

Home

PUBLIC

Stack Overflow

Tags

Users

FIND A JOB

Jobs

Companies

TEAMS

What's this?

Free 30 Day Trial

Pre-Authenticate Powershell WebClient requests to Team City 8.0 REST API

Ask Question

Asked 7 years, 3 months agoActive 7 years, 1 month agoViewed 1k times

I'm trying to run Powershell scripts in my Team City build steps.

2

The scripts use WebClient to connect to Team City's REST API; currently, I have to login to Team City and hardcode a username and password as arguments in my Powershell build step.

I'm wondering if anyone knows a way to pass the credentials I am currently using to authenticate to Team City in my Powershell scripts without hardcoding any passwords

restauthenticationpowershellwebclientteamcity-8.0

shareimprove this questionfollow

asked Aug 22 '13 at 14:25
ShaneC
2,06122652

If not hardcode them then when do you wish to supply Credentials to your powershell script? – Mitul Aug 22 '13 at 14:31

I'm trying to use the credentials I supplied to login to Team City, the only possible way any of these scripts can be run is through Team City web UI, after a username and password has been supplied to access the server – ShaneC Aug 22 '13 at 14:35

add a comment

2 Answers

ActiveOldestVotes

If you only need read access in the REST api (ie you don't want to do POST/PUT/DELETE, only GET) then use the teamcity generated user name and password.

4

This username/password pair is generated per each build and valid only during the build run. This is how you can access them in your powershell script:

- read the \$env:TEAMCITY_BUILD_PROPERTIES_FILE environment variable which holds the full path to the build properties file that are generated/valid for this build
- this file is a simple key=value java prop file. You need to parse out the values for teamcity.auth.userId and teamcity.auth.password properties. Or better yet, parse all the props always in your script init phase and put them into a hash table in your powershell script.

If you need write access to the REST api, you can't use this uid/pwd pair. For this I am using a keychain on osx and a keepass db on windows. Keepass has a nice .net api that you can access from powershell. Create an new keepass db, make it unlockable with a key, not with a password, make sure your user running the build agent has access to this key and no one else, then use keepass api to unlock the db, read out your teamcity admin account and password who can do POST/PUT/DELETE in the rest api.

shareimprove this answerfollow

answered Sep 14 '13 at 10:03
iwo
405413

1 Instead of parsing the properties file inside your script, you can also declare the \$username and \$password as powershell script params and pass them from the Build Step configuration page by the 'Script arguments' textbox like this: %system.teamcity.auth.userId% %system.teamcity.auth.password%. You can also pass the server url as %teamcity.serverUrl%. – Sergio Acosta Jan 15 '14 at 3:43

That's exactly what I wound up doing, at the time of posting the question, I was very unfamiliar with Team City and it's REST API. Most of the scripts I run through TC now use the same method for automatic authentication, some builds require a user to supply their own account name and password, as described here: blog.jetbrains.com/teamcity/2013/05/... – ShaneC Jan 15 '14 at 9:03

add a comment

Thanks for the answer but we wound up providing the username and password as build parameters.

1

TeamCity's built in password protection helped us out here.

In this way, we're using one account to run our powershell scripts but we can still see who kicked off the build from the credentials they used to login to the web UI.

So we maintained traceable responsibility and stopped the constant entering of username and passwords.

More info: confluence.jetbrains.com/display/TCD7/Typed+Parameters

shareimprove this answerfollow

answered Oct 14 '13 at 11:03
ShaneC
2,06122652

add a comment

Your Answer

B *I*

Sign up or log in

Sign up using Google

Sign up using Facebook

Sign up using Email and Password

Post Your Answer

By clicking "Post Your Answer", you agree to our terms of service, privacy policy and cookie policy

Post as a guest

Name

Email

Required, but never shown

Not the answer you're looking for? Browse other questions tagged rest authentication powershell webclient teamcity-8.0 or ask your own question.

The Overflow Blog

- How to write an effective developer resume: Advice from a hiring manager
- Podcast 290: This computer science degree is brought to you by Big Tech

Featured on Meta

- "Question closed" notifications experiment results and graduation
- MAINTENANCE WARNING: Possible downtime early morning Dec 2/4/9 UTC (8:30PM...)
- Congratulations VonC for reaching a million reputation

Related

- 189 Creating an API for mobile applications - Authentication and Authorization
- 117 RESTful web service - how to authenticate requests from other services?
- 304 Running a command as Administrator using PowerShell?
- 21 powershell http post REST API basic authentication
- 86 REST API for website which uses Facebook for authentication
- 1 Creating encrypted build parameter in TeamCity 8.0 from REST API
- 54 REST API authentication for web app and mobile app
- 10 Azure Websites Kudu REST API - Authentication
- 49 Using mail and password to authenticate via the REST API [Firebase]
- 6 Pass a Team City Parameter to a PowerShell file

Hot Network Questions

- Can I temporarily disable the @@ replacement in LaTeX3?
- Way to restore the data in the accidentally overwritten layer by its duplicate layer in QGIS?
- How much do propellers stutter?
- Connecting an axle to a stud on the ground for railings
- Why do some languages have genders and some don't?

more hot questions

Question feed



STACK OVERFLOW

- Questions
- Jobs
- Developer Jobs Directory
- Salary Calculator
- Help
- Mobile
- Disable Responsiveness

PRODUCTS

- Teams
- Talent
- Advertising
- Enterprise

COMPANY

- About
- Press
- Work Here
- Legal
- Privacy Policy
- Contact Us

STACK EXCHANGE NETWORK

- Technology
- Life / Arts
- Culture / Recreation
- Science
- Other

Blog Facebook Twitter LinkedIn Instagram