



- [Skip to content](#)
- [Books](#)
- [Speaking](#)
- [Donate](#)
- [Contact](#)

Triggering Jenkins builds by URL

Posted at October 6, 2015 by Nicholas C. Zakas

Tags: [Jenkins](#), [Automation](#), [Builds](#)

As you might have read not too long ago, I recently moved my site from Wordpress to Jekyll[1]. In so doing, I ended up using Jenkins[2] to periodically build and upload my site to S3. Having a Jenkins instance running turns out to be quite useful for all sorts of tasks and so I've been trying to take advantage of it to automate more of my routine tasks.

The goals

Recently, I received some useful feedback from readers of my newsletter:

1. The default URL for archived newsletters looks suspicious. Here's an example: <http://us11.campaign-archive2.com/?u=06398d19c8d37d8d6ba9d9cf7&id=7ab9b1fc75>. Yeah, that looks pretty scary to anyone who knows anything about web security. I decided I'd try to figure out a way to generate an archive on my site.
2. Some folks wanted to subscribe via RSS instead of email. While Mailchimp does provide an RSS feed, it's of everything you send to a particular list. That means both my blog posts and newsletter updates are merged into the same feed, so I didn't have a feed of just newsletter updates. It seemed like having a feed could also help me with generating an archive page, so I decided this would be a nice first step.

Mailchimp supports webhooks, so it will post to any URL you want when it sends out an email. I wanted that to trigger generation of the archive page on my site so it would be available as soon as the email was sent. Jenkins allows you to trigger a build by posting to a URL, but it took me a little while to figure out exactly how to get this to work correctly.

Note: This assumes you're using Jenkins' own user database for user management rather than LDAP or another directory service.

Step 1: Setting up a new user

Trigger a build via URL means that the Jenkins endpoint is open to anyone who can hit the server. Naturally, that means you want to ensure you've secured this endpoint as much as possible and the first step is to create a user with limited access to Jenkins. To create a new user:

1. Click on **Manage Jenkins**
2. Click on **Manage Users**
3. Click on **Create User**
4. Fill in the information for your user (I'll assume you've called this user "auto")
5. Click the **Sign Up** button

This will take you back to the user list.

Step 2: Enable the URL job trigger

Go to the job that you want to trigger and click Configure to edit the job. Under Build Triggers, check the box next to "Trigger Builds Remotely". You'll be asked to provide a secure token for validation. This should not in any way be related to the "auto" user, so don't reuse the password. You might want to generate a new key using a tool like the Random Key Generator[3]. Click Save to save the job information.

Build Triggers

☒ Trigger builds remotely (e.g., from scripts)

Authentication Token

iFBDObhNhaxL4T9ass93HRXun2JF161Z

Use the following URL to trigger build remotely:
JENKINS_URL/job/Example/build?token=TOKEN_NAME or
/buildWithParameters?token=TOKEN_NAME
Optionally append &cause=Cause+Text to provide text that will be
included in the recorded build cause.

☐ Build after other projects are built

☐ Build periodically

☐ Build when a change is pushed to GitHub

☐ Poll SCM

Step 3: Enable permission for “auto”

In order to allow “auto” to trigger the build, the user needs to have the following permissions set:

- Overall - Read
- Job - Build
- Job - Read
- Job - Workspace

To configure these permissions:

1. Click on **Manage Jenkins**
2. Click on **Configure Global Security**
3. Assuming you’re using matrix-based security: add “auto” to the list and check off the boxes for the necessary permissions
4. Click **Save**




If you are not using matrix-based security, then this step is likely not necessary. Just make sure the “auto” user has the correct permissions.

Step 4: Create the URL

In order to make an external call using the “auto” user, you’ll need to use an API token. From the user list, click on the “configure” icon (the wrench and screw driver) next to the “auto” user.

Users

These users can log into Jenkins. This is a sub set of [this list](#), which also contains auto-created users who really just made some commits on some projects and have no direct Jenkins access.

User Id	Name	
 auto	auto	 

Underneath the user’s full name and description is a section labeled “API Token”. Click on the “Show API Token” button. This will reveal the API token you need to provide when triggering a job by URL.

