

Description [Plain text] Preview

Discard old builds
GitHub project
Project url https://github.com/s390xlinux/jenkins-test/
Advanced...
This build requires lockable resources
This project is parameterized
String Parameter
Name github_webhook_uid
Default Value \${x_github_delivery}
Description
[Plain text] Preview
Trim the string
Add Parameter

Throttle builds
Disable this project
Execute concurrent builds if necessary
Advanced...

Source Code Management

None
Git
Repositories
Repository URL https://github.com/s390xlinux/jenkins-test.git
Credentials - none -
Advanced...
Add Repository
Branches to build
Branch Specifier (blank for 'any') */master
Add Branch
Repository browser (Auto)
Advanced sub-modules behaviours
Disable submodules processing
Recursively update submodules
Update tracking submodules to tip of parent
Use credentials from default remote of parent repository
Path of the reference repo to use during submodule update
Timeout (in minutes) for submodules operations
Number of threads to use when updating submodules
Shallow clone
Advanced clone behaviours
Fetch tags
Honor refspec on initial clone
Path of the reference repo to use during clone
Timeout (in minutes) for clone and fetch operations
Shallow clone
Shallow clone depth
Add

Subversion

Build Triggers

Trigger builds remotely (e.g., from scripts)
Build after other projects are built
Build periodically
Generic Webhook Trigger
Is triggered by HTTP requests to http://JENKINS_URL/generic-webhook-trigger/invoke
There are example configurations in the Git repository.
You can fiddle with JSONPath here. You may also want to checkout the syntax here.
You can fiddle with XPath here. You may also want to checkout the syntax here.
You can fiddle with regular expressions here. You may also want to checkout the syntax here.
If your job is not parameterized, then the resolved variables will just be contributed to the build. If your job is parameterized, and you resolve variables that have the same name as those parameters, then the plugin will populate the parameters when triggering job. That means you can, for example, use the parameters in combination with an SCM plugin, like GIT Plugin, to pick a branch.

Post content parameters
Variable github_webhook_payload
Name of variable
Expression \$
JSONPath
XPath
Expression to evaluate in POST content. Use JSONPath for JSON or XPath for XML.
Value filter
Optional. Anything in the evaluated value, matching this regular expression, will be removed. Having [*0-9] would only allow numbers. The regex syntax is documented here.
Default value
Optional. This value will be used if expression does not match anything.
Add
If you want value of param1 from post content { "param1": "value1" } to be contributed, you need to add \$.param1 here.

Header parameters
Request header x-github-delivery
Name of request header in lowercase. Resulting variable name has '-' instead of '.' characters.
Value filter
Optional. Anything in the evaluated value, matching this regular expression, will be removed. Having [*0-9] would only allow numbers. The regex syntax is documented here.
Request header x-github-event
Name of request header in lowercase. Resulting variable name has '-' instead of '.' characters.
Value filter
Optional. Anything in the evaluated value, matching this regular expression, will be removed. Having [*0-9] would only allow numbers. The regex syntax is documented here.
Add
If you want value of header param1 to be contributed, you need to add "param1" here.

Request parameters
Add
If you want value of query parameter param1 to be contributed, you need to add "param1" here.

Token
Optional token. If it is specified then this job can only be triggered if that token is supplied when invoking http://JENKINS_URL/generic-webhook-trigger/invoke. It can be supplied as:
Query parameter /invoke?token=TOKEN_HERE
A token header token: TOKEN_HERE
A Authorization: Bearer header Authorization: Bearer TOKEN_HERE

Token Credential
Add
Same as token above, but configured with a secret text credential.

Cause
Github Webhook Event \${x_github_delivery} \${x_github_event}
This will be displayed in any triggered job. You can use the variables here to create a custom cause like "Name committed to \$branch", if you have configured variables named name and branch.

Override Quiet Period
Allow the trigger to override this job's quiet period. If selected you can provide a quiet period (an integer number of seconds) and the build will use that quiet period instead of its default. If this is selected and no quiet period is given the job's quiet period will still be used. It can be supplied as:
Query Parameter /invoke?jobQuietPeriod=QUIET_PERIOD_HERE
A quiet period header jobQuietPeriod: QUIET_PERIOD_HERE

Silent response
Avoid responding with information about triggered jobs.

Print post content
Print post content in job log.

Print contributed variables
Print contributed variables in job log.

