TOSHIBA

Search GitLab, Redmine, and repositories with a single query

Deploy an enterprise search server with Fess

FOSDEM 2023

Takashi Kumagai, TOSHIBA Corporate Software Engineering Center
February 4, 2023

Today’s talk

- Our engineers use multiple content management tools
- **Searching** became a problem
    - Can’t search laterally across multiple tools
    - Can’t search texts inside binary files
- How we use Fess to solve the problem of search
- How we overcame (some of the) shortcomings of Fess
- Contributions (patches)
- Conclusion

Contents

01 Introduction

02 Background

03 Fess and enterprise search

04 Customization (patching Fess)

05 Automating configurations

06 Conclusion

Contents

01 Introduction

02 Background

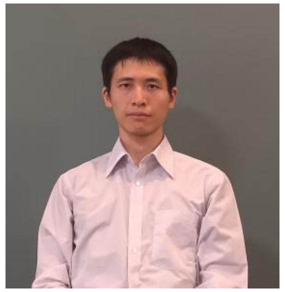
03 Fess and enterprise search

04 Customization (patching Fess)

05 Automating configurations

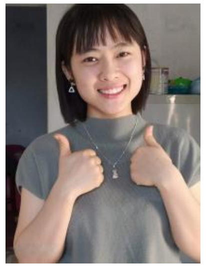
06 Conclusion

Who am I

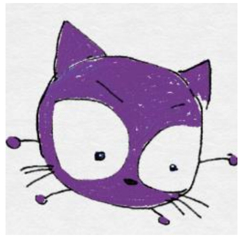


- An engineer at Corporate Software Engineering Center in Toshiba
- 2006 – 2015
  - Develop applications for Toshiba’s products (HDTV, etc.)
- 2015 – 2022
  - Maintain and improve the company’s cloud infrastructure and automation processes.
    - Build in-house tools
    - Write automation scripts
    - Automate E2E tests
    - Configure and deploy search servers to departments in other Toshiba companies