Full Name ?

Description

API Token[Show API Token...](#)**API Token**

User ID

API Token ?

[Change API Token](#)

;

With this information, you can now create a URL that looks like this:

<http://auto:8702d1cb53a83f8748d9433ebca494fb@your-jenkins.com/job/JobName/build?token=iFBD0BhNhaxL4T9ass93HRXun2JF161Z>

Note that “JobName” should be replaced with the job name you’re triggering. The correctly escaped version of this URL is shown when you set the authentication token (step 2).

This URL has three key pieces of information:

1. The username “auto”
2. The API token (after the colon)
3. The authentication token

All three must be correct, and the permissions for the user must be correct in Jenkins, in order for this URL to trigger the job. In order to trigger the job, you must send a POST request. You can test it out using cURL:

```
curl -X POST http://auto:8702d1cb53a83f8748d9433ebca494fb@your-jenkins.com/job/JobName/build?token=iFBD0BhNhaxL4T9ass93HRXun2JF161Z
```

Once you send the request, log in to Jenkins and verify that the job is running. You can then use this URL for most webhooks.

Conclusion

I now have a job triggered remotely whenever Mailchimp sends an email. The script downloads the newsletter RSS feed, filters out blog posts, and generates an HTML archive page on my site. The most time-consuming part of the process was going setting up Jenkins to build via URL, as the resources I found online all had pieces of information missing. I hope that this post helps others make progress faster.

References

1. [From Wordpress to Jekyll: My new blog setup](#) by me (nczonline.net)
2. [Jenkins](#) (jenkins-ci.org)
3. [Random Key Generator](#) (randomkeygen.com)

Was this helpful?

If you found this post helpful, please consider donating to support my work.

- [Donate with GitHub](#)
- [Donate with PayPal](#)

New in the Store



Join the Mailing List

Never miss an update by joining 3,000 other mailing list members.

Enter your email address:

Subscribe

Comments for this thread are now closed



0 Comments

Human Who Codes

 Stephan K. Murphy ▾




 Recommend 69

 Tweet

 Share

Sort by Best ▾

This discussion has been closed.

 Subscribe  Add Disqus to your siteAdd DisqusAdd  Disqus' Privacy PolicyPrivacy PolicyPrivacy Policy

Additional Information



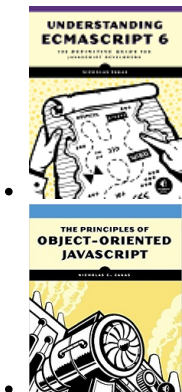
[Bring your team together with Slack, the collaboration hub for work.ads via Carbon](#)

Subscribe by Email

Enter your email address:

Subscribe

My Books





Recent Snippets

- [Posting to Medium using Node.js and fetch\(\)](#)
- [Special formatting for SVG images with CSS](#)
- [Flattening a JavaScript array with a generator](#)
- [Reading streams with promises in Node.js](#)
- [Setting up Visual Studio Code intellisense for Jest globals](#)

Archives

- [2019](#)
- [2018](#)
- [2016](#)
- [2015](#)
- [2014](#)
- [2013](#)
- [2012](#)
- [2011](#)
- [2010](#)
- [2009](#)
- [2008](#)
- [2007](#)
- [2006](#)
- [2005](#)
- [2004](#)



Follow 56.7K followers

About the Human

Hi, I'm Nicholas C. Zakas, an independent software developer living in Mountain View, California. I've been a software architect at companies like Yahoo and Box, as well as an author and speaker. I created the [ESLint](#) open source project and wrote several [books](#). At the moment, I'm unemployed as I'm [recovering from Lyme disease](#) and haven't been able to leave my home much in the past five years. ([Health update](#), [More about me](#))

On the Web

- [Twitter](#)
- [GitHub](#)
- [Patreon](#)
- [Instagram](#)
- [YouTube](#)
- [LinkedIn](#)
- [Slideshare](#)
- [Amazon](#)

Copyright © 2004-2019 Nicholas C. Zakas. All Rights Reserved. All code examples on all pages, unless otherwise indicated, are [BSD licensed](#). [Blog Feed](#)