

**AURA**

# **Руководство пользователя**

# Оглавление

<b>Оглавление</b>	<b>2</b>
<b>Сбор данных</b>	<b>3</b>
Описание процесса сбора данных	3
Требования к рабочему месту пользователя	3
Инструкция по сбору данных	4
<b>Анализ данных</b>	<b>6</b>
Описание сводной диаграммы	6
<b>Приложение 1. Описание форматов файлов</b>	<b>9</b>
Список изменений	9
Связи изменений между собой	10
Статистика работы инструментария сбора данных	11
<b>Приложение 2. Техническая поддержка</b>	<b>12</b>

# Сбор данных

## Описание процесса сбора данных

Сбор данных из системы контроля версий Git осуществляется на стороне пользователя с использованием инструментария для сбора данных. Скрипты для автоматизации работы инструментарий для удобства использования и сопровождения размещены на GitHub, инструментарий упакован в контейнер Docker и опубликован на Docker Hub.

Выгружаемые инструментарием данные не содержат исходный код пользователя.

Собранные данные размещаются пользователем на облачном файловом хранилище<sup>1</sup> и ссылка на файлы отправляется чат-боту Aura в Telegram. Сервис Aura скачивает файлы по указанной ссылке из файлового хранилища и затем обрабатывает их, после чего пользователь получает возможность проанализировать обработанные данные.

## Требования к рабочему месту пользователя

- Операционная система Windows, Linux или MacOS;
- Docker;
- Telegram;
- Доступ в интернет<sup>2</sup>.

---

<sup>1</sup> На данный момент поддерживается [Yandex Disk](#) или любое другое файловое хранилище, позволяющее скачать файл по протоколу https.

<sup>2</sup> Для работы Telegram и для загрузки Docker-образа инструментария для сбора данных

## Инструкция по сбору данных

1. Если для чтения вашего git-репозитория требуется аутентификация, то необходимо получить токен доступа<sup>3</sup>
2. Получить адрес git-репозитория (repository url), который вы хотите проанализировать
3. Запустить сбор данных git-репозитория
  - a. Для Linux и MacOS

```
./collect <токен доступа> <repository url>
```
  - b. Для Windows

```
./collect.bat <токен доступа> <repository url>
```

### Пример (Linux / MacOS):

с авторизацией по токenu

```
./collect b_9FM2xTM5AE8Yz-6eYG https://git.company.com/project
```

без авторизации

```
./collect "" https://github.com/kubernetes/kubernetes
```

### Пример (Windows):

с авторизацией по токenu

```
./collect.bat b_9FM2xTM5AE8Yz-6eYG https://git.company.com/project
```

без авторизации

```
./collect.bat "" https://github.com/kubernetes/kubernetes
```

Сбор данных в зависимости от размера репозитория может занять от нескольких секунд (размер - мегабайты) до нескольких часов (размер - гигабайты).

Сбор данных состоит из следующих стадий:

1. Stage clone - клонирование репозитория на локальный диск;
2. Stage payload - сбор и первичная обработка данных;
3. Stage final - очистка временных данных и вывод статистики о работе инструментария сбора данных.

---

<sup>3</sup> Например, в GitLab зайдите в учетную запись, далее "Edit profile \ Access Tokens", создать токен, указав уровень доступа "read\_repository" или выше.

В результате работы инструментария в текущую папку будут записаны три файла:

<наименование репозитория>.changes.csv

<наименование репозитория>.commit.csv

<наименование репозитория>.stat.csv

Наименование типа файла	Описание
*.commit.csv	<a href="#">Список изменений</a>
*.changes.csv	<a href="#">Связи изменений между собой</a> Если в git-репозитории только одно изменение, то файл *.changes.csv не будет создан.
*.stat.csv	<a href="#">Статистика работы инструментария сбора данных</a>

Описание формата каждого из файлов приведено в [Приложении 1](#).

Все три файла необходимо загрузить на облачное файловое хранилище<sup>4</sup> и отправить в чат Telegram-боту [“Aura. Загрузка данных”](#) https-ссылки на загруженные файлы. Загрузка файлов осуществляется путем нажатия на кнопку, соответствующую типу загружаемого файла и отправки ссылки на файл **в ответ** на появившееся сообщение от бота с приглашением загрузки.

