Jira -> Azure Devops: Create new sprints in ADO that don't exist yet, with REST API and Exalate

Introduction

Hello all,

The scene:

You have set up a succesful integration between Jira and Azure Devops. You want to integrate Sprints aswell. In Jira (any deployment), the sprints already exist, but in Azure Devops, it could be that the sprints don't exist yet.

Here is where Exalate can help you out. With the use of Exalate, we can create the new sprints in Azure Devops.

The first thing we need to do is to send out all the information about a Sprint from the Jira outgoing side.

If we have a look at the data that will be sent out to Azure Devops, this is how it looks:

If we add this line of code in the Outgoing Sync in Jira;

Outgoing Sync Jira

```
replica.customFields."Sprint" = issue.customFields."Sprint"
```

Sprint Data

```
"customFields": {
      "Sprint": {
        "id": 10104,
        "name": "Sprint",
        "uid": "10104",
        "description": "Jira Software sprint field",
        "type": "SPRINTS",
        "value": [
          {
            "id": "2",
            "state": "FUTURE",
            "name": "SprintTwo",
            "originBoardId": "7",
            "startDate": 1697476800000,
            "endDate": 1698686400000,
            "sequence": 2,
            "goal": "This sprint does not exist yet in ADO",
            "eventTriggerContext": {}
          }
        ]
      }
    },
```

Exalate will use this information to create a Sprint in Azure Devops.

In Azure Devops we need to add this code to the incoming sync.

Please be mindful that you need to change the projectKey after implementing following code;

