

DOCUMENT CENTER

GitHub: <https://github.com/DoCentre>

Website: <http://docentre.codex.tw>



DOCENTRE

Team 11

312551105 羅羽軒 312552026 蔡濟謙

312551073 賴奕廷 312551124 馮信華

112550052 林哲安



Architecture



SOFTWARE ARCH.

Layered Architecture Pattern

DOCUMENT CENTER SOFTWARE ARCHITECTURE

User Interface



FRONTEND

Document	
POST	/document Create document
DELETE	/document Delete document
POST	/document/add/viewer Add viewer
POST	/document/content Get document content
POST	/document/histories Get document histories
PUT	/document/update Update document
PUT	/document/update/status Set document status
POST	/documents/approver Get approver documents
POST	/documents/author Get author documents
POST	/documents/verify Get verify documents

Restful API

User

POST	/login Login a user
POST	/user Create a user
POST	/users Get users by username

Monolithic application

Presentation Controller



BACKEND

Application Service

Document Service

User Service

Persistence Repository



DATABASE

Data Entity





SOFTWARE ARCH.

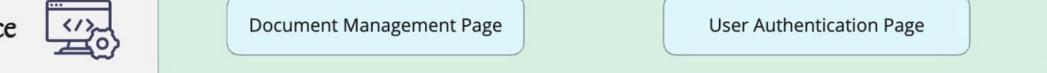
User Interface



Document Request

```
controllers
  - document_controller.go
  - user_controller.go
```

```
services
  - document_service.go
  - user_service.go
```



FRONTEND

Document		
POST	/document	Create document
DELETE	/document	Delete document
POST	/document/add/viewer	Add viewer
POST	/document/content	Get document content
POST	/document/histories	Get document histories
PUT	/document/update	Update document
PUT	/document/update/status	Set document status
POST	/documents/approver	Get approver documents
POST	/documents/author	Get author documents
POST	/documents/verify	Get verify documents

Restful API

User		
POST	/login	Login a user
POST	/user	Create a user
POST	/users	Get users by username

