

## Data Extractors and Operators in SA-FLY

### A generic overview of the commonly utilized

**Get Commits Data Extractor:** As of now 4 GET extractors are used to gather data from repositories. These are GET Github Commits , GET Bitbucket Commits , GET Builds , GET Issues and GET Metrics extractors. Figure 1 shows the GET COMMITS data extractor.

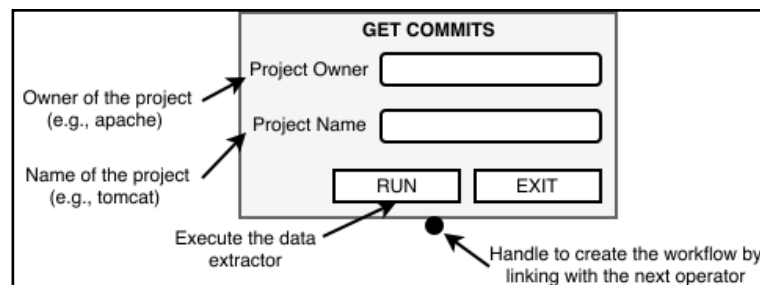


Figure 1 : GET COMMITS Data Extractor

**FOR EACH Operator:** The data integration logic of the query interface depends on the JSON data structure provided by various heterogeneous repositories. Github, Bitbucket, Jira and Jenkins provide different JSON structures. Therefore, supporting dynamic usage of common operators across various repositories in our query framework is unfeasible. To overcome that barrier, separate FOR EACH operators are used to extract data from respective data repositories.

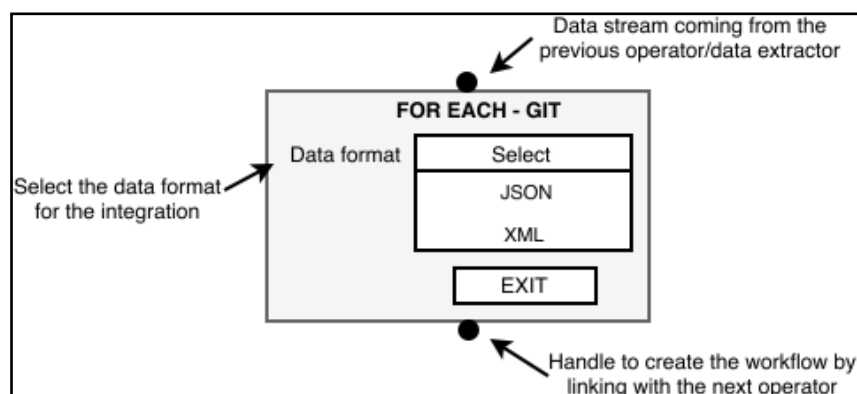


Figure 2 : FOR EACH Operator

**FILTER BY operators:** To overcome the same issue encountered by FOR EACH operators, separate operators are used to filter the data. As of now, our query interface facilitates filtering by date range and conditions. For example, FILTER BY can be used to filter the Github commits of a particular project that are recorded within a given date range. Moreover, FILTER BY assists conditional filtering such as filter by name or email in version control repositories, filter by result or timestamp in build repositories and filter by status or assignee in bug repositories.

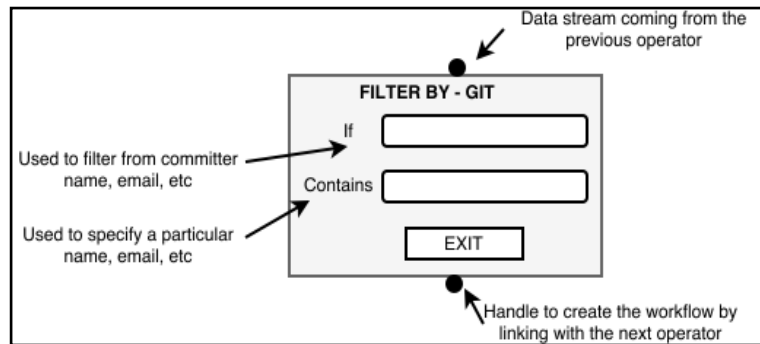


Figure 1 : FILTER BY Operator

**COUNT operators:** These operators count the number of commits in a version control repository or number of bugs in a bug repository based on the filtering operations. COUNT operators are easy to use and no parameters are required for execution.

**PERCENTAGE operators:** They count the percentage number of resulted commits or the percentage number of resulted bugs based on the filtering operations performed.

**SELECT operator:** This operator is available only for build repositories to select the build element that is to be in the result set. The logic of this operator is identical to the 'select' operation in any standard query language such as SQL.

**METRICS operator:** This operator acts as a metrics calculator, which facilitates calculating code specific metrics (i.e., size, complexity) as well as project specific metrics (i.e., test coverage, security, duplications).

**OUTPUT operator:** Repository specific output operators are used if the query interacts with a single repository to answer the question. When multiple repositories are integrated, a generic common output operator is used.