests)

A SAMPLE FILE TO COPY (ADJUST TO YOUR API)

- @host = http://localhost:8000A file variable
- # @name healthNames the request that follows
- GET {{host}}/healthA simple GET using the variable
- POST {{host}}/itemsFollowed by `Content-Type: application/json` body
- GET {{host}}/items/{{create.response.body.id}}
- ###Separator between requests

CTRL+ALT+K MAY CLASH WITH THE CLAUDE CODE EXTENSION

REST Client uses `Ctrl+Alt+K` to cancel a request in an HTTP file, and the Claude Code extension also binds `Ctrl+Alt+K` (insert the current selection as a reference) whenever an editor has focus. If cancel does not work, open the Keyboard Shortcuts editor (`Ctrl+K Ctrl+S`), search for `ctrl+alt+k`, and give one of them a different key, or use the cancel link or *Rest Client: Cancel Request* from the Command Palette instead.

NEVER COMMIT REAL TOKENS IN A .HTTP FILE

A request file is plain text and usually lives in your project, so anything typed into it ends up in Git. Put secrets in a `.env` file (listed in `.gitignore`) and read them with `{{${dotenv NAME}}}`, or read them from the operating system with `{{${processEnv NAME}}}`. Environments defined in your *user* settings stay out of the repository, but those in a workspace `.vscode/settings.json` are committed with it.