**Hygieia Collector Configuration**

**Jira Collector**:

* Go to Hygeia Folder Open with cmd
* Type cd \collectors\feature\jira
* Create application.properties file with the following fields

**# Database Name**

**dbname=dashboard**

**# Database HostName - default is localhost**

**dbhost=localhost**

**# Database Port - default is 27017**

**dbport=27017**

**# MongoDB replicaset**

**dbreplicaset=false**

**#dbhostport=[localhost:27017,127.0.0.1:27017,127.0.0.:27017]**

**# Database Username - default is blank**

**dbusername=dashboarduser**

**# Database Password - default is blank**

**dbpassword=dbpassword**

**# Logging File location**

**logging.file=./logs/jira.log**

**# PageSize - Expand contract this value depending on Jira implementation's**

**# default server timeout setting (You will likely receive a SocketTimeoutException)**

**feature.pageSize=100**

**feature.sprintDays=60**

**feature.sprintEndPrior=7**

**feature.scheduledPriorMin=2**

**# Delta change date that modulates the collector item task**

**# Occasionally, these values should be modified if database size is a concern**

**feature.deltaStartDate=2019-03-01T00:00:00.000000**

**feature.masterStartDate=2019-03-01T00:00:00.000000**

**feature.deltaCollectorItemStartDate=2019-03-01T00:00:00.000000**

**# Chron schedule: S M D M Y [Day of the Week]**

**feature.cron=0/5 \* \* \* \* \***

**# ST Query File Details - Required, but DO NOT MODIFY**

**feature.queryFolder=jiraapi-queries**

**feature.storyQuery=story**

**feature.epicQuery=epic**

**# JIRA CONNECTION DETAILS:**

**# Enterprise Proxy - ONLY INCLUDE IF YOU HAVE A PROXY**

**feature.jiraBaseUrl=http://10.7.176.92:9099/**

**feature.jiraQueryEndpoint=rest/api/2/**

**# For basic authentication, requires username:password as string in base64**

**# This command will make this for you: echo -n username:password | base64**

**feature.jiraCredentials=c3VkaGVlcmlhdDpzdWRoZWVy**

**# OAuth is not fully implemented; please blank-out the OAuth values:**

**feature.jiraOauthAuthtoken=**

**feature.jiraOauthRefreshtoken=**

**feature.jiraOauthRedirecturi=**

**feature.jiraOauthExpiretime=**

**#############################################################################**

**# In Jira, general IssueType IDs are associated to various 'issue'**

**# attributes. However, there is one attribute which this collector's**

**# queries rely on that change between different instantiations of Jira.**

**# Please provide a string name reference to your instance's IssueType for**

**# the lowest level of Issues (for example, 'user story') specific to your Jira**

**# instance. Note: You can retrieve your instance's IssueType Name**

**# listings via the following URI: https://[your-jira-domain-name]/rest/api/2/issuetype/**

**# Multiple comma-separated values can be specified.**

**#############################################################################**

**feature.jiraIssueTypeNames[0]=Story**

**feature.jiraStoryId=10002**

**#EpicId**

**feature.jiraEpicId=10000**

**#############################################################################**

**# In Jira, your instance will have its own custom field created for 'sprint' or 'timebox' details,**

**# which includes a list of information. This field allows you to specify that data field for your**

**# instance of Jira. Note: You can retrieve your instance's sprint data field name**

**# via the following URI, and look for a package name com.atlassian.greenhopper.service.sprint.Sprint;**

**# your custom field name describes the values in this field:**

**# https://[your-jira-domain-name]/rest/api/2/issue/[some-issue-name]**

**#############################################################################**

**feature.jiraSprintDataFieldName=customfield\_10001**

**#############################################################################**

**# In Jira, your instance will have its own custom field created for 'super story' or 'epic' back-end ID,**

**# which includes a list of information. This field allows you to specify that data field for your instance**

**# of Jira. Note: You can retrieve your instance's epic ID field name via the following URI where your**

**# queried user story issue has a super issue (for example, epic) tied to it; your custom field name describes the**

**# epic value you expect to see, and is the only field that does this for a given issue:**

**# https://[your-jira-domain-name]/rest/api/2/issue/[some-issue-name]**

**#############################################################################**

**#feature.jiraEpicIdFieldName=customfield\_10018**

