

JiraTriggerHelper <i>handle trigger operations</i>
+
+ processProjectAfterInsert(List<Jira_Project__c>): void + processIssueAfterInsert(List<Jira_Issue__c>): void

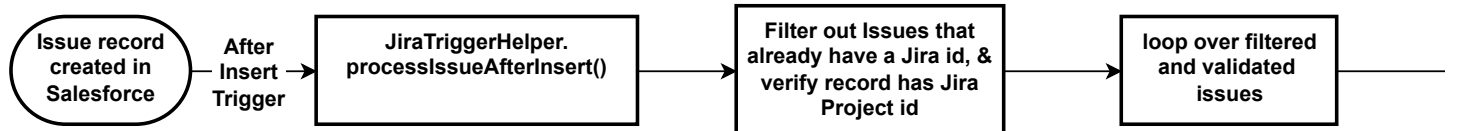
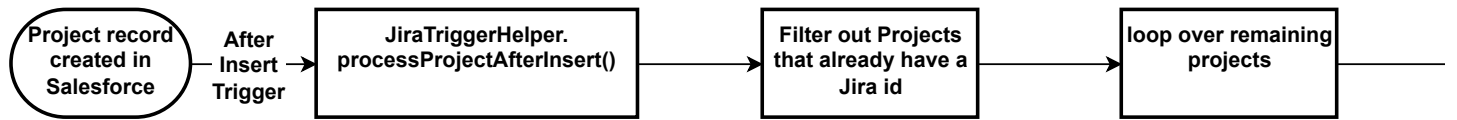
JiraCalloutQueueable <i>make async callouts</i>
+ <u>TYPE_PROJECT</u> : String + <u>TYPE_ISSUE</u> : String - calloutType: String - recordId: Id - projectWrapper: JiraWrapper.ProjectWrapper - issueWrapper: JiraWrapper.IssueWrapper
+ JiraCalloutQueueable(Id, String, String, String): void <i>project constructor</i> + JiraCalloutQueueable(Id, String, String, String, String): void <i>issue constructor</i> + execute(QueueableContext): void - updateJiraProject(String): void <i>updates sObject with Jira id</i> - updateJiraIssue(String): void <i>updates sObject with Jira id</i>