После получения ссылок Telegram-бот [“Aura. Загрузка данных”](#) загрузит файлы, обработает их содержимое и отправит в чат пользователю в ответном сообщении результат обработки. Если загрузка и обработка завершились успешно, то можно запросить данные для анализа в чате у Telegram-бота [“Aura. Диаграммы”](#) путем нажатия кнопки “Отобразить сводную диаграмму”.

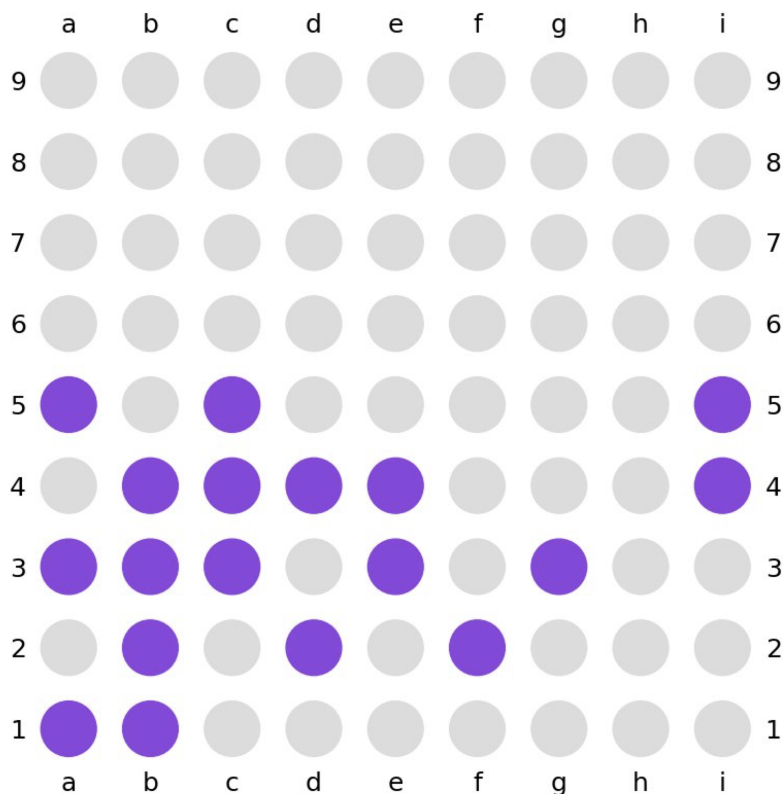
---

<sup>4</sup> На данный момент поддерживается [Yandex Disk](#) или любое другое файловое хранилище, позволяющее скачать файл по протоколу https.

## Анализ данных

Данные для анализа можно запросить в чате у Telegram-бота [“Aura. Диаграммы”](#) путем нажатия кнопки “Отобразить сводную диаграмму”.

### Описание сводной диаграммы



Сводная диаграмма представляет собой квадратное поле 9 x 9 точек. На сводной диаграмме выведены данные всех разработчиков проекта. Данные объединены<sup>5</sup> в точки. Если точка содержит данные разработчиков, то она отмечена фиолетовым цветом, если не содержит, то серым.

Горизонтальная ось отражает степень активности каждого разработчика в системе контроля версий, при этом, точки с индексом “a” имеют наименьшую степень активности, точки с индексом “i” - наибольшую.

Вертикальная ось отражает степень переработки исходного кода, то есть, степень исправления уже существующего кода. При этом, индекс “1”

<sup>5</sup> Кластеризованы

соответствует минимальному объему повторной работы, а “9” - максимальному. Данные по всем разработчикам усреднены за последние 30 дней.

Список разработчиков отображается в чате ниже сводной диаграммы в файле “legend.html”. Файл необходимо скачать, и открыть на просмотр.

Пример отображения фрагмента файла со списком разработчиков:

		Разработчики
a	1	cathryn_west13@hotmail.com
a	1	kurt.shields30@hotmail.com
a	3	ashtyn_kohler57@yahoo.com
a	5	samantha13@hotmail.com
a	5	carli.hamill4@hotmail.com
b	1	jessy_hegmann8@hotmail.com
b	2	nikko.gulgowski81@gmail.com
b	2	vaughn.sauer@hotmail.com
b	3	hector_becker@hotmail.com
b	3	lina.corwin@gmail.com
b	3	chelsey.gottlieb@yahoo.com
b	4	kamryn_kris@gmail.com
b	4	daphne37@gmail.com

В первых двух колонках выведен адрес точки на сводной диаграмме, в которую попал разработчик. В третьей колонке имя пользователя в системе контроля версий Git<sup>6</sup>.

