Research source manager

Digital Excellence

Jurjen Verbruggen (GitHub)

Ramon Peek (GitLab)

Inhoud

[Research questions 3](#_Toc34747375)

[Research methods (GitHub) 4](#_Toc34747376)

[**Library** 4](#_Toc34747377)

[Method 4](#_Toc34747378)

[strategy 4](#_Toc34747379)

[outcomes 4](#_Toc34747380)

[**Workshop** 7](#_Toc34747381)

[Method 7](#_Toc34747382)

[strategy 7](#_Toc34747383)

[outcomes 7](#_Toc34747384)

[Research methods (GitLab) 8](#_Toc34747385)

[**Field** 8](#_Toc34747386)

[Method 8](#_Toc34747387)

[STRATEGY 8](#_Toc34747388)

[Outcomes 8](#_Toc34747389)

[**Library** 8](#_Toc34747390)

[Method 8](#_Toc34747391)

[strategy 8](#_Toc34747392)

[outcomes 8](#_Toc34747393)

[**Workshop** 10](#_Toc34747394)

[Method 10](#_Toc34747395)

[strategy 10](#_Toc34747396)

[outcomes 10](#_Toc34747397)

# Research questions

Before we start our research, we have set several questions we would like to be answered. These questions are as follows:

* Which data is available?
* Can you search through data within a subdirectory?
* Does the API have any limits set?
* How do we make the API calls?
* What do these APIs return?
* What do these APIs require to access their data?
* What is the response-time of the APIs?

These questions do both apply to GitLab and GitHub separately. We will define the research methods individually for these sources.

# Research methods (GitHub)

## **Library**

### Method

For this research we will be doing a **Literature Study**. This will be the most convenient, since this platform is being used world-wide, and thus they provide much information about GitHub online.

### strategy

During this research we will be asking ourselves the following questions:

* Which data is available?
* Can you search through data within a subdirectory?
* Does the API have any limits set?
* How do we make the API calls?
* What do these APIs return?
* What do these APIs require to access their data?

I expect to find answers in the GitHub documentation, on the GitHub questionnaire, or on third party developer forums, such as Stack Overflow.

GitHub documentation: <https://developer.github.com/v3/>

### outcomes

**Which data is available?**

It turns out the GitHub REST API is very detailed and there is a lot of data available to gather. Some data is restricted to what the authenticated user can see, for obvious reasons. You should be only able to gather data you have access to.

**Can you search through data within a subdirectory?**

To our knowledge, FHICT doesn’t have a subdirectory on GitHub. Projects of FHICT members are spread out over the whole database, and there will be no way of sorting them, but adding them individually. We will have to save GitHub projects links in our own database, referring to the projects hosted on GitHub itself. We will be able to search through this database, but it will be functioning separately from GitHub.

**How do we make the API calls?**

Before making any calls to the API, we want to make sure that we’re authenticated. This will enable us to make more API calls per hour.

Authentication can currently be done either two ways, through Basic Authentication or through OAuth 2. We will be using OAuth2, since it is better to store a access token in your application, than user credentials; we would not have to store any user passwords.

A problem arises when we want to start using OAuth2 in the way they say on their website; it uses curl. Because curl is a command line program, we cannot use it. We will need a library to do the job for us. This raises another question;

**Is there a library to make authenticated API calls to GitHub for .NET?**

It turns out there is, and it is called Octokit. Plenty of information can be found on their GitHub page, such as how to get started and a reference to their documentation.

**Does the API have any limits set?**

GitHub has a rate limiter installed for API requests. Authenticated requests can be sent 5000 times per hour at maximum, while unauthenticated requests are only allowed up to 60 times per hour. When the rate limit has been reached, a 403 HTTP error will be returned containing a message stating the rate limit has been reached.

GitHub’s search API has a custom rate limit. Authenticated requests can be sent up to 30 times per minute, while unauthenticated requests can only be sent 10 times per minute.

**What do these APIs return?**

Since we will be using the .NET client library specifically, we will not have to worry handling HTTP requests and the JSON data that is returned. All data will be returned in data structures defined in our library.

**What do these APIs require to access their data?**

We have mostly covered this question within the question about how we make the API calls. We can make requests through a HTTP call, but we will be using a .NET library, which will take care of this for us.

What we have not talked about is the authentication process, which will be essential to make authenticated requests. We will authenticate users following the OAuth flow.

A full reference to this flow can be found in the references at the end of this chapter.