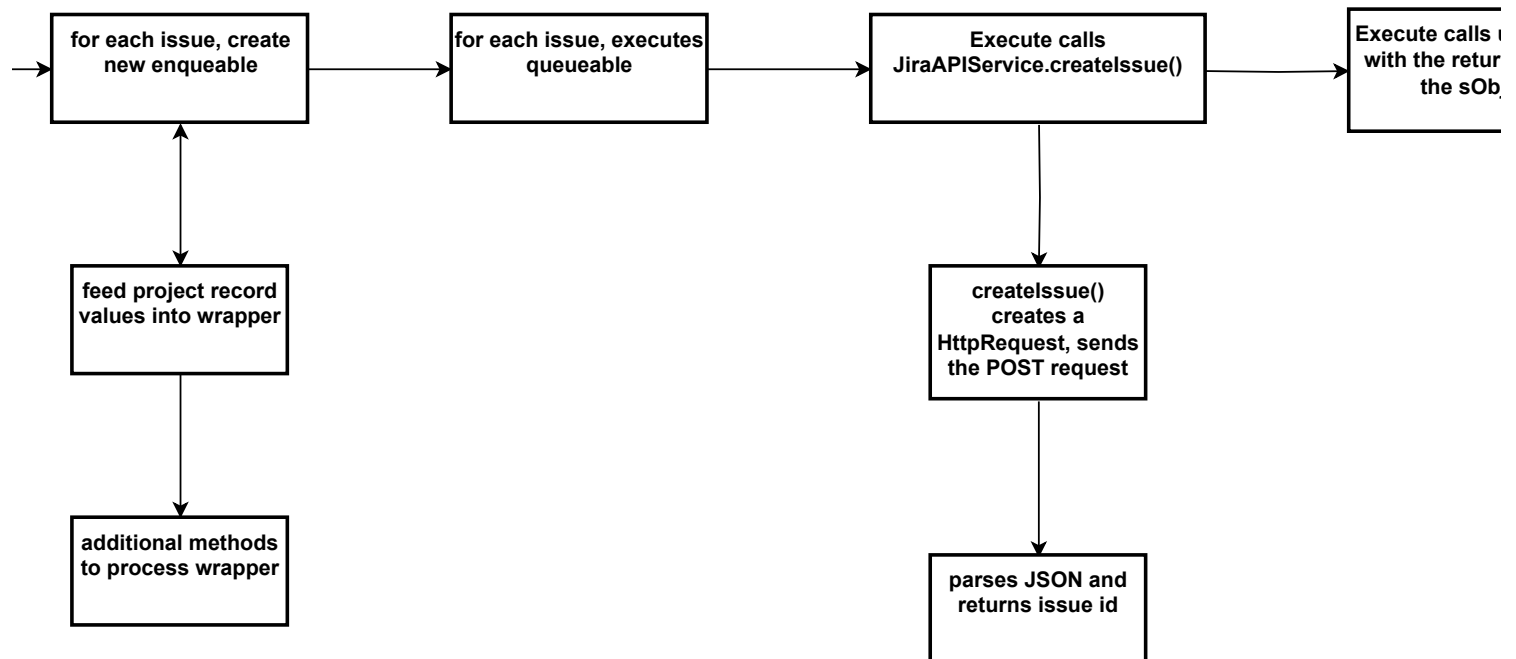
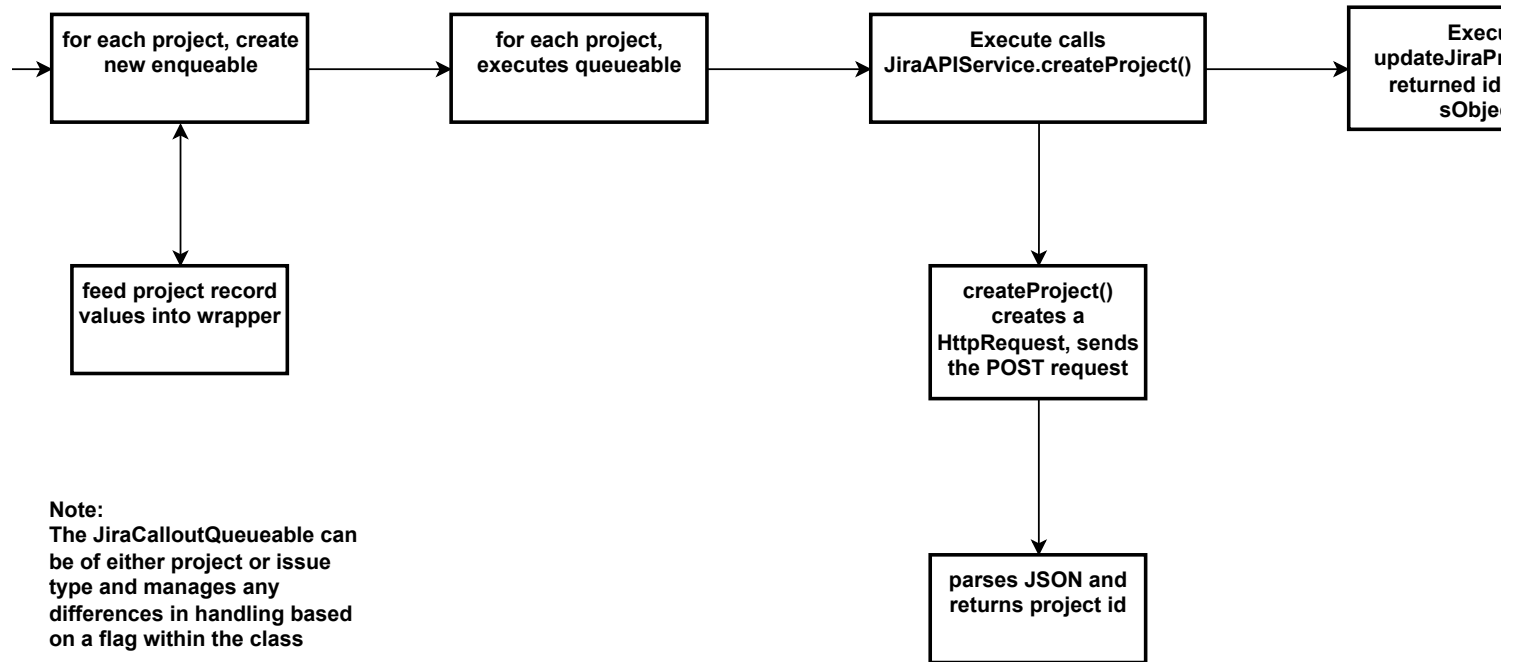
JiraAPIService <i>for making callouts to jira API</i>
+ createProject(JiraWrapper.ProjectWrapper): String + createIssue(JiraWrapper.IssueWrapper): String

JiraWrapper <i>for handling Jira payloads</i>
- <u>LEAD_ACCOUNT_ID</u> : String - <u>PROJECT_TEMPLATE_KEY</u> : String

ProjectWrapper <i>wrapper for Jira Project</i>
+ name: String + key: String + description: String + leadAccountId: String + projectTemplateKey: String
+ ProjectWrapper(String, String, String): void

IssueWrapper <i>wrapper for Jira Issue</i>
+ name: String + key: String + description: String + projectKey: String + summary: String + issueType: String
+ IssueWrapper(String, String, String, String): void





ute calls
roject() with the
l to update the
ct record

updateJiralssue()
ned id to update
ject record