

Deactivate a User via REST

As the official API does not have this function (currently), we can accomplish it via [Adaptavist Scriptrunner](#).

Add a REST endpoint and use the code:

```
import com.onresolve.scriptrunner.runner.rest.common.CustomEndpointDelegate
import groovy.transform.BaseScript
import com.atlassian.jira.component.ComponentAccessor
import com.atlassian.jira.issue.Issue
import com.atlassian.jira.issue.IssueManager
import com.atlassian.jira.issue.issueType.IssueType
import com.atlassian.jira.project.ProjectManager
import com.atlassian.jira.issue.CustomFieldManager
import javax.ws.rs.core.MultivaluedMap
import javax.servlet.http.HttpServletRequest
import javax.ws.rs.core.Response
import com.atlassian.crowd.embedded.api.CrowdService
import com.atlassian.crowd.embedded.api.UserWithAttributes
import com.atlassian.crowd.embedded.impl.ImmutableUser
import com.atlassian.jira.bc.user.UserService
import com.atlassian.jira.component.ComponentAccessor
import com.atlassian.jira.user.ApplicationUser
import com.atlassian.jira.user.ApplicationUsers
import com.atlassian.jira.user.util.UserUtil

@BaseScript CustomEndpointDelegate delegate

deactivateuser(httpMethod: "GET", groups: ["jira-administrators"]) { MultivaluedMap queryParams ->

    def issueManager = ComponentAccessor.getIssueManager()
    def userManager = ComponentAccessor.getUserManager()

    String userName = queryParams.getFirst("username") as String
    def builder = new groovy.json.JsonBuilder()
        UserUtil userUtil = ComponentAccessor.userUtil
        CrowdService crowdService = ComponentAccessor.crowdService
        UserService userService = ComponentAccessor.getComponent(UserService)
        ApplicationUser updateUser
        UserService.UpdateUserValidationResult updateUserValidationResult

    // Get user
    UserWithAttributes user = crowdService.getUserWithAttributes(userName)

    if (user)
    {
        updateUser = ApplicationUsers.from(ImmutableUser.newUser(user).active(false).toUser())
        updateUserValidationResult = userService.validateUpdateUser(updateUser)
        if (updateUserValidationResult.isValid())
        {
            userService.updateUser(updateUserValidationResult)
            builder { result "User Deactivated" }
            return Response.status(201).entity(builder.toString()).build()
        }
        else
        {
            builder { result "Update of ${user.name} failed: ${updateUserValidationResult.
getErrorCollection().getErrors().entrySet().join(', ')}" }
            return Response.status(400).entity(builder.toString()).build()
        }
    }
    else
    {
        //No such username
        builder { result "User did not exist" }
        return Response.status(400).entity(builder.toString()).build()
    }
}
```

Testing is like:

```
curl -v -u username:password -H "X-Atlassian-Token: nocheck" -H "Content-Type: application/json" -X "GET" https://servicedesk.server.dk/rest/scriptrunner/latest/custom/deactivateuser?username=bnp@mos-eisley.dk
```

If successful, the return is a 200 OK:

```
{"result":"User Deactivated"}
```