**References**

* REST API GitHub - <https://developer.github.com/v3>
* Request rate limit GitHub - <https://developer.github.com/v3/#rate-limiting>
* Custom search request rate limit GitHub - <https://developer.github.com/v3/search/#rate-limit>
* Octokit, GitHub API client library for .NET - <https://github.com/octokit/octokit.net>
* Octokit documentation - <http://octokitnet.readthedocs.io/en/latest/>
* OAuth flow Octokit - <https://github.com/octokit/octokit.net/blob/master/docs/oauth-flow.md>

# Research methods (GitLab)

## **Field**

### Method

Fontys currently has its own GitLab servers, which makes it very approachable. Because of this ease of approach, we will **interview** people who have worked with the Fontys GitLab before, and students who have worked with the GitLab API before.

### STRATEGY

* Does the API have any limits set?
* How do we make the API calls?

### Outcomes

**Does the API (FHICT specific) have any (extra) limits set?**

I have sent an email to Marcel Narings with some questions regarding GitLab. I have asked whether there were any specific limitations for the FHICT GitLab, or if they were the same as the default for GitLab. Marcel replied saying that all limitations for the FHICT GitLab are the same as for the default GitLab.

I also asked in what way GitLab was hosted for FHICT (locally or externally). Marcel replied that everything is hosted locally within Fontys.

**How do we make the API calls?**

Assuming we will be using the GitLabAPIClient, all it requires is a host URL (<http://git.fhict.nl>), and either a token, or credentials (username and password). With this data, the library can use the full capacity of the GitLab endpoints.

**References**

Marcel Narings

<https://github.com/nmklotas/GitLabApiClient>

## **Library**

### Method

In order to find answers to the more technical questions, we will perform a **literature study** in the form of researching the GitLab documentation and the web. Where possible, we will use the official documentation and to answer any unanswered questions we will search the web for answers.

### strategy

* What data is available?
* Can you search through data within a subdirectory?
* Does the API (GitLab default) have any limits set?
* How do we make the API calls?
* What do these APIs return?
* What do these APIs require to access their data?

### outcomes

**What data is available?**

The GitLab API ([see endpoints documentation](https://docs.gitlab.com/ee/api/api_resources.html)) provides a lot of data of the projects and users, way more than we probably need to use in this project. Data that stood out, and which could be useful for the project were:

It is possible to get all members of a project, search for projects (when using the project-scope) and access project wikis for (for example) documentation. It is also possible to access project issues, which could for example contribute to the decision-making of the continuation of projects for our end-users. Snippets and todos are also available.

**Can you search through data within a subdirectory?**

Within GitLab it does not seem possible to specify a certain subdirectory to search in. As the Fontys GitLab only has access to the projects created on there, this would not be too big of a problem as these would need to be auto-imported anyways. The GitLab API does require to define a scope per search request, in which the user is able to specify the type of result is needed.

Available scopes:

* Projects
* Issues
* Merge requests
* Milestones
* Snippet titles
* Snippet blobs
* Wiki blobs
* Commits
* Blobs (Project files for example)
* Users

**Does the API (GitLab default) have any limits set?**

For performance reasons, if the number of resources is more than 10,000, the X-Total and X-Total-Pages headers as well as the rel="last" Link are not present in the response headers. GitLab.com also responds with HTTP status code 429 to API requests that exceed 10 requests per second per IP address.

**How do we make the API calls?**

There are several GitLab API clients available online, of which [GitLabAPIClient](https://github.com/nmklotas/GitLabApiClient) seems to be the best in our case. GitLabAPIClient is a fully async .NET-library capable of multi core paging. It is simple to use and handles URL encoding for us.

**What do these APIs return?**

The GitLab API returns the data in JSON format. For the sake of ease, we will provide an example of the structure of example json-data returned per scope:

* [Projects](https://pastebin.com/sffraqzx)
* [Issues](https://pastebin.com/c4i6bTqT)
* [Merge requests](https://pastebin.com/zj2ktR0c)
* [Milestones](https://pastebin.com/EupDnGFt)
* [Snippet titles](https://pastebin.com/QPUUVeS9)
* [Snippet blobs](https://pastebin.com/dT6LdnE8)
* [Wiki blobs](https://pastebin.com/mudpiJF5)
* [Commits](yrYKWfM3)
* [Blobs](https://pastebin.com/APDJm2fY) (Project files for example)
* [Users](https://pastebin.com/7LT5WMj2)