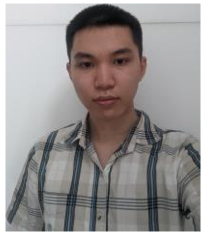
Development team engineers



Vũ Thị Thanh Thanh



Đỗ Thành Trung



Hoàng Trung Hiếu

Lê Tiến

Contents

01 Introduction

02 Background

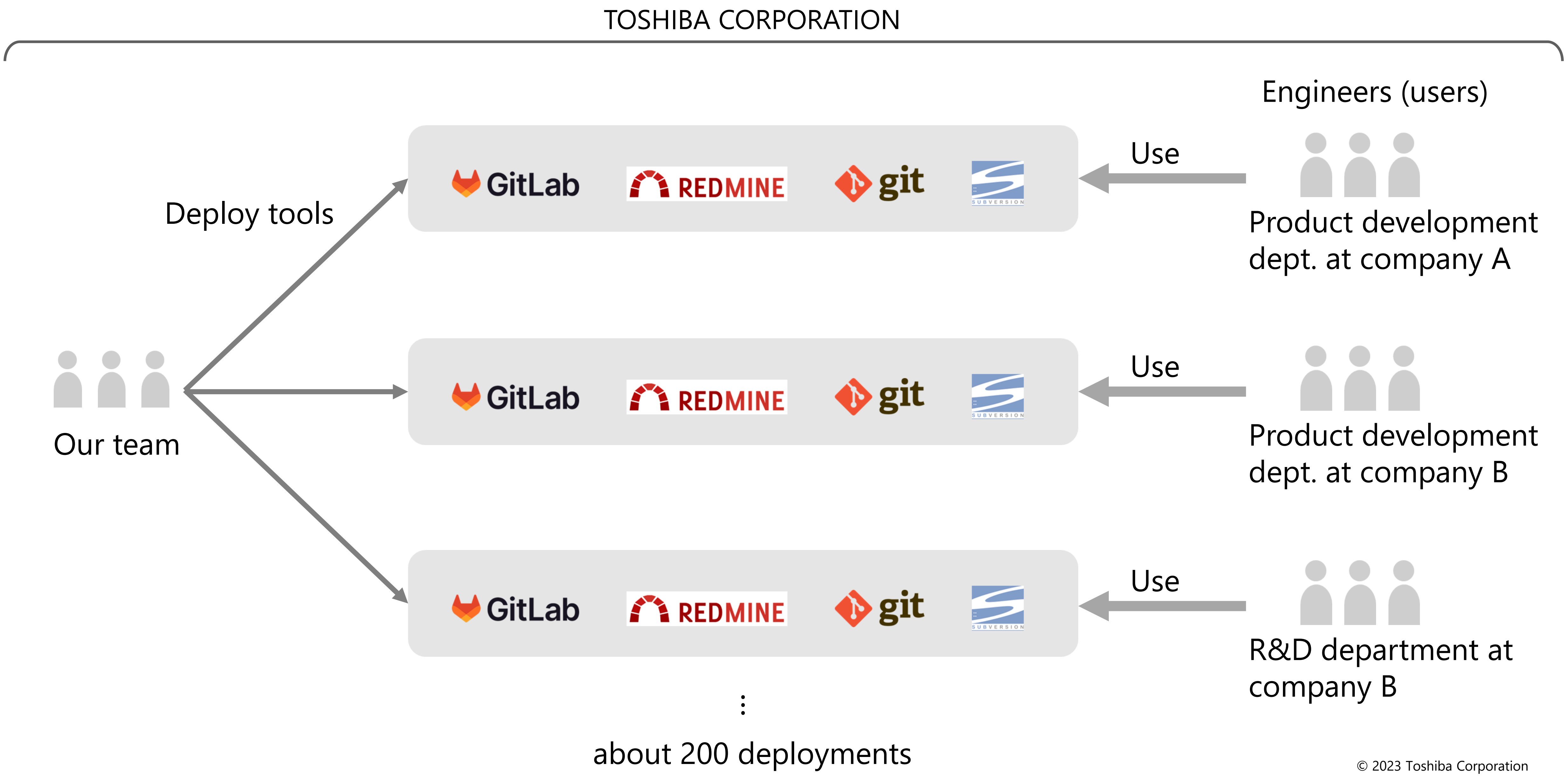
03 Fess and enterprise search

04 Customization (patching Fess)

05 Automating configurations

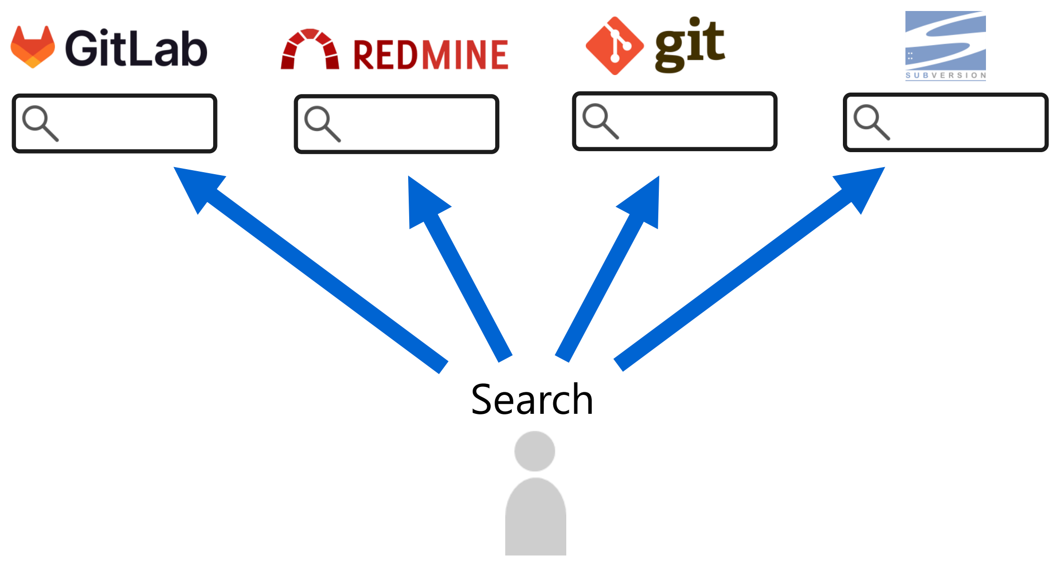
06 Conclusion

Providing software development tools



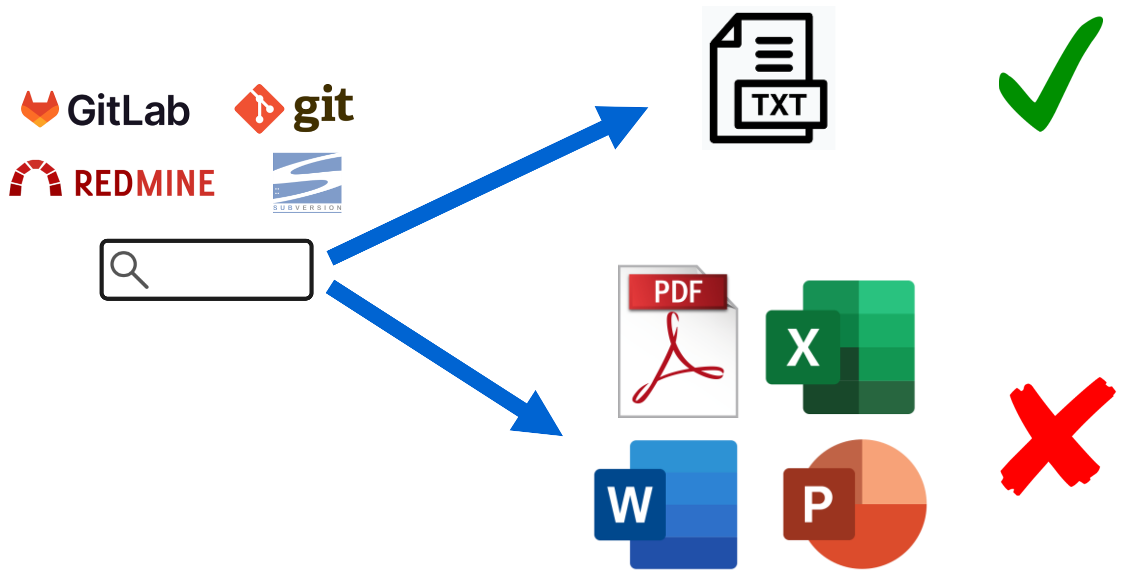
2 major issues resulting from using multiple tools

Can't search across tools



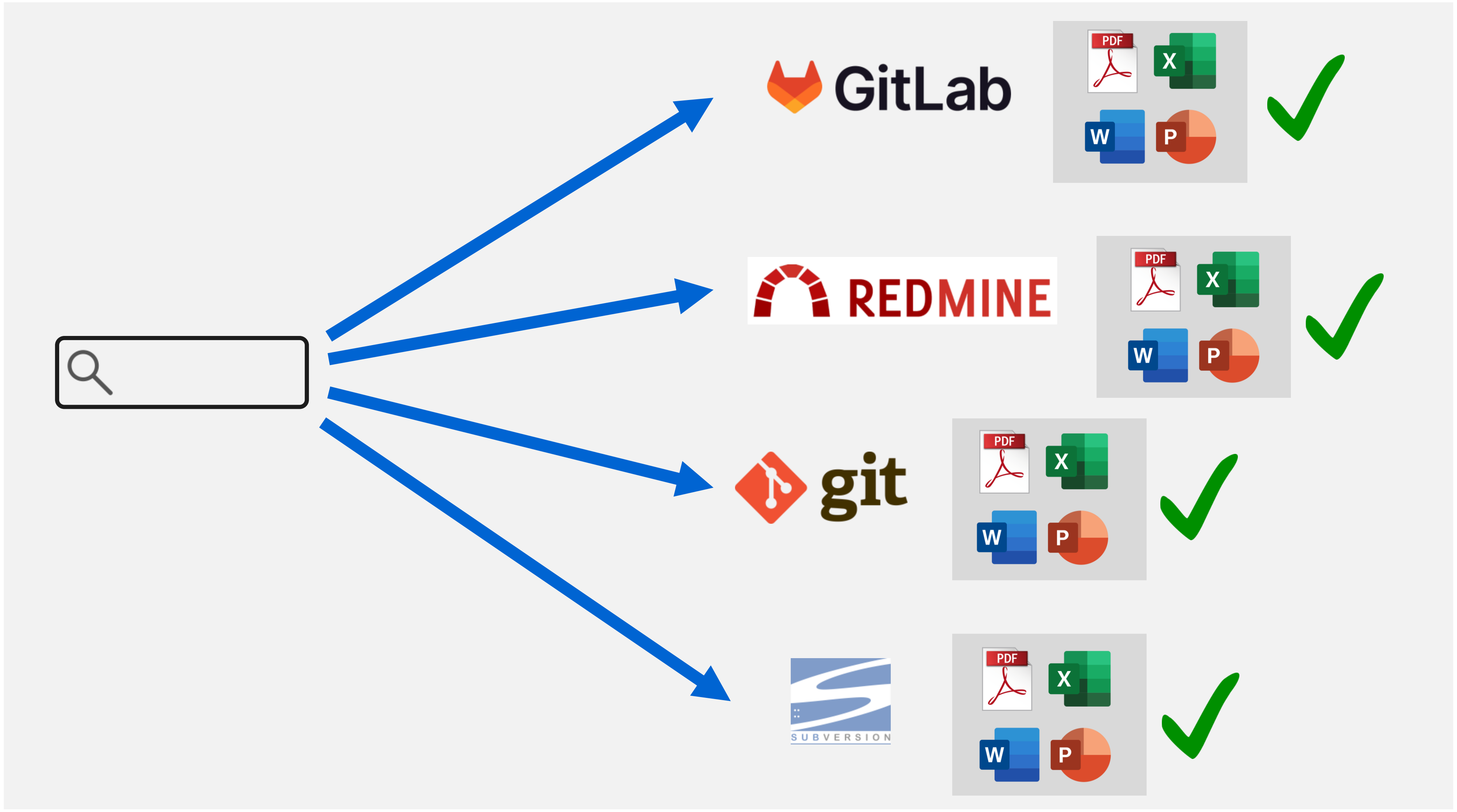
Resources, e.g. source code, documents, and wiki, are stored in multiple content management tools (fragmentation).

Can't search texts inside binary files



Goal

One search box to rule them all



Approach

- Making a search engine by ourselves would be too expensive.
- Better to find an **OSS search server** and configure/customize it for our needs.

- Requirements:
  - Shall solve the aforementioned 2 problems.
  - Shall be complete with the essential features we need, e.g.
    - Search box UI for users
    - Features for admins (easily set up and run web crawlers, etc.).
  - *(A few more requirements; more on them later)*

- An OSS search server named **Fess** stood out as the closest to what we needed.

Contents

01 Introduction

02 Some background

03 Fess and enterprise search

04 Customization (patching Fess)

05 Automating configurations

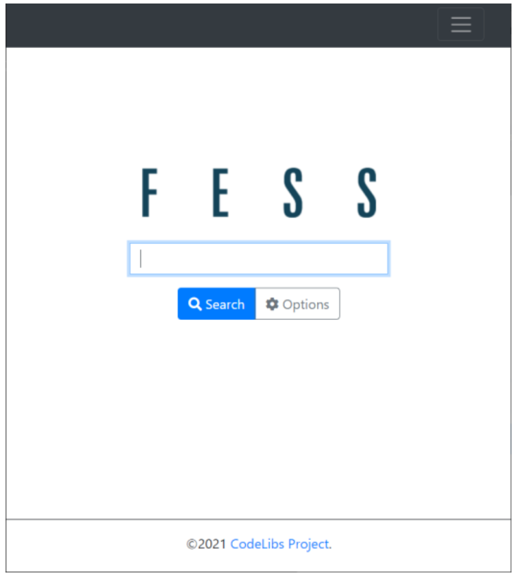
06 Conclusion

Fess

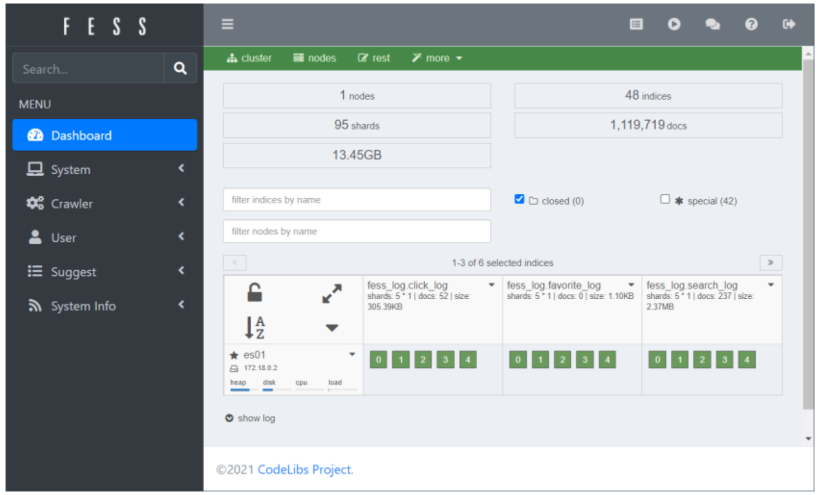
*software for searching for information within an enterprise

- *An enterprise search* server*
- Developed by a company named CodeLibs
- Uses Elasticsearch as the search engine
  - Supports indexing of certain binary files, e.g. Microsoft Office, PDF, and zip.
- Comes with several types of crawlers, which can crawl
  - documents on a web server
  - file system
  - data store (such as a CSV or database).