Optional filter

Expression
Regular expression to test on the evaluated text specified below. The regex syntax is documented here.
Text
Text to test for the given expression. You can use any combination of the variables you configured above.
This is an optional feature. If specified, this job will only trigger when given expression matches given text.

GitHub Pull Request Builder
GitHub hook trigger for GITScm polling
Poll SCM

Build Environment

Delete workspace before build starts
Advanced...
Use secret text(s) or file(s)
Abort the build if it's stuck
Add timestamps to the Console Output
Build inside a Docker container
Inspect build log for published Gradle build scans
Set GitHub commit status with custom context and message (Must configure upstream job using GHPRB trigger)
With Ant

Build

Execute shell
Command
echo ghrbActualCommit = \${ghrbActualCommit}
echo ghrbActualCommitAuthor = \${ghrbActualCommitAuthor}
echo ghrbActualCommitAuthorEmail = \${ghrbActualCommitAuthorEmail}
echo ghrbPullDescription = \${ghrbPullDescription}
echo ghrbPullId = \${ghrbPullId}
echo ghrbPullLink = \${ghrbPullLink}
echo ghrbPullTitle = \${ghrbPullTitle}
echo ghrbSourceBranch = \${ghrbSourceBranch}
echo ghrbCommentBody = \${ghrbCommentBody}
echo sha1 = \${sha1}
echo BRANCH_NAME = \${BRANCH_NAME}
echo CHANGE_ID = \${CHANGE_ID}
echo CHANGE_URL = \${CHANGE_URL}
echo CHANGE_TITLE = \${CHANGE_TITLE}
echo CHANGE_AUTHOR = \${CHANGE_AUTHOR}
echo CHANGE_AUTHOR_DISPLAY_NAME = \${CHANGE_AUTHOR_DISPLAY_NAME}
echo CHANGE_AUTHOR_EMAIL = \${CHANGE_AUTHOR_EMAIL}
echo CHANGE_TARGET = \${CHANGE_TARGET}
echo CHANGE_BRANCH = \${CHANGE_BRANCH}
echo CHANGE_FORK = \${CHANGE_FORK}
echo TAG_NAME = \${TAG_NAME}
echo TAG_TIMESTAMP = \${TAG_TIMESTAMP}
echo TAG_URL = \${TAG_URL}
echo TAG_DATE = \${TAG_DATE}
echo BUILD_NUMBER = \${BUILD_NUMBER}
echo BUILD_ID = \${BUILD_ID}
echo BUILD_DISPLAY_NAME = \${BUILD_DISPLAY_NAME}
echo JOB_NAME = \${JOB_NAME}
echo JOB_BASE_NAME = \${JOB_BASE_NAME}
echo BUILD_TAG = \${BUILD_TAG}
echo EXECUTOR_NUMBER = \${EXECUTOR_NUMBER}
echo NODE_NAME = \${NODE_NAME}
echo NODE_LABELS = \${NODE_LABELS}
echo WORKSPACE = \${WORKSPACE}
echo WORKSPACE_TMP = \${WORKSPACE_TMP}
echo JENKINS_HOME = \${JENKINS_HOME}
echo JENKINS_URL = \${JENKINS_URL}
echo BUILD_URL = \${BUILD_URL}
echo JOB_URL = \${JOB_URL}
echo GIT_COMMIT = \${GIT_COMMIT}
echo GIT_PREVIOUS_COMMIT = \${GIT_PREVIOUS_COMMIT}
echo GIT_PREVIOUS_SUCCESSFUL_COMMIT = \${GIT_PREVIOUS_SUCCESSFUL_COMMIT}
echo GIT_BRANCH = \${GIT_BRANCH}
echo GIT_LOCAL_BRANCH = \${GIT_LOCAL_BRANCH}
echo GIT_CHECKOUT_DIR = \${GIT_CHECKOUT_DIR}
echo GIT_URL = \${GIT_URL}
echo GIT_COMMITTER_NAME = \${GIT_COMMITTER_NAME}
echo GIT_AUTHOR_NAME = \${GIT_AUTHOR_NAME}
echo GIT_COMMITTER_EMAIL = \${GIT_COMMITTER_EMAIL}
echo GIT_AUTHOR_EMAIL = \${GIT_AUTHOR_EMAIL}
bash .buildbot/z13.sh
See the list of available environment variables
Advanced...
Add build step

Post-build Actions
Add post-build action