

How to Access a Workload Attribute Provided by Tempo in Jira On-premise

Last Modified on 01/15/2026 6:27 pm EST

This article shows how to access the services by Tempo.

The use case is to extract attribute values from a worklog and make decisions from them.

Introduction

Tempo on Jira on-premise has a Java API. It is not public but still straightforward to use. A good inspiration about how to use the API can be found in [this library](#).

The real challenge is to get to the service which provides access to that information. For more information, please read [How to sync Tempo worklog attributes](#).

Accessing the Worklogattribute Service

As an example - access the workAttributeService

```
// don't forget the import - it belongs at the start of the processor
import com.atlassian.jira.component.ComponentAccessor

/*
** First look up the attribute id for account through the workAttributeService
**
** To get access to this service - lookup the class using the findClass method
** and then use the getOSGiComponentInstanceOfType to get the service itself
** (the throw is a handy trick to see if you get the results you expect
*/
def workAttributeServiceClass = ComponentAccessor.pluginAccessor.classLoader.findClass("com.tempoplugin.core.workattribute.api.WorkAttributeService")
def workAttributeService = ComponentAccessor.getOSGiComponentInstanceOfType(workAttributeServiceClass)
//throw new Exception ("Work AttributeService - ${workAttributeService.properties}")

// Get the account attribute
def accountAttribute = workAttributeService.getWorkAttributeByKey("_Account_").returnedValue
```

Accessing the Workload Attribute

Getting the value of the Tempo Worklog Attribute is then as follows

```

/*
** Then look up the value of the attribute using the workAttributeValueService
** the approach is the same (lookup class, get the instance)
*/

def workAttributeValueServiceClass = ComponentAccessor.pluginAccessor.classLoader.findClass("com.tempoplugin.core.workattribute.api.WorkAttributeValueService")
def workAttributeValueService = ComponentAccessor.getOSGiComponentInstanceOfType(workAttributeValueServiceClass)
def worklogs = issue.workLogs

//throw new Exception ("Work AttributeService - ${workAttributeValueService.properties}")

def worklogAccount = [:]

issue.workLogs.each {

    worklog ->

        def attributeValue = workAttributeValueService.getWorkAttributeValueByWorklogAndWorkAttribute(worklog.id, accountAttribute.id).returnedValue
    }
}

```

Product

[About Us](#) 

ON THIS PAGE

[Release History](#) 

[Glossary](#) 

[Introduction](#)

[API Reference](#) 

[Accessing the Worklogattribute Service](#)

[Pricing and Licensing](#) 

Resources

[Subscribe for a weekly Exalate hack](#) 

[Academy](#) 

[Blog](#) 

[YouTube Channel](#) 

[Ebooks](#) 

Still need help?

[Join our Community](#) 

[Visit our Service Desk](#) 

[Find a Partner](#) 