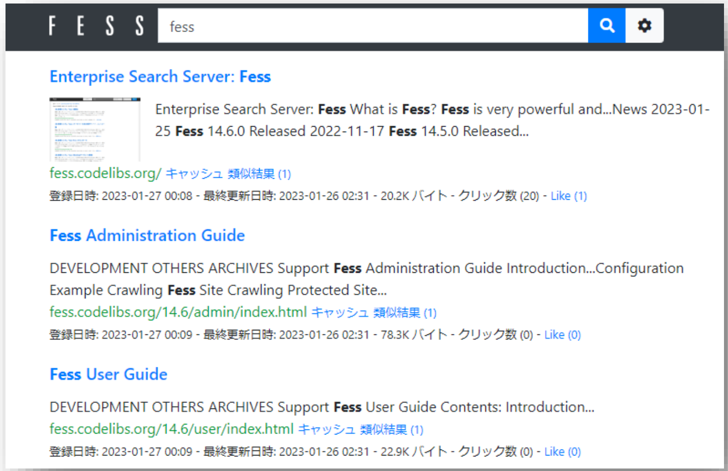
https://github.com/codelibs/fess



Search box GUI



Admin console



Enterprise Search Server: **Fess**

Enterprise Search Server: **Fess** What is **Fess**? **Fess** is very powerful and...News 2023-01-25 **Fess** 14.6.0 Released 2022-11-17 **Fess** 14.5.0 Released...

fess.codelibs.org/ キャッシュ 類似結果 (1)

登録日時: 2023-01-27 00:08 - 最終更新日時: 2023-01-26 02:31 - 20.2K バイト - クリック数 (20) - Like (1)

Fess Administration Guide

DEVELOPMENT OTHERS ARCHIVES Support **Fess** Administration Guide Introduction...Configuration Example Crawling **Fess** Site Crawling Protected Site...

fess.codelibs.org/14.6/admin/index.html キャッシュ 類似結果 (1)

登録日時: 2023-01-27 00:09 - 最終更新日時: 2023-01-26 02:31 - 78.3K バイト - クリック数 (0) - Like (0)

Fess User Guide

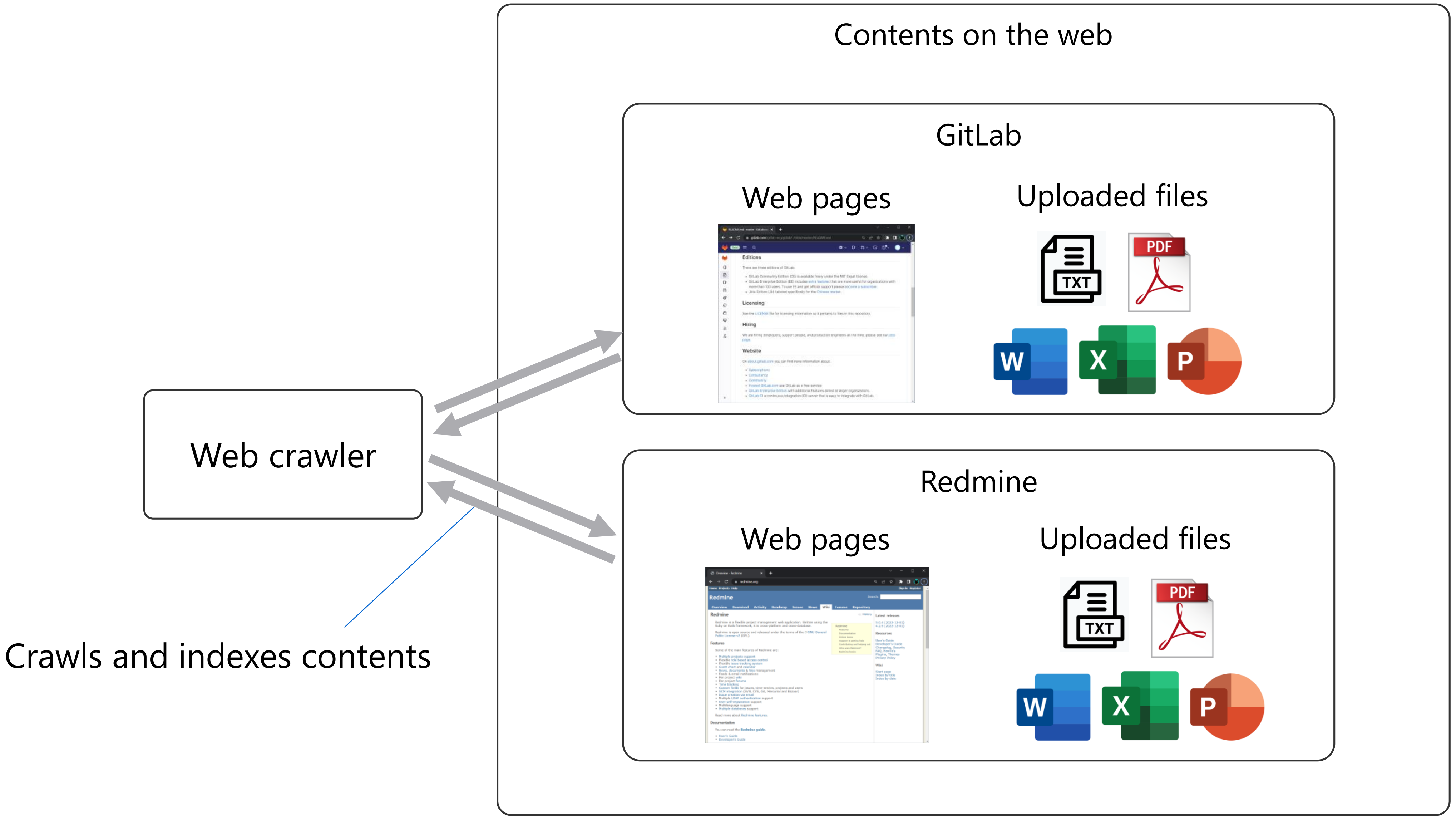
DEVELOPMENT OTHERS ARCHIVES Support **Fess** User Guide Contents: Introduction...

fess.codelibs.org/14.6/user/index.html キャッシュ 類似結果 (1)

登録日時: 2023-01-27 00:09 - 最終更新日時: 2023-01-26 02:31 - 22.9K バイト - クリック数 (0) - Like (0)