Incoming Sync Azure Devops

```
if(firstSync){
   // Set type name from source entity, if not found set a default
   workItem.projectKey  =  "Mathieu"
      def typeMap = [
       "Epic" : "Epic",
       "Story" : "User Story"
       ]
    workItem.typeName = nodeHelper.getIssueType(typeMap[replica.type?.name],workItem.projectKey)?.name ?: "Task"
    workItem.summary      = replica.summary
    store(issue)
}


if(workItem.typeName == "Task"){
workItem.summary      = replica.summary
workItem.description  = replica.description
workItem.attachments  = attachmentHelper.mergeAttachments(workItem, replica)
workItem.comments     = commentHelper.mergeComments(workItem, replica)
workItem.labels       = replica.labels
workItem.priority     = replica.priority
}


def getCurrentSprint = { -> replica."Sprint".find {!it.state.equalsIgnoreCase("CLOSED")} }

if (replica.customFields."Sprint"?.value != null && !replica.customFields."Sprint"?.value?.empty &&
getCurrentSprint() != null) {
  def project = connection.trackerSettings.fieldValues."project"
  def area = workItem.areaPath ?: workItem.project?.key ?: project
  def sprint = getCurrentSprint()
  def iteration = sprint.name
  def iterationPath = area + "\\" + iteration
  if (iterationPath != workItem.iterationPath) {
      def adoClient = new AdoClient(httpClient, nodeHelper, debug)
      def encode = {
    str ->
    if (!str) str
    else
     java.net.URLEncoder.encode(str, java.nio.charset.StandardCharsets.UTF_8.toString())
      }
        def projectName = workItem.project?.key ?: workItem.projectKey
      def existingIterations = adoClient
  .http (
      "GET",
      "/${encode(projectName)}/_apis/wit/classificationnodes/Iterations".toString(),
      ["api-version":["5.0"], "\$depth":["1"]],
      null,
      ["Accept":["application/json"]]
    ) { res ->
      if(res.code >= 400) debug.error("Failed to GET /${encode(projectName)}/_apis/work/teamsettings/iterations?
api-version=7.1-preview.1 RESPONSE: ${res.code} ${res.body}")
      else (new groovy.json.JsonSlurper()).parseText(res.body)
    }
```

```
  ?.children

      if (!existingIterations.name.any {it.equalsIgnoreCase(sprint.name)}) {
          //if we need to create iterations
          def await = { f -> scala.concurrent.Await$.MODULE$.result(f, scala.concurrent.duration.Duration.apply
(1, java.util.concurrent.TimeUnit.MINUTES)) }
          def creds = await(httpClient.azureClient.getCredentials())
          def token = creds.accessToken()
          def baseUrl = creds.issueTrackerUrl()

          def dateFormat = new java.text.SimpleDateFormat("yyyy-MM-dd'T'HH:mm:ss'Z'")
          def createIterationBody = [name:sprint.name]
          def attributes = null
          if(sprint.startDate) {
              def sd = dateFormat.format(sprint.startDate)
              attributes = ["startDate":sd]
          }
          if(sprint.endDate) {
              def ed = dateFormat.format(sprint.endDate)
              attributes = attributes ?: [:]
              attributes."finishDate" = ed
          }
          if(attributes != null) {
              createIterationBody."attributes" = attributes
          }
                def remoteIterationName = sprint.name
          def iterationId = adoClient
      .http (
        "POST",
        "/${encode(projectName)}/_apis/wit/classificationnodes/Iterations".toString(),
        ["api-version":["5.0"]],
        groovy.json.JsonOutput.toJson(["name": remoteIterationName]),
        ["Accept":["application/json"], "Content-Type":["application/json"]]
      ) { res ->
      if(res.code >= 400) debug.error("POST ${encode(projectName)}/_apis/wit/classificationnodes/Iterations?api-
version=5.0 failed: ${res.code} ${res.body}")
      else (new groovy.json.JsonSlurper()).parseText(res.body)
    }?."identifier"
    //and associate it with the team
    adoClient
      .http (
        "POST",
        "/${encode(projectName)}/_apis/work/teamsettings/iterations".toString(),
        ["api-version":["7.1-preview.1"]],
        groovy.json.JsonOutput.toJson(["id": iterationId]),
        ["Accept":["application/json"], "Content-Type":["application/json"]]
      ) { res ->
      if(res.code >= 400) debug.error("POST ${encode(projectName)}/_apis/work/teamsettings/iterations failed:
${res.code} ${res.body}")
      else (new groovy.json.JsonSlurper()).parseText(res.body)
    }
      }
      workItem.iterationPath = iterationPath
  }

}

class AdoClient {
    // SCALA HELPERS
    private static <T> T await(scala.concurrent.Future<T> f) {
        scala.concurrent.Await$.MODULE$.result(f, scala.concurrent.duration.Duration$.MODULE$.Inf())
    }
    private static <T> T orNull(scala.Option<T> opt) { opt.isDefined() ? opt.get() : null }
    private static <T> scala.Option<T> none() { scala.Option$.MODULE$.<T> empty() }
    @SuppressWarnings("GroovyUnusedDeclaration")
    private static <T> scala.Option<T> none(Class<T> evidence) { scala.Option$.MODULE$.<T> empty() }
    private static <L, R> scala.Tuple2<L, R> pair(L l, R r) { scala.Tuple2$.MODULE$.<L, R> apply(l, r) }
    // SERVICES AND EXALATE API
    private static def getGeneralSettings() {
        def gsOptFuture = nodeHelper.azureClient.generalSettingsService.get()
        def gsOpt = await(gsOptFuture)
```

```
        def gs = orNull(gsOpt)
        gs
    }
    private static String getIssueTrackerUrl() {
        final def gs = getGeneralSettings()
        def removeTailingSlash = { String str -> str.trim().replace("/+\$", "") }
        final def issueTrackerUrl = removeTailingSlash(gs.issueTrackerUrl)
        issueTrackerUrl
    }
    private httpClient
    private static nodeHelper
    private debug
    def parseQueryString = { String string ->
        string.split('&').collectEntries { param ->
            param.split('=', 2).collect { URLDecoder.decode(it, 'UTF-8') }
        }
    }
    //Usage examples: https://gist.github.com/treyturner/4c0f609677cbab7cef9f
    def parseUri
    {
        parseUri = { String uri ->
            def parsedUri
            try {
                parsedUri = new URI(uri)
                if (parsedUri.scheme == 'mailto') {
                    def schemeSpecificPartList = parsedUri.schemeSpecificPart.split('\\?', 2)
                    def tempMailMap = parseQueryString(schemeSpecificPartList[1])
                    parsedUri.metaClass.mailMap = [
                            recipient: schemeSpecificPartList[0],
                            cc       : tempMailMap.find { it.key.toLowerCase() == 'cc' }.value,
                            bcc      : tempMailMap.find { it.key.toLowerCase() == 'bcc' }.value,
                            subject  : tempMailMap.find { it.key.toLowerCase() == 'subject' }.value,