**#############################################################################**

**# In Jira, your instance will have its own custom field created for 'story points'**

**# This field allows you to specify that data field for your instance**

**# of Jira. Note: You can retrieve your instance's storypoints ID field name via the following URI where your**

**# queried user story issue has story points set on it; your custom field name describes the**

**# story points value you expect to see:**

**# https://[your-jira-domain-name]/rest/api/2/issue/[some-issue-name]**

**#############################################################################**

**feature.jiraStoryPointsFieldName=customfield\_10006**

**#############################################################################**

**# In Jira, your instance will have its own custom field created for 'team'**

**# This field allows you to specify that data field for your instance**

**# of Jira. Note: You can retrieve your instance's team ID field name via the following URI where your**

**# queried user story issue has team set on it; your custom field name describes the**

**# team value you expect to see:**

**# https://[your-jira-domain-name]/rest/api/2/issue/[some-issue-name]**

**#############################################################################**

**feature.jiraTeamFieldName=**

**# Set this to true if you use boards as team**

**feature.jiraBoardAsTeam=false**

**Jenkins Collector:**

* Go to collectors/build/Jenkins and open with cmd
* Create application.properties with the following properties

**# Database Name**

**dbname=dashboard**

**# Database HostName - default is localhost**

**dbhost=localhost**

**# Database Port - default is 27017**

**#dbport=9999**

**# Database Username - default is blank**

**dbusername=dashboarduser**

**# Database Password - default is blank**

**dbpassword=dbpassword**

**# Collector schedule (required)**

**jenkins.cron= ex: 0 0/1 \* \* \* \***

**# The page size**

**jenkins.pageSize=1000**

**# The folder depth - default is 10**

**jenkins.folderDepth=10**

**# Jenkins server (required) - Can provide multiple**

**jenkins.servers[0]=Server:port ex: (**[**http://localhost:8084**](http://localhost:8084)**)**

**# If using username/token for API authentication**

**# (required for Cloudbees Jenkins Ops Center) For example,**

**#jenkins.servers[1]=http://gopi:1156378e6e511358522f39b0966c4f054f@localhost:8084**

**# Another option: If using same username/password Jenkins auth,**

**# set username/apiKey to use HTTP Basic Auth (blank=no auth)**

**#jenkins.usernames[0]=**

**#jenkins.apiKeys[0]=**

**# Determines if build console log is collected - defaults to false**

**jenkins.saveLog=true**

**# Search criteria enabled via properties (max search criteria = 2)**

**jenkins.searchFields[0]= options.jobName**

**jenkins.searchFields[1]= niceName**

**SONARQUBE COLLECTOR:**

* Go to collectors/build/sonar and open with cmd
* Create application.properties with the following properties

**# Database Name**

**dbname=dashboard**

**# Database HostName - default is localhost**

**dbhost=localhost**

**# Database Port - default is 27017**

**dbport=27017**

**# MongoDB replicaset**

**#dbreplicaset=[false if you are not using MongoDB replicaset]**

**#dbhostport=[host1:port1,host2:port2,host3:port3]**

**# Database Username - default is blank**

**dbusername=dashboarduser**

**# Database Password - default is blank**

**dbpassword=dbpassword**

**# Collector schedule (required)**

**sonar.cron=0/1 \* \* \* \* \***

**# Sonar server(s) (required) - Can provide multiple**

**sonar.servers[0]=http://localhost:9000**

**# Sonar version, match array index to the server. If not set, will default to version prior to 6.3.**

**sonar.versions[0]=6.7**

**# Sonar Metrics - Required.**

**#Sonar versions lesser than 6.3**

**#sonar.metrics[0]=ncloc,line\_coverage,violations,critical\_violations,major\_ violations,blocker\_violations,violations\_density,sqale\_index,test\_success\_density,test\_failures,test\_errors,tests**

**# For Sonar version 6.3 and above**

**sonar.metrics[0]=ncloc,violations,new\_vulnerabilities,critical\_violations,major\_violations,blocker\_violations,tests,test\_success\_density,test\_errors,test\_failures,coverage,line\_coverage,sqale\_index,alert\_status,quality\_gate\_details**

**# Sonar login credentials**

**sonar.username=admin**

**sonar.password=admin**

**GitHub Collector:**

* Go to collectors/scm/github and open with cmd
* Create application.properties file with the following properties