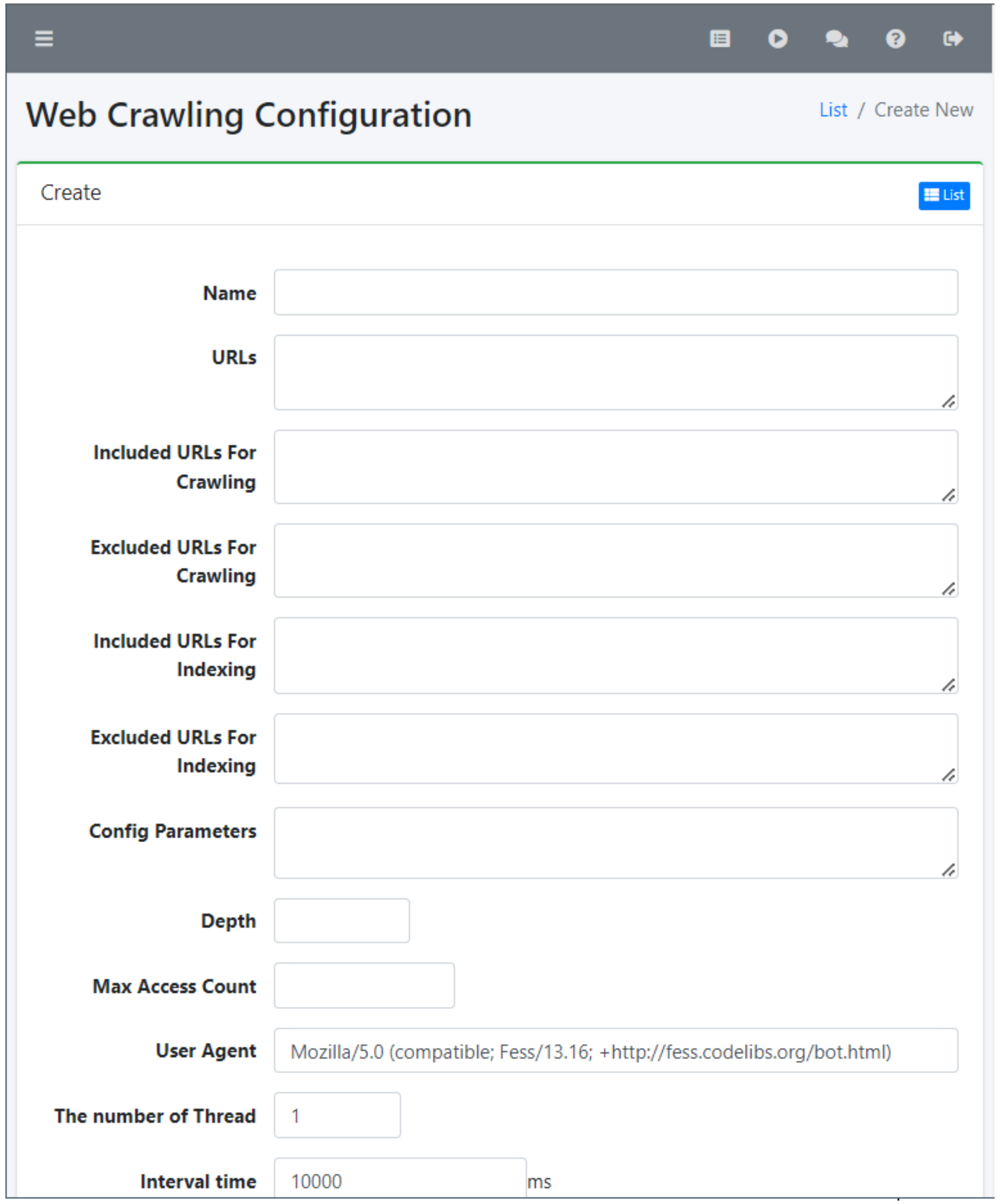
Search engine results page (SERP)

Fess web crawler



Web crawler configuration basics (1 of 3)

- Crawls and indexes web page contents (texts, uploaded files, etc.) by following links recursively.
- Fess provides an admin console GUI to create a crawler.



Name	
URLs	
Included URLs For Crawling	
Excluded URLs For Crawling	
Included URLs For Indexing	
Excluded URLs For Indexing	
Config Parameters	
Depth	
Max Access Count	
User Agent	Mozilla/5.0 (compatible; Fess/13.16; +http://fess.codelibs.org/bot.html)
The number of Thread	1
Interval time	10000 ms

Web crawler configuration basics (2 of 3)

Web crawler parameters

Parameter name	Description	Example
URLs	Starting point of the web crawling/indexing	https://mygitlab.io
Include URLs (regex)	Web crawler will crawl the page only if the URL matches any of the listed regex.	^https://mygitlab\.io/myproject/-/issues/\d+$ ^https://mygitlab\.io/myproject/-/wikis(?!.*/edit$).*$
Exclude URLs (regex)	Prevent crawlers from indexing irrelevant pages.	^https://mygitlab\.io/help($\|[/?].*$) ^https://mygitlab\.io/(?!.*/-/.*/-/).+/-/new/.*$
Depth	Maximum recursion depth	1000
Max Access Count	The maximum number of pages to crawl.	10000000
Permissions	Determines who can find the indexed contents	{group}mygroup {user}user1 {user}user2 {user}user3

Web crawler configuration basics (3 of 3)

Permission settings and access control

Parameter name	Description	Example
Permissions	Determines who can find the indexed contents	{group}mygroup {user}user1 {user}user2 {user}user3

- Allows admin to implement per-user access control
- Permission settings are **per-web crawler**
  - e.g. 100 projects on GitLab → create 100 web crawlers, each of which has its own permission settings
- **username** in {user}**username** can be either
  - Users created on Fess
  - Users authenticated via LDAP directory service

Contents

01 Introduction

02 Some background

03 Fess and enterprise search

04 Customization (patching Fess)

05 Automating configurations

06 Conclusion

Customization (patching Fess)

- Patches delineated in this segment:
    - Authentication for web crawlers
    - Crawling and indexing repository contents

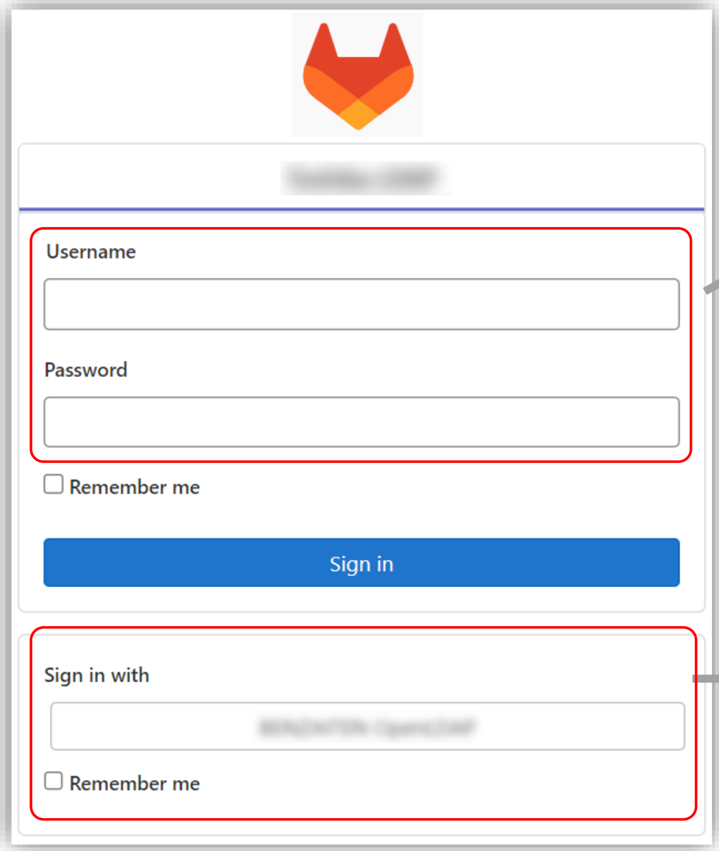
• Patches merged upstream
• Patch submitted but not merged
• Patches kept proprietary

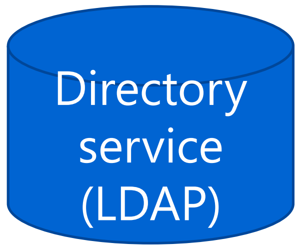
	Patch	Link* / patched file(s)
1	Disable default PostConstruct	codelibs/fess-crawler/pull/56
2	Make webdriver quit after it finishes crawling	codelibs/fess-crawler/pull/58
3	(webdriver) add option for specifying chrome arguments	codelibs/fess-crawler/pull/57
4	(webdriver) add option for chrome.prefs	codelibs/fess-crawler/pull/57
5	(webdriver) add authentication	-
6	(webdriver) support crawl files	-
7	(webdriver) add basic auth	-
8	Add saml auth	-
9	map filesystem paths to URLs	-
10	(webdriver) customize last modified	-
11	(webdriver) prioritize fess cookies	-
12	(webdriver) add waiting configurations	-