                            body     : tempMailMap.find { it.key.toLowerCase() == 'body' }.value
                    ]
                }
                if (parsedUri.fragment?.contains('?')) { // handle both fragment and query string
                    parsedUri.metaClass.rawQuery = parsedUri.rawFragment.split('\\?')[1]
                    parsedUri.metaClass.query = parsedUri.fragment.split('\\?')[1]
                    parsedUri.metaClass.rawFragment = parsedUri.rawFragment.split('\\?')[0]
                    parsedUri.metaClass.fragment = parsedUri.fragment.split('\\?')[0]
                }
                if (parsedUri.rawQuery) {
                    parsedUri.metaClass.queryMap = parseQueryString(parsedUri.rawQuery)
                } else {
                    parsedUri.metaClass.queryMap = null
                }
                if (parsedUri.queryMap) {
                    parsedUri.queryMap.keySet().each { key ->
                        def value = parsedUri.queryMap[key]
                        if (value.startsWith('http') || value.startsWith('/')) {
                            parsedUri.queryMap[key] = parseUri(value)
                        }
                    }
                }
            } catch (e) {
                throw new com.exalate.api.exception.IssueTrackerException("Parsing of URI failed: $uri $e ", e)
            }
            parsedUri
        }
    }
    AdoClient(httpClient, nodeHelper, debug) {
        this.httpClient = httpClient
        this.nodeHelper = nodeHelper
        this.debug = debug
    }
    String http(String method, String path, java.util.Map<String, List<String>> queryParams, String body, java.
util.Map<String, List<String>> headers) {
        http(method, path, queryParams, body, headers) { Response response ->
            if (response.code >= 300) {
                throw new com.exalate.api.exception.IssueTrackerException(
                        """Failed to perform the request $method $path (status ${response.code}),
```

```
and body was: ```$body```
Please contact Exalate Support: """.toString() + response.body
                )
            }
            response.body as String
        }
    }
    public <R> R http(String method, String path, java.util.Map<String, List<String>> queryParams, String body,
java.util.Map<String, List<String>> headers, Closure<R> transformResponseFn) {
        def gs = getGeneralSettings()
        def unsanitizedUrl = issueTrackerUrl + path
        def parsedUri = parseUri(unsanitizedUrl)
        def embeddedQueryParams = parsedUri.queryMap
        def allQueryParams = embeddedQueryParams instanceof java.util.Map ?
                ({
                    def m = [:] as java.util.Map<String, List<String>>;
                    m.putAll(embeddedQueryParams as java.util.Map<String, List<String>>)
                    m.putAll(queryParams)
                })()
                : (queryParams ?: [:] as java.util.Map<String, List<String>>)
        def urlWithoutQueryParams = { String url ->
            URI uri = new URI(url)
            new URI(uri.getScheme(),
                    uri.getUserInfo(), uri.getHost(), uri.getPort(),
                    uri.getPath(),
                    null, // Ignore the query part of the input url
                    uri.getFragment()).toString()
        }
        def sanitizedUrl = urlWithoutQueryParams(unsanitizedUrl)
        //debug.error("#debug ${sanitizedUrl}")
        def response
        try {
            def request = ({
                try { httpClient.azureClient }
                catch (e) { httpClient.issueTrackerClient }
              })()
              .ws
              .url(sanitizedUrl)
              .withMethod(method)
            if (!allQueryParams.isEmpty()) {
                def scalaQueryParams = scala.collection.JavaConversions.asScalaBuffer(
                        queryParams
                                .entrySet()
                                .inject([] as List<scala.Tuple2<String, String>>) { List<scala.Tuple2<String,
String>> result, kv ->
                                    kv.value.each { v -> result.add(pair(kv.key, v) as scala.Tuple2<String,
String>) }
                                    result
                                }
                )
                request = request.withQueryString(scalaQueryParams)
            }
            if (headers != null && !headers.isEmpty()) {
                def scalaHeaders = scala.collection.JavaConversions.asScalaBuffer(
                        headers
                                .entrySet()
                                .inject([] as List<scala.Tuple2<String, String>>) { List<scala.Tuple2<String,
String>> result, kv ->
                                    kv.value.each { v -> result.add(pair(kv.key, v) as scala.Tuple2<String,
String>) }
                                    result
                                }
                )
                request = request.withHeaders(scalaHeaders)
            }
            if (body != null) {
                def writable = play.api.libs.ws.WSBodyWritables$.MODULE$.writeableOf_String()
                request = request.withBody(body, writable)
            }
            def credentials = await(httpClient.azureClient.credentials)
            def token = credentials.accessToken
```

```
            //debug.error("${play.api.libs.ws.WSAuthScheme$.class.code}")
            request = request.withAuth(token, token, play.api.libs.ws.WSAuthScheme$BASIC$.MODULE$)
            response = await(request.execute())
        } catch (Exception e) {
            throw new com.exalate.api.exception.IssueTrackerException(
                    """Unable to perform the request $method $path with body:```$body```,
please contact Exalate Support: """.toString() + e.message,
                    e
            )
        }
        java.util.Map<String, List<String>> javaMap = [:]
        for (scala.Tuple2<String, scala.collection.Seq<String>> headerTuple : scala.collection.JavaConverters.
bufferAsJavaListConverter(response.allHeaders().toBuffer()).asJava()) {
            def javaList = []
            javaList.addAll(scala.collection.JavaConverters.bufferAsJavaListConverter(headerTuple._2().
toBuffer()).asJava())
            javaMap[headerTuple._1()] = javaList
        }
        def javaResponse = new Response(response.body(), new Integer(response.status()), javaMap)
        return transformResponseFn(javaResponse)
    }
    public static class Response {
        final String body
        final Integer code
        final java.util.Map<String, List<String>> headers
        Response(String body, Integer code, java.util.Map<String, List<String>> headers) {
            this.body = body
            this.code = code
            this.headers = headers
        }
    }
}
```

