*Group 9 - Iota*
*Sprint 3 and 4 Progress Document*
*CS 4320*
*7 December 2022; 19 December 2022*

# Information

---

## Instructions

To use an endpoint, activate the iota augur server under the dev-group9 branch, and follow the link to the corresponding endpoint (located beneath the implemented queries).

## Challenges Faced

While the majority of recent work was directed towards completing and refining the SQL queries to be used by the endpoints, the primary challenge faced was being able to determine whether the implemented endpoints were functioning properly. Initially, pulling changes down to the group server and restarting augur was difficult; this was resolved by pulling in several changes from the upstream repository and determining the proper way to restart augur once the endpoints were added. Another difficulty experienced by the group was related to the fact that non-functional endpoints were troublesome to debug due to lacking specific information on what had gone wrong. Fortunately, we were able to solve this issue by observing logs and other already implemented endpoints. Along with these, implementing tests for the new endpoints proved to be another obstacle. However, this was largely mitigated by referencing other tests and ensuring that the new tests were working with data that is known to be populated in the database.

*Group 9 - Iota*
*Sprint 3 and 4 Progress Document*
*CS 4320*
*7 December 2022; 19 December 2022*

# SQL Queries to be Used in Endpoints

---

- Proportion of merged/closed pull requests
    - This query returns a table of repos and the corresponding merged/closed pull request ratio for each repo. If there are no closed pull requests for the repo, then the ratio is null (division by 0)

```
SELECT
	–a.repo_id,
	CAST(merged AS FLOAT) / NULLIF(CAST(closed AS FLOAT),0) AS
merged_to_closed_ratio FROM
	(SELECT
		repo_id, COUNT(pr_merged_at) AS merged
		FROM pull_requests
		WHERE pr_merged_at IS NOT NULL
		GROUP BY repo_id) a
	INNER JOIN
	(SELECT
		repo_id, COUNT(pr_closed_at) AS closed
		FROM pull_requests
		WHERE pr_merged_at IS NULL
		GROUP BY repo_id) b
	ON a.repo_id = b.repo_id
	WHERE a.repo_id = :repo_id;
```

This endpoint has been successfully implemented.
http://iota.osshealth.io:5076/api/unstable/repo-groups/1/pull-request-ratio-merged-to-closed

---

*Group 9 - Iota*
*Sprint 3 and 4 Progress Document*
*CS 4320*
*7 December 2022; 19 December 2022*

- Sum of additions/removals per contributor per repository.
    - This query will return a table of contributor email addresses mapped to the net contributions by that contributor. The table will be empty if there are no common contributors between the contributors and commits tables.

```
SELECT
        cntrb_email,SUM(cmt_added)-SUM(cmt_removed) AS net_contributions FROM
        contributors, commits
                WHERE cntrb_email=cmt_committer_email
                GROUP BY cntrb_email;
```

This endpoint has been successfully implemented.
http://iota.osshealth.io:5076/api/unstable/iota/get-net-contributions

---

- Release/tag frequency
    - This query will return a ratio of published releases over time from a given repo.
    - It is specifically intended to be used when the period is “days”. However, the current implementation in augur accounts for this by manipulating the date appropriately.
    - Returns a ratio of releases per day.

```
SELECT
        COUNT(release_published_at) / ((DATE_PART('day', AGE(:end_date,
'2022-10-10'))/365) +
        (DATE_PART('month', AGE(:end_date, :begin_date))/12) +
        DATE_PART('year', AGE(:end_date, :begin_date))) AS releases_over_time
FROM
        releases
WHERE release_published_at BETWEEN :begin_date AND :end_date
AND repo_id = :repo_id
GROUP BY repo_id;
```

This endpoint has been successfully implemented.
http://iota.osshealth.io:5076/api/unstable/repo-groups/1/release-frequency

---

*Group 9 - Iota*
*Sprint 3 and 4 Progress Document*
*CS 4320*
*7 December 2022; 19 December 2022*

- Sustained contributors
    - This will retrieve the name of the contributors who have committed to the repository more than once within a certain window of time into the past, as well as their number of commits. The table will be empty if there are no sustained contributors.

```
SELECT * FROM
        (SELECT
                cmt_author_email AS sustained_cntrb_email,
                COUNT(cmt_id) AS total_contributions
                FROM commits
                WHERE repo_id = :repo_id
                AND (cmt_author_email = '') IS NOT TRUE
                AND cmt_author_date IS NOT NULL
                AND cmt_author_date > :begin_date
                AND cmt_author_date < :end_date
                GROUP BY cmt_author_email) a
WHERE total_contributions > 1;
```

This endpoint has been successfully implemented.
http://iota.osshealth.io:5076/api/unstable/repo-groups/10/sustained-contributors

---

- Messages by users on issues/PRs
    - This query returns how many messages were sent in a given repo between a start and end date.

```
SELECT
        COUNT(msg_id) AS msg_count
        FROM message
        WHERE msg_timestamp BETWEEN :begin_date AND :end_date
        AND repo_id = :repo_id
        GROUP BY repo_id;
```

This endpoint has been successfully implemented.
http://iota.osshealth.io:5076/api/unstable/repos/1/messages