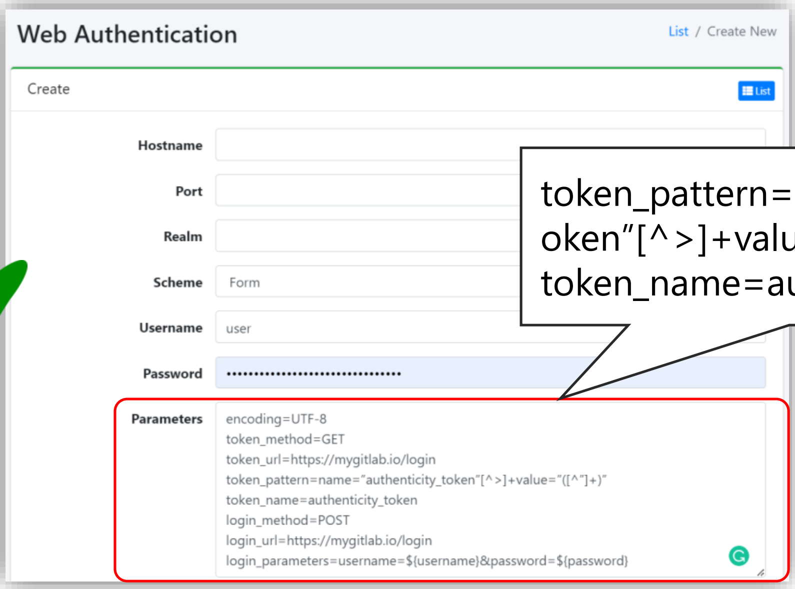
*paths of Link URLs following https://github.com/

Authentication for web crawlers

- Web crawlers need authentication.
- Fess's **web authentication** supports certain types of login pages but not all.






Web Authentication

List / Create New

Create

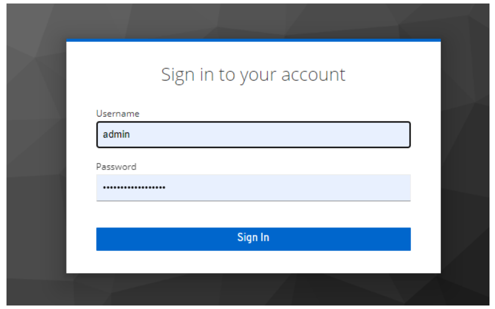
Hostname	
Port	
Realm	
Scheme	Form
Username	user
Password	••••••••••••••••••••••••••••••••

Parameters

```
encoding=UTF-8
token_method=GET
token_url=https://mygitlab.io/login
token_pattern=name="authenticity_token"[^>]+value="([^"]+)"
token_name=authenticity_token
login_method=POST
login_url=https://mygitlab.io/login
login_parameters=username=${username}&password=${password}
```

```
token_pattern=name="authenticity_token"[^>]+value="([^"]+)"
token_name=authenticity_token
```

Keycloak/SAML



Not supported

- **We wrote a patch to authenticate the web crawler through our SAML-enabled GitLab sign-in page**
  - **Patched webdriver client and http form**

Patching webdriver client and HTTP form

New optional parameters on Fess admin console

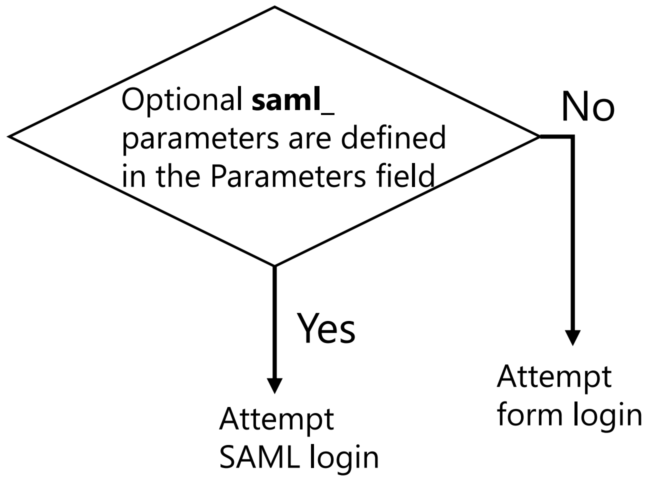
Fess console:
(web authentication)

Scheme	Form
Username	user
Password	******

Parameters	encoding=UTF-8
	token_method=GET
	token_url=https://tccloud2.toshiba.co.jp/ditc/gitlab/users/sign_in
	token_pattern=name="authenticity_token" +value="([^\"]+)"
	token_name=authenticity_token
	login_method=POST
	login_parameters=username=${username}&password=${password}
	saml_url=https://mygitlab.io/users/auth/saml
	saml_login_url_pattern=action="([^"]+)"
	saml_response_pattern=name="SAMLResponse" +value="([^"]+)"
	saml_response_name=SAMLResponse
	saml_callback_url=https://mygitlab.io/users/auth/saml/callback

The patched form parser checks for optional **saml_** parameters and stores them if they are present

The patched web driver client attempts SAML login if optional parameters are defined

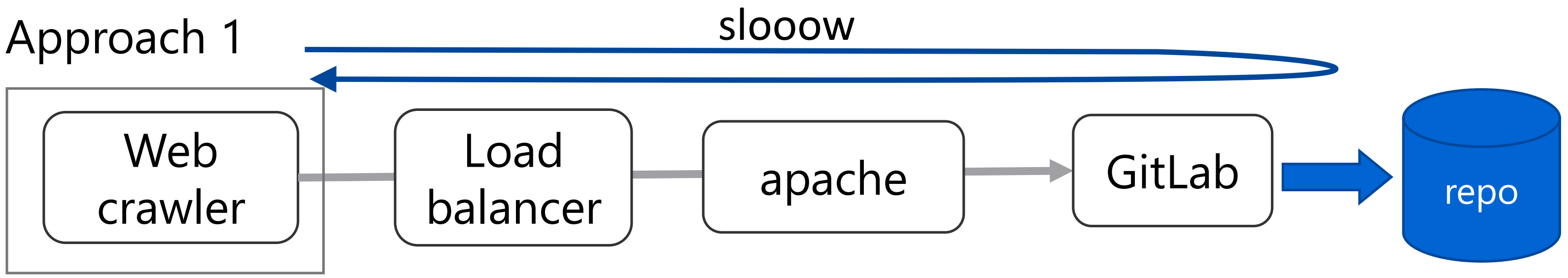

Optional **saml_** parameters are defined in the Parameters field
No
Yes
Attempt SAML login
Attempt form login

Crawler supports SAML login

Crawling and indexing repository contents

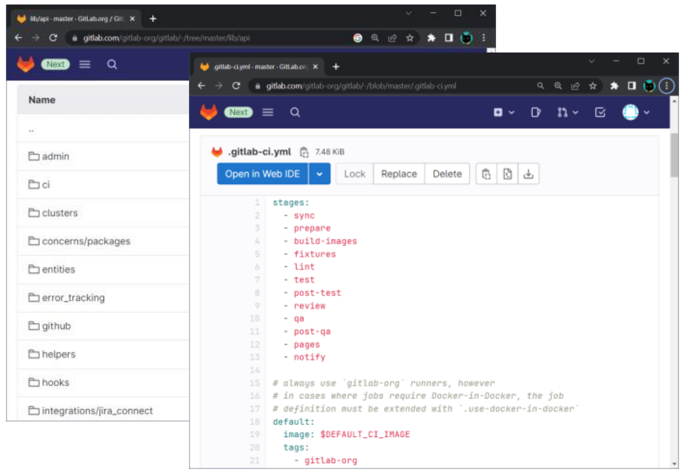
- Many of our repositories are several GBs in size.
- Indexing the repo contents using web crawlers turned out to be too slow.