**# Database Name**

**dbname=dashboard**

**# Database HostName - default is localhost**

**dbhost=localhost**

**# Database Port - default is 27017**

**dbport=27017**

**# MongoDB replicaset**

**#dbreplicaset=[false if you are not using MongoDB replicaset]**

**#dbhostport=[host1:port1,host2:port2,host3:port3]**

**# Database Username - default is blank**

**dbusername=dashboarduser**

**# Database Password - default is blank**

**dbpassword=dbpassword**

**# Logging File location**

**logging.file=./logs/github.log**

**# Collector schedule (required)**

**github.cron=0 0/1 \* \* \* \***

**github.host=github.com**

**# Maximum number of previous days from current date, when fetching commits**

**github.commitThresholdDays=15**

**# Optional: Error threshold count after which collector stops collecting for a collector item. Default is 2.**

**github.errorThreshold=1**

**# This is the key generated using the Encryption class in core (optional for public repos)**

**github.key=zlLmdj5RvFHyYVKoid/Ts9OD6nqDBJ68**

**# Personal access token generated from github and used for making authentiated calls(optional for public repos)**

**github.personalAccessToken=cf13338bc28ec4b71c362f24898a93a1b7c8a00c**

**# Github repository Connect Timeout value in milliseconds, default value is 20000 (20s)**

**#github.connectTimeout=**

**# Github repository Read Timeout value in milliseconds, default value is 20000 (20s)**

**#github.readTimeout=**

**Gitlab Collector:**

* Go to collectors/build/scm/gitlab and open with cmd
* Create application.properties with the following properties

**# Database Name**

**dbname=dashboard**

**# Database HostName - default is localhost**

**dbhost=localhost**

**# Database Port - default is 27017**

**dbport=27017**

**# MongoDB replicaset**

**dbreplicaset=false**

**#dbhostport=[host1:port1,host2:port2,host3:port3]**

**# Database Username - default is blank**

**dbusername=dashboarduser**

**# Database Password - default is blank**

**dbpassword=dbpassword**

**# Logging File location**

**logging.file=./logs/gitlab.log**

**#Collector schedule (required)**

**gitlab.cron=0/1 \* \* \* \* \***

**#Gitlab host (optional, defaults to 'gitlab.com')**

**gitlab.host=code.swecha.org**

**#Gitlab protocol (optional, defaults to 'http')**

**gitlab.protocol=http**

**#Gitlab port (optional, defaults to protocol default port)**

**#gitlab.port=80**

**#Gitlab path (optional, if your instance of gitlab requires a path)**

**#gitlab.path=/gitlab/resides/here**

**#If your instance of Gitlab is using a self-signed certificate, set to true, default is false**

**gitlab.selfSignedCertificate=true**

**#Gitlab API Token (required, user token the collector will use by default, can be overridden on a per repo basis from the UI. API token provided by Gitlab)**

**gitlab.apiToken=rY6SoAozyzL-\_zJ\_zkzr**

**#Maximum number of previous days from current date, when fetching commits**

**gitlab.commitThresholdDays=15**

**# Gitlab key for private repos**

**#gitlab.key=a4:5f:be:73:10:7c:9c:2c:cc:09:ff:b8:2d:8e:80:3d**

**Reportportal Collector:**

* Go to collectors/testresults/reportportal and open with cmd
* Create application.properties file with the following properties

**# Database Name**

**dbname=dashboard**

**# Database HostName - default is localhost**

**dbhost=localhost**

**# Database Port - default is 27017**

**dbport=27017**

**# MongoDB replicaset**

**#dbreplicaset=[false if you are not using MongoDB replicaset]**

**#dbhostport=[host1:port1,host2:port2,host3:port3]**

**# Database Username - default is blank**

**dbusername=dashboarduser**

**# Database Password - default is blank**

**dbpassword=dbpassword**

**# Collector schedule (required)**

**reportportal.cron=0/5 \* \* \* \* \***

**# server(s) (required) - Can provide multiple**

**reportportal.servers[0]=http://10.7.177.73:8084**

**# reportportal login credentials**

**reportportal.username=sathya**

**reportportal.password=sathya**

**#reportportal bearerToken**

**reportportal.bearerToken=38ec14bc-0196-4499-9c41-93c78e9d1792**

**#reportportal projectName**

**reportportal.projectName=pof\_addressbook**

**Running Collectors:**

**After creating application.properties file run the following commands to run each collector**

* **Open cmd**
* **Go to target folder in each collector and run the below**

**Jira collector:**

**Java -jar jira-feature-collector.jar --spring.config.name=feature --spring.config.location=path- to\application.properties**

**Jenkins Collector:**

**java -jar jenkins-build-collector-3.0.1.jar --spring.config.name=jenkins --spring.config.location=C:\SW\hygiea-setup\Hygieia-3.0.2\collectors\build\jenkins\src\main\resources\application.properties**

**SonarQube Collector:**

**java –jar sonar-codequality-collector-3.0.1.jar --spring.config.name=sonar --spring.config.location=path-to\application.properties**

**Github Collector:**

**java -jar github-scm-collector-3.0.1.jar --spring.config.name=github --spring.config.location=path-to\collectors\scm\github\application.properties**

**Gitlab Collector:**

**java -jar gitlab-scm-collector-3.0.1.jar --spring.config.name=gitlab --spring.config.location=path-to\collectors\scm\gitlab\application.properties**

**ReportPortal Collector:**

**java -jar gitlab-scm-collector-3.0.1.jar --spring.config.name=gitlab --spring.config.location=path-to\collectors\scm\gitlab\application.properties**