**What do these APIs require to access their data?**

Most API requests require authentication or will only return public data when authentication is not provided. For those cases where it is not required, this will be mentioned in the documentation for each individual endpoint. There are four ways to authenticate with the GitLab API:

* [OAuth2 tokens](OAuth2%20tokens)
* Personal access tokens -> This one will be used if we decide to log users in using SURF.
* [Session cookies](https://docs.gitlab.com/ee/api/#session-cookie)
* [GitLab CI job tokens](https://docs.gitlab.com/ee/api/#gitlab-ci-job-token)

**References**

<https://docs.gitlab.com/ee/api/api_resources.html>

<https://docs.gitlab.com/ee/api/search.html>

<https://docs.gitlab.com/ee/api/issues.html>

<https://docs.gitlab.com/ee/user/gitlab_com/index.html#gitlabcom-specific-rate-limits>

<https://docs.gitlab.com/ee/security/rate_limits.html>

<https://github.com/nmklotas/GitLabApiClient>

<https://docs.gitlab.com/ee/api/>

## **Workshop**

### Method

This stage of the research will consist mainly of **proof of concepts**. We will create small projects which will be used to test the previously determined results.

### strategy

* What do these APIs return?
* What is the response-time of the APIs?

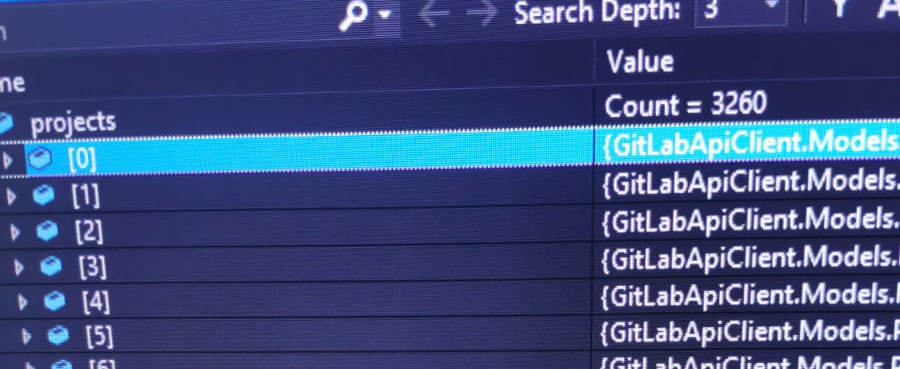
### outcomes

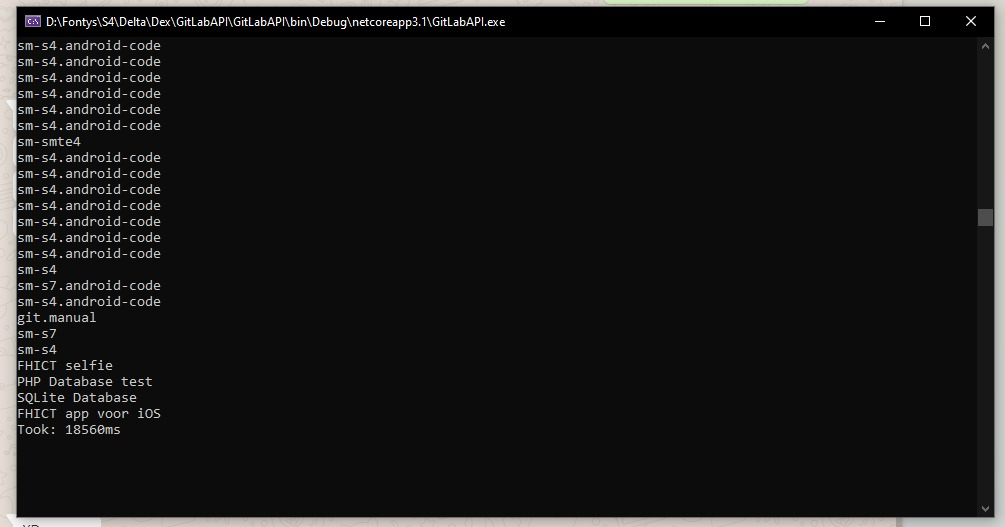
**What do these APIs return?**

The GitLabAPIClient returns full models which contain all the data which is presented by the API endpoints. This means all the data of projects etc. Is available in predefined models.

**What is the response-time of the API?**

The only situation I was able to test, was fetching all projects from the FHICT GitLab. This operation took approximately 18 seconds (fetching 3200 projects).





**References**

<https://github.com/nmklotas/GitLabApiClient>