Approach 1



Web crawlers can index files in repo but it takes much longer
- HTTP request to GitLab → GitLab fetches repo file contents → GitLab renders the file contents on the page*

*Repo files/folders as seen on the browser (GitLab)

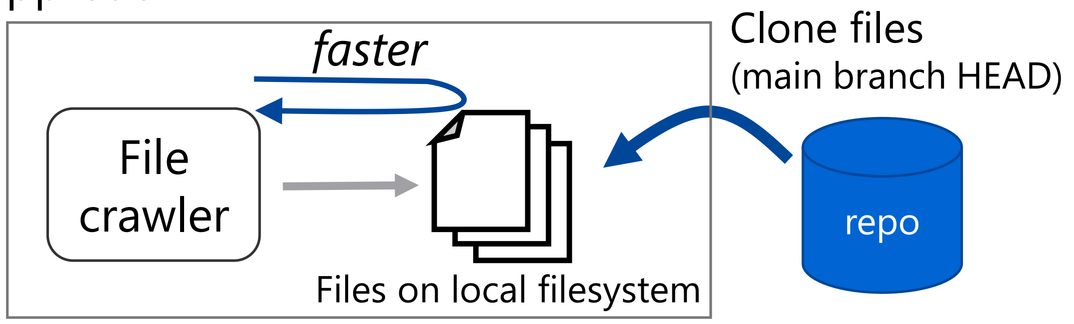

lib/api · master · GitLab.org / Git
gitlab.com/gitlab-org/gitlab/-/tree/master/lib/api
Next
.gitlab-ci.yml · master · GitLab.org
gitlab.com/gitlab-org/gitlab/-/blob/master/.gitlab-ci.yml
Open in Web IDE
Lock
Replace
Delete

Name
..
admin
ci
clusters
concerns/packages
entities
error_tracking
github
helpers
hooks
integrations/jira_connect

.gitlab-ci.yml 7.48 KiB

```
stages:
  - sync
  - prepare
  - build-images
  - fixtures
  - lint
  - test
  - post-test
  - review
  - qa
  - post-qa
  - pages
  - notify

# always use `gitlab-org` runners, however
# in cases where jobs require Docker-in-Docker, the job
# definition must be extended with `.use-docker-in-docker`
default:
  image: $DEFAULT_CI_IMAGE
  tags:
    - gitlab-org
```

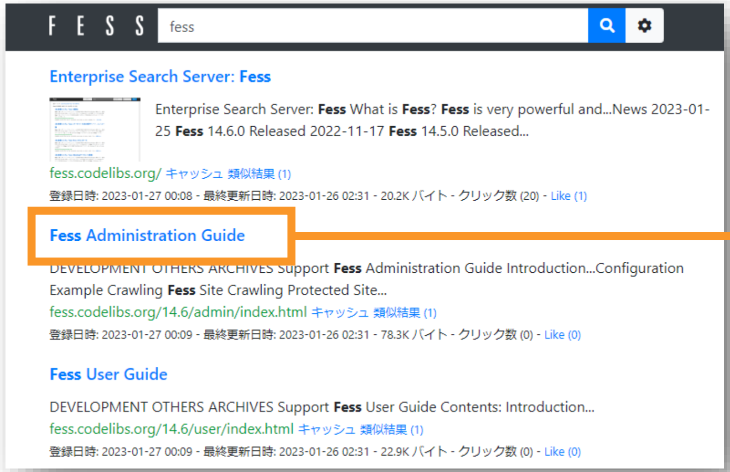
Approach 2



Open the file locally and read its content → much faster

Problem with approach 2

File path pointed to the local file on the Fess instance



Enterprise Search Server: Fess

Enterprise Search Server: **Fess** What is **Fess**? **Fess** is very powerful and...News 2023-01-25 **Fess** 14.6.0 Released 2022-11-17 **Fess** 14.5.0 Released...

fess.codelibs.org/ キャッシュ 類似結果 (1)

登録日時: 2023-01-27 00:08 - 最終更新日時: 2023-01-26 02:31 - 20.2K バイト - クリック数 (20) - Like (1)

Fess Administration Guide

DEVELOPMENT OTHERS ARCHIVES Support **Fess** Administration Guide Introduction...Configuration Example Crawling **Fess** Site Crawling Protected Site...

fess.codelibs.org/14.6/admin/index.html キャッシュ 類似結果 (1)

登録日時: 2023-01-27 00:09 - 最終更新日時: 2023-01-26 02:31 - 78.3K バイト - クリック数 (0) - Like (0)

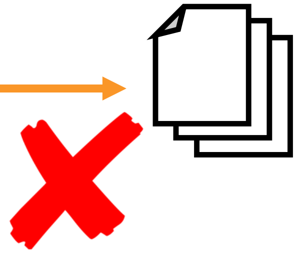
Fess User Guide

DEVELOPMENT OTHERS ARCHIVES Support **Fess** User Guide Contents: Introduction...

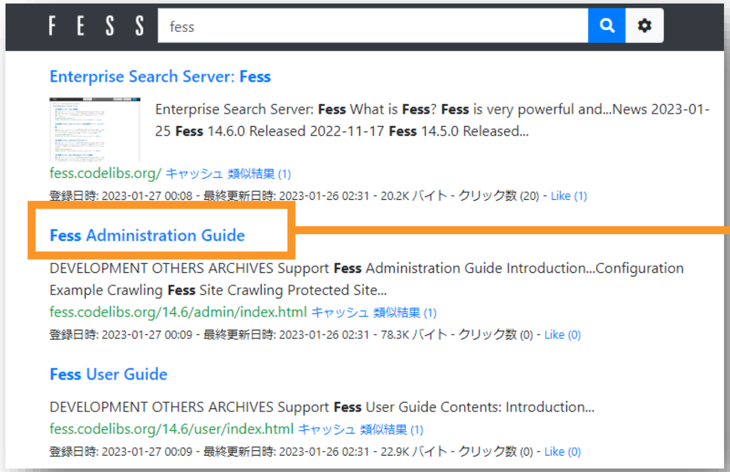
fess.codelibs.org/14.6/user/index.html キャッシュ 類似結果 (1)

登録日時: 2023-01-27 00:09 - 最終更新日時: 2023-01-26 02:31 - 22.9K バイト - クリック数 (0) - Like (0)

file:///home/user/repo/myproject/README.md



Patched web driver client re-maps the filesystem path to URL



Enterprise Search Server: Fess

Enterprise Search Server: **Fess** What is **Fess**? **Fess** is very powerful and...News 2023-01-25 **Fess** 14.6.0 Released 2022-11-17 **Fess** 14.5.0 Released...

fess.codelibs.org/ キャッシュ 類似結果 (1)

登録日時: 2023-01-27 00:08 - 最終更新日時: 2023-01-26 02:31 - 20.2K バイト - クリック数 (20) - Like (1)

Fess Administration Guide

DEVELOPMENT OTHERS ARCHIVES Support **Fess** Administration Guide Introduction...Configuration Example Crawling **Fess** Site Crawling Protected Site...

fess.codelibs.org/14.6/admin/index.html キャッシュ 類似結果 (1)

登録日時: 2023-01-27 00:09 - 最終更新日時: 2023-01-26 02:31 - 78.3K バイト - クリック数 (0) - Like (0)

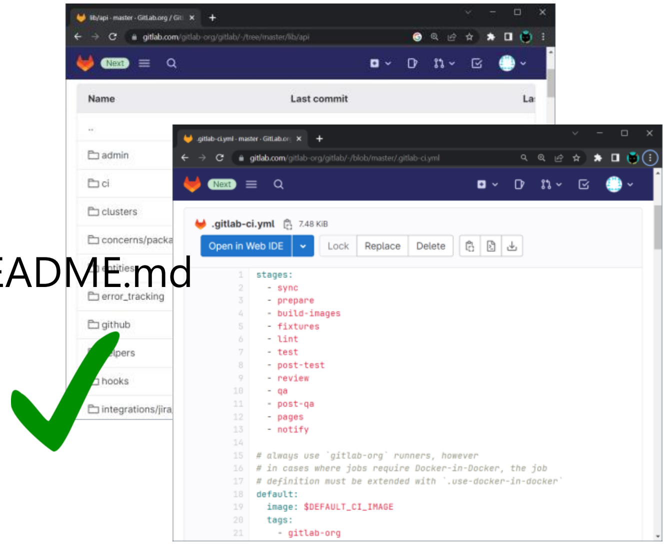
Fess User Guide

DEVELOPMENT OTHERS ARCHIVES Support **Fess** User Guide Contents: Introduction...

fess.codelibs.org/14.6/user/index.html キャッシュ 類似結果 (1)