User Request

Monolithic application

Presentation Controller



Document Controller

User Controller

Application Service

Document Service

User Service

Persistence Repository



Document Repository

User Repository

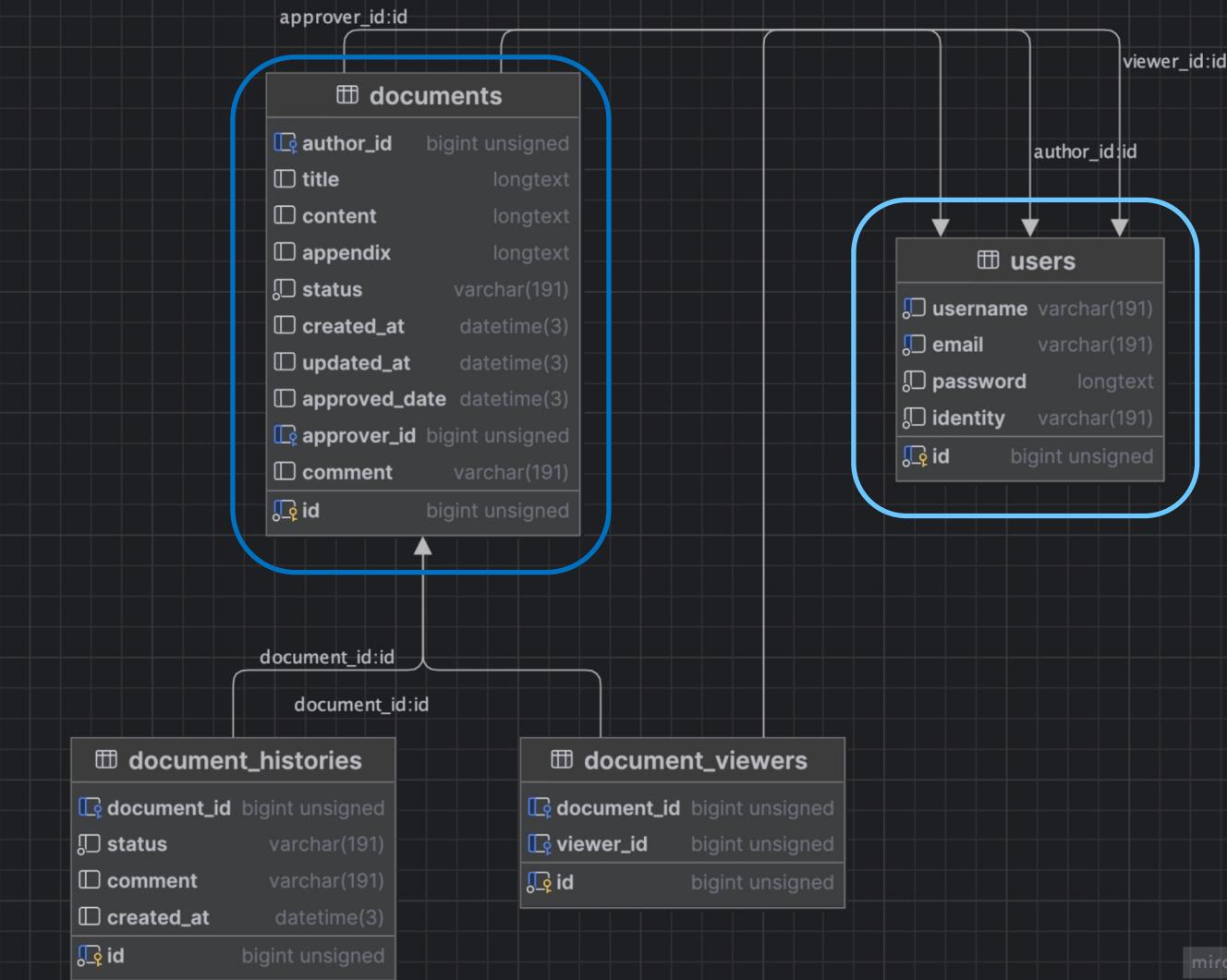


BACKEND

Entity Store & Retrieve



SOFTWARE ARCH. ER DIAGRAM

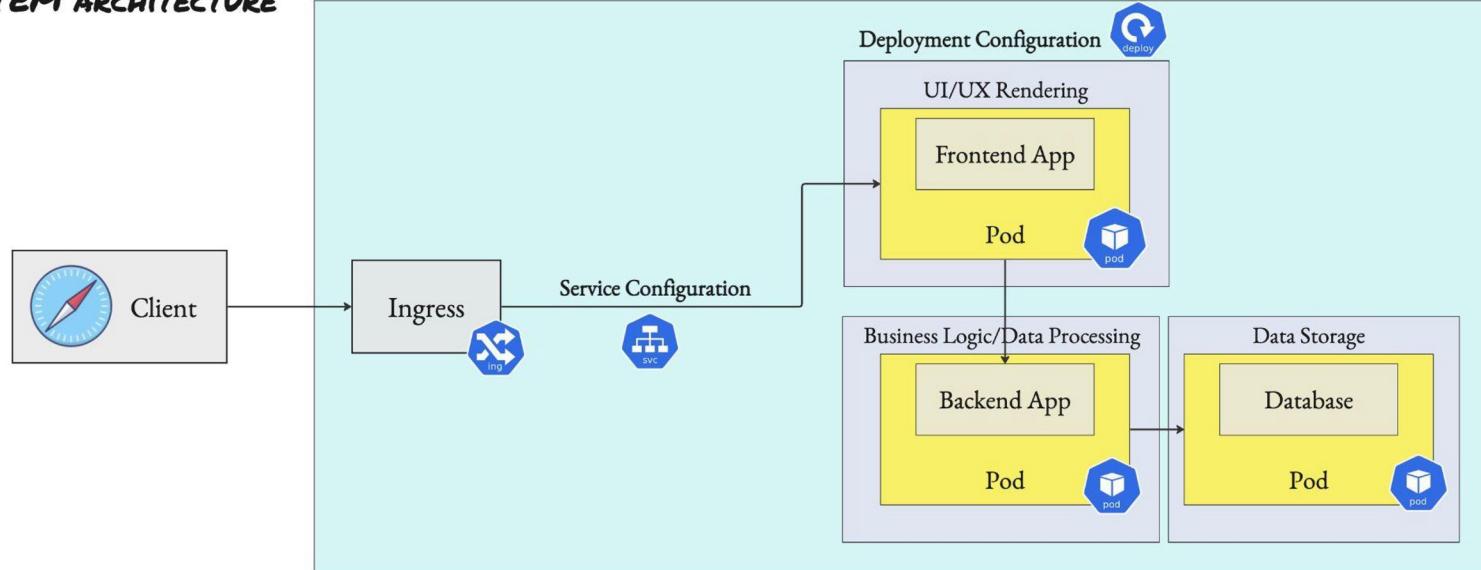




SYSTEM ARCH.

