

Need advice of if this should be a...



[Joshua Kent](#) 1 posts since

Oct 15, 2009

I am in the process of developing a Automated Deployment application for ASP.NET and JAVA. I want to support teamcity as a build server but to do this seamlessly I would like to be able to easily determine a list of successful builds that have artifacts that can be deployed.

I don't see how I can do this with any of the data that TeamCity exposes via their "web-service like" calls.

Do you think it would be smart to build a plugin for TeamCity that would just expose this information to my deployment server?

Josh

Tags: deployment, plugins, web_service



[Yegor Yarko](#) 1,459 posts since

May 5, 2004 1. **Re: Need advice of if this should be a plugin** Oct 26, 2009 3:47 PM

Joshua,

You can create a Java plugin for TeamCity that exposes all the necessary information. This is the most capable approach, but also probably requiring most effort.

One more approach is to try using [Ivy](#)-compatible requests to get the list of available revisions from TeamCity.

Need advice of if this should be a...

Alos, currently we have a prototype of [REST API](#) plugin for TeamCity. It can expose the necessary information via REST, but current state of the plug-in is that the API will change a lot and thus it cannot be recommended for production use. The plugin may be included in TeamCity 5.0 release (there is no final decision yet). Anyway, the plugin can be used as an example of the code.

Do you need to list all the builds? Maybe last successful (last pinned) will be enough?

Is your tool a generic one? Another approach to deployment handling may be to put the deployment logic inside TeamCity (e.g. as one build build configuration). BTW, in TeamCity 5.0 there is a feature "promote a build" that allows to run another build configuraiton based on the current build.



[Jim Chaney](#) 5 posts since

Oct 23, 2009 2. **Re: Need advice of if this should be a plugin** Oct 26, 2009 1:58 PM

Alternatively a simple, and most likely very unsupported, method is to use direct access to the underlying storage engine (MySQL in our case) and write queries directly against the backend database. Once you have the IDs for the projects of interest, you can find the artifacts quite easily.

Of course, I only used this method because I know databases, and dont know what REST even means 😊