登録日時: 2023-01-27 00:09 - 最終更新日時: 2023-01-26 02:31 - 22.9K バイト - クリック数 (0) - Like (0)

https://mygitlab.io/myproject/-/blob/main/README.md



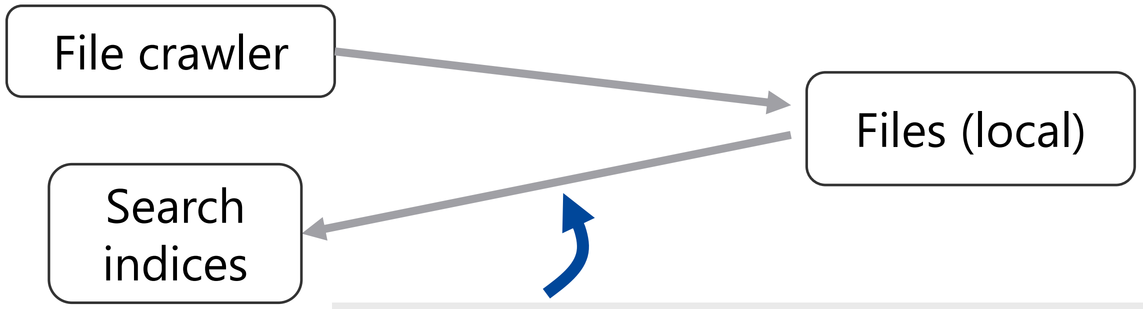
```
stages:
  - sync
  - prepare
  - build-images
  - fixtures
  - lint
  - test
  - post-test
  - review
  - qa
  - post-qa
  - pages
  - notify

# always use `gitlab-org` runners, however
# in cases where jobs require Docker-in-Docker, the job
# definition must be extended with `.use-docker-in-docker`
default:
  image: $DEFAULT_CI_IMAGE
  tags:
    - gitlab-org
```

Patching file system client (re-mapping file system paths to URLs)

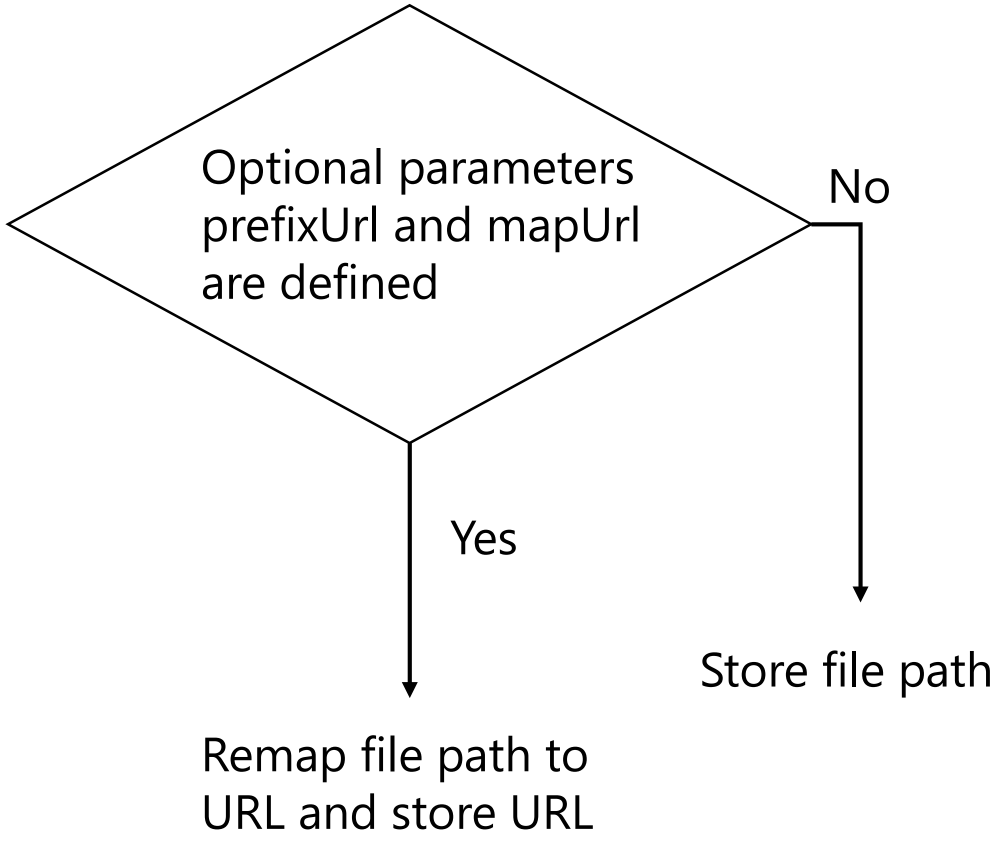
Fess admin console
(file crawling configuration)

URLs	https://mygitlab.io
Included URLs	https://mygitlab\.io/.+
…	…
Config Parameters	**client.prefixUrl=/home/fess/repo/mygroup/ client.mapUrl="https://mygitlab.io/mygroup/" + project + "/-/blob/main/" + relativeFilePath**

The patched config parser checks for optional **client.prefixUrl** and **client.mapUrl** parameters and stores them if they are present



```
If( client.prefixUrl && client.mapUrl ) {
    url = remap(filepath)
  }
```


Optional parameters prefixUrl and mapUrl are defined
No
Yes
Store file path
Remap file path to URL and store URL

Contents

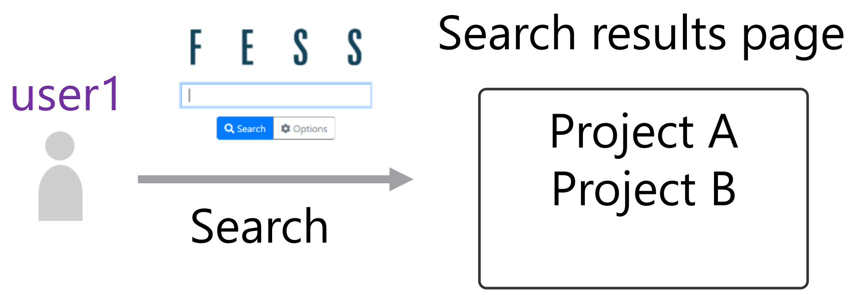
01 Introduction
02 Some background
03 Fess and enterprise search
04 Customization (patching Fess)
05 Automating configurations
06 Conclusion

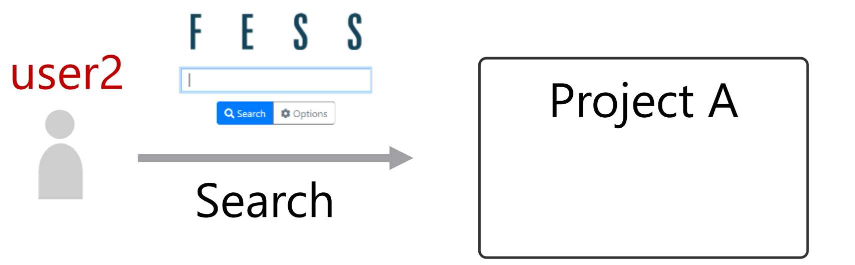
Automating configurations

- Manual configuration using GUI became impractical as the number of configurations increased.
- For each Fess instance,
  - More than 10 configurations
  - Several hundred web crawlers to create