Например, для того, чтобы определить по диаграмме, кто попал в точку с минимальной активностью по изменению исходного кода и максимальным уровнем переработки исходного кода, требуется найти точку с минимальной координатой по горизонтальной оси, и с максимальной по вертикальной, то есть а5. По таблице с легендой определяем, что в эту точку попали разработчики samantha13@hotmail.com и carli.hamill4@hotmail.com

С другой стороны, для того, чтобы определить степень активности и степень переработки исходного кода для заданного разработчика, например, для kurt.shields30@hotmail.com, по таблице определяем, что он попал в точку а1, что

<sup>6</sup> Зачастую, в качестве имени пользователя используется адрес электронной почты

соответствует минимальной активности и минимальной степени переработки исходного кода.

Высокая или низкая активность или степень переработки исходного кода сами по себе не являются положительными или отрицательными метриками работы разработчиков. Эти данные используются для поиска разработчиков, активность и степень переработки исходного кода которых аномальны по сравнению с остальными разработчиками. Это может быть маркером, как высокой эффективности, так и обратной ситуации в зависимости от контекста.



## Приложение 1. Описание форматов файлов

Файлы с собранными данными имеют формат CSV<sup>7</sup>, значения полей разделены “,”.

### Список изменений

Порядковый номер поля	Описание
1	Наименование репозитория
2	Уникальный идентификатор внесенного изменения <sup>8</sup>
3	Дата и время внесенного изменения в формате Unix time <sup>9</sup>
4	Количество файлов, затронутых изменением
5	Количество добавленных строк
6	Количество удаленных строк
7	Уникальные идентификаторы “родительских” изменений
8	Имя автора изменения, закодированное в Base64 <sup>10</sup>
9	Адрес электронной почты автора изменения, закодированный в Base64
10	Имя вносящего изменение, закодированное в Base64
11	Адрес электронной почты вносящего изменения, закодированный в Base64
12	Комментарий к изменению, закодированный в Base64

<sup>7</sup> Comma-Separated Values, <https://ru.wikipedia.org/wiki/CSV>

<sup>8</sup> Commit checksum - SHA-1 hash

<sup>9</sup> <https://ru.wikipedia.org/wiki/Unix-%D0%B2%D1%80%D0%B5%D0%BC%D1%8F>

<sup>10</sup> <https://ru.wikipedia.org/wiki/Base64>

# Связи изменений между собой

Порядковый номер поля	Описание
1	Наименование репозитория
2	Уникальный идентификатор изменения, в рамках которого были добавлены строки
3	Количество строк кода, которые были добавлены изменением с уникальным идентификатором из поля 2 и удалены изменением с уникальным идентификатором из поля 4
4	Уникальный идентификатор изменения, в рамках которого были удалены строки

# Статистика работы инструментария сбора данных

Порядковый номер поля	Описание
1	Наименование репозитория <sup>11</sup>
2	Время завершения работы инструментария сбора данных <sup>12</sup>
3	Количество файлов, затронутых изменениями в репозитории
4	Количество удаленных строк кода в репозитории
5	Количество добавленных строк кода в репозитории
6	Количество внесенных изменений <sup>13</sup> в репозитории
7	Количество слияний <sup>14</sup> веток исходного кода в репозитории
8	Время клонирования репозитория в секундах
9	Время обработки репозитория в секундах
10	Время очистки временных данных в секундах
11	Общее время работы инструментария сбора данных

---

<sup>11</sup> Git Repository  
<sup>12</sup> В формате [unix time](#)  
<sup>13</sup> Git Commits  
<sup>14</sup> Git Merge

## Приложение 2. Техническая поддержка

Техническая поддержка по использованию сервиса Auga, в том числе по установке, конфигурированию и использованию инструментария для сбора данных, осуществляется по электронной почте [info@aura.expert](mailto:info@aura.expert).