Here is a video;

Thank you and happy exalating.

Questions

Recent Questions

Ask a question

1. 0
   votes
   Lifecycle blocking error

- **1** answer
- Jozsef Lehocz
- May 13, 2024
- Space: Exalate
- exalate
- blocking
- error
- upgrade

2. 0
votes
How to sync issues from two Jira cloud sites to one? Is this possible?
- **1** answer
- Vedant Kulkarni
- Mar 27, 2024
- Space: Exalate
- jira-cloud
- connector-cloud-jira

3. 0
votes
Issues are not syncing to local destination desk
- **1** answer
- John Lombardo
- Mar 21, 2024
- Space: Exalate
- exalate
- connector-cloud-jira
- jira-cloud

4. 0
votes
Not getting the Custom Field sync value from JIRA to Zendesk
- **1** answer
- Dinesh Gupalan
- Mar 18, 2024
- Space: Exalate
- connector-onpremise-jira-zendesk

5. 0
votes
Need to split source comment into multiple destination comments. (The entered text is too long.)
- **1** answer
- Kyle K
- Feb 14, 2024
- Space: Exalate
- exalate
- jira-cloud
- comments

6. 0
votes
exalate plugin of jira server
- **1** answer
- Sreenivasaraju P
- Feb 08, 2024
- Space: Exalate
- exalate

7. 0
votes
How to have keys match between jira cloud instances
- **2** answers
- Chris Matthews
- Feb 07, 2024
- Space: Exalate
- connector-cloud-jira

8. 0
votes
how decrease dimension of /var/atlassian/application-data/jira/data/exalate
- **1** answer
- Francesco Doricchi
- Dec 14, 2023
- Space: Exalate
- exalate

9. 1
vote
How to Impersonate Attachments in Jira Cloud?
- **0** answers
- Valeriia Solianikova
- Dec 12, 2023
- Space: Exalate

- exalate
- jira
- attachments

10. 0
votes
status
How to sync field that has been added to an existing connection?
- **2** answers
- Daniel
- Dec 03, 2023
- Space: Exalate
- connector-onpremise-jira
- connector-cloud-salesforce

Ask a question