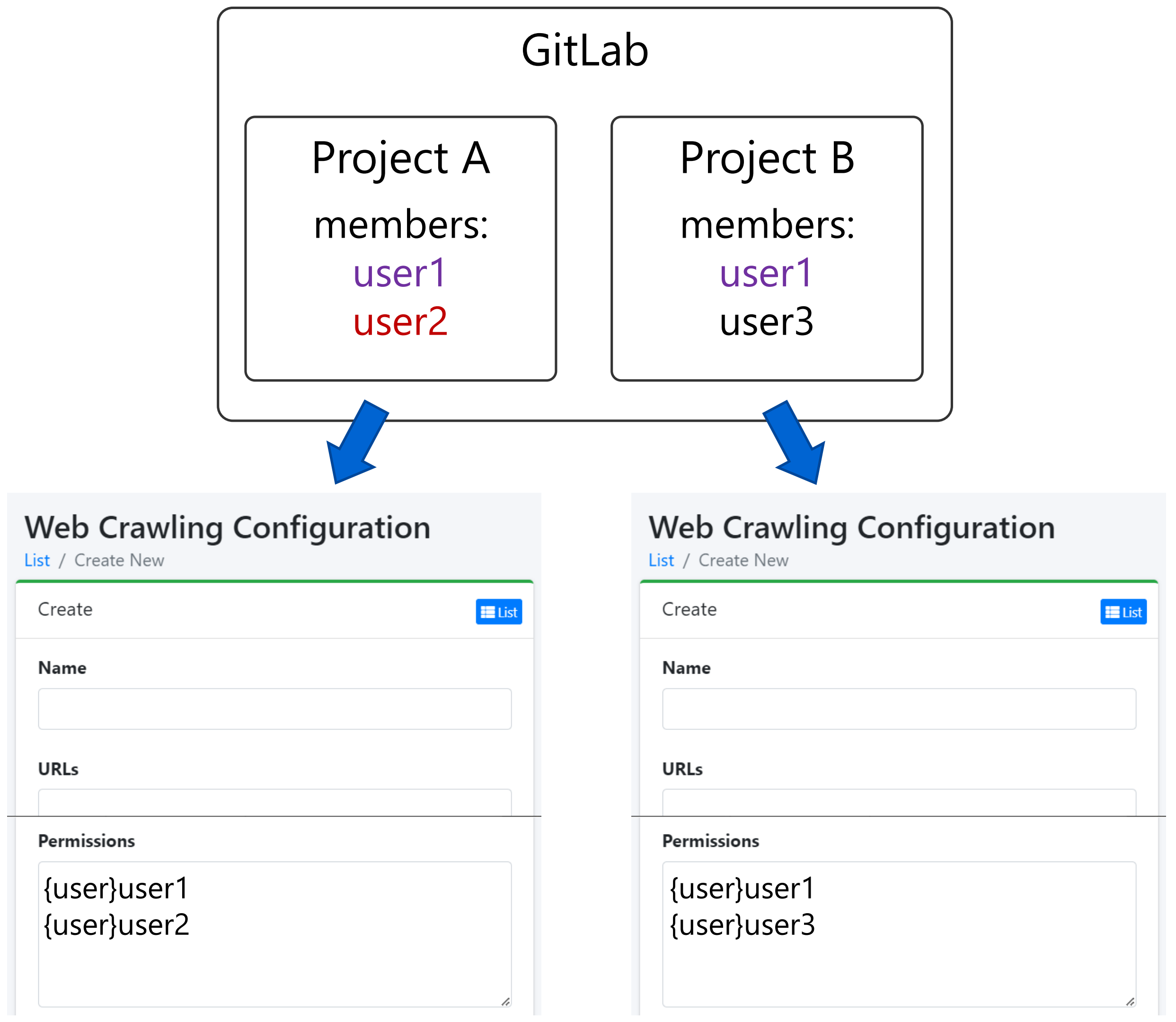
- Without automation, the administrator would have to
  - manually edit lots of config files.
  - do repetitive yet complicated GUI operations to create web crawlers, web auth objects, schedulers, etc. on admin console.

Why automate





Search results should show only the resources (pages, files, etc.) in projects the user has access to.



Web crawler for project A — Permissions:

```
{user}user1
{user}user2
```

Web crawler for project B — Permissions:

```
{user}user1
{user}user3
```

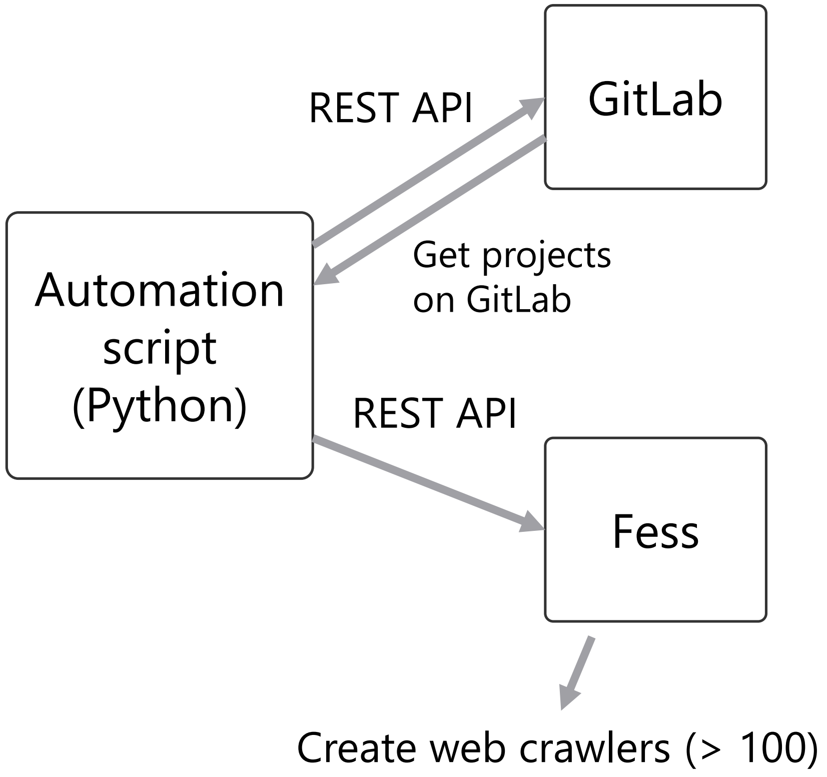
Web crawler for project A

Web crawler for project B

Need to create as many web crawlers as the number of the GitLab projects. → …

Automation example using Fess APIs

Creating web crawlers for each GitLab project:



```
# Get all the projects on GitLab (GitLab API)
projects = gitlab.get_all_projects()

for project in projects:
  # Get all members of project
  users = gitlab.get_project_members(project)

  # Create web crawler for project (Fess API)
  webcrawler = fess.create_webcrawler(project.url, users, ...)

  # Create web authentication for the web crawler
  webauth = fess.create_webauth(project.url + '/login', webcrawler,
  username, password)

  # Create job scheduler for the web crawler
  ...
```

Contents

01 Introduction

02 Some background

03 Fess and enterprise search

04 Customization (patching Fess)

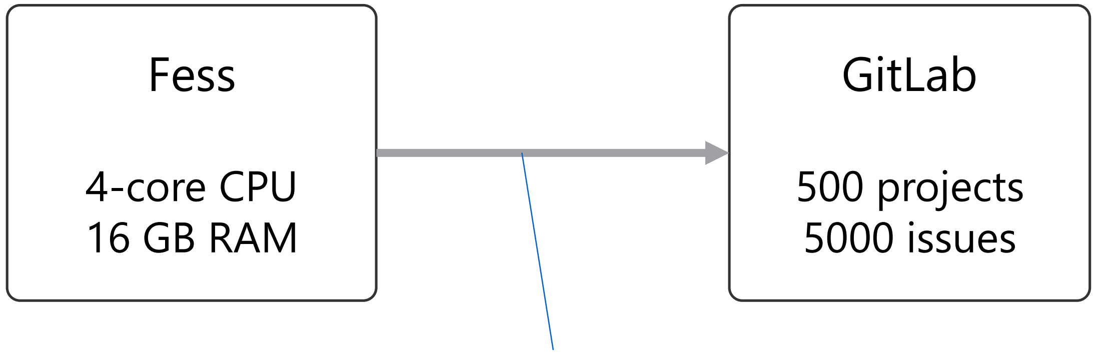
05 Automating configurations

06 Conclusion

Conclusion

- Did Fess solve our problems?
  - Can’t search across tools
    - → Solved. Our users can now search content management tools laterally.
  - Can’t search texts inside binary files
    - → Solved. Our users can find PDF and Office documents by texts contained in them.

- Web crawler performance
  - What our preliminary deployments revealed



Takes about a couple of days
to index everything

Committed to People,
Committed to the Future.

Appendix: Crawling dynamic web pages

- Patches merged upstream
- **Patches kept proprietary**

1. Fess's webdriver-based crawler client can index most of the dynamically loaded contents (Fess 13)
   - You need to change the crawler client from the default to a webdriver-based one, which requires editing config files (XML) and downloading and copying jar files to a specific directory (but no patching is required).
   - Web driver client tends to be less tested/mature; we fixed memory leaks and the PRs were accepted and merged upstream.

2. Our dev team wrote a patch to index contents on GitLab 15
   - The stock webdriver client didn't support issue pages on GitLab 15
   - **Patched web driver client to wait for the specified elements to appear on the page.**

- Starting Fess 14, webdriver client is no longer supported.
- A new crawler client powered by playwright is on its way for Fess 14, but no roadmap has been given by CodeLibs

TOSHIBA