DOCUMENT CENTER SYSTEM ARCHITECTURE

GKE (k8s in GCP)



利用 k8s 架構，Ingress 接收來自瀏覽器的 HTTP 請求，Service 將流量轉發到 Deployment 中運行的 Pod，確保應用的可用性和可擴展性。Deployment 管理 Pod 的 Replica 數量和健康狀態，以確保應用正常運行。

良好的 Software Arch. 及 System Arch. (使用 Layer Arch. & k8s) 同時確保了開發與維護系統的可靠性，同時也提高開發團隊的效率和協作能力。



User Story

EPIC

身 為：公司的一名員工，
我 想 要：完善的文件管理系統，
這 樣 才 能：有效地管理所有文件流程。

身 為：編輯者
我 想 要：撰寫文件的功能
這 樣 才 能：完成交辦的任務

身 為：審核者
我 想 要：審核送審的文件
這 樣 才 能：確保文件的品質

USER
STORY

- 文字編輯
- 圖片上傳
- 選擇審核者
- 送出審核
- 編輯已審核文件

- 收到文件審核通知
- 待審核文件列表
- 退回說明欄位
- 完成/退回審核

TASK



EPIC

身 為：公司的一名員工，
我 想 要：完善的文件管理系統，
這 樣 才 能：有效地管理所有文件流程。

身 為：觀看者
我 想 要：閱覽已審核的文件
這 樣 才 能：確認是否有需要協助的項目

身 為：管理者
我 想 要：觀看所有文件狀態
這 樣 才 能：追蹤每個工作進度

USER STORY

- 查看已審核文件

- 追蹤文件進度
- 查看文件內容
- 編輯文件審核者

TASK



Twelve factors

TWELVE FACTORS

1. **Codebase:** Host on GitHub; deploy through CD.
2. **Dependencies:** Golang has its dependencies locked in `go.mod/sum`; JavaScript has `package.json/package-lock.json`.
3. **Config:** Specify customized configuration in `.env` (environments variables).
4. **Backing Services:** Interact with database through `ORM`, allows migrating and changing database easier.

TWELVE FACTORS

- 5. Processes: Both frontend and backend are **stateless**.
- 6. Port Binding:
 - 5173 -> frontend, 5273 -> OAuth
 - 8080 -> backend, 3306 -> MySQL
- 7. Concurrency: Scaling out using **Kubernetes**.
- 8. Disposability: Fast startup and graceful shutdown with **Kubernetes** and **docker-compose**.
- 9. Dev/Prod Parity: Run GitHub CD when merging into the main branch, deploying production image into GCP.



Testing to Code Quality

- CI & CD
- Pre-Commit
- Static Code Analysis
 - SonarQube

UNIT TESTS

```
~/Desktop/NYCU/112-2/Cloud-Native-Dev/DoCentre main*
> make test
go test "-test.shuffle=on" -v github.com/docentre/docentre github.com/docentre/docentre/controllers github.com/docentre/docentre/docs github.com/docentre/docentre/models github.com/docentre/repositories github.com/docentre/router github.com/docentre/services
?      github.com/docentre/docentre/controllers [no test files]
?      github.com/docentre/docentre/docs [no test files]
?      github.com/docentre/docentre/repositories [no test files]
?      github.com/docentre/docentre/router [no test files]
-test.shuffle 1717310094362770000
*** RUN  TestCheckHealth
[GIN-debug] [WARNING] Creating an Engine instance with the Logger and Recovery middleware already attached.

[GIN-debug] [WARNING] Running in "debug" mode. Switch to "release" mode in production.
- using env: export GIN_MODE=release
- using code: gin.SetMode(gin.ReleaseMode)

[GIN-debug] GET   /metrics          --> github.com/docentre/docentre/router.setPrometheus.WrapH.func1 (3 handlers)
[GIN-debug] GET   /health           --> github.com/docentre/docentre/router.CheckHealth (3 handlers)
[GIN-debug] POST  /user            --> github.com/docentre/docentre/controllers.UserCreate (3 handlers)
[GIN-debug] POST  /login           --> github.com/docentre/docentre/controllers.UserLogin (3 handlers)
[GIN-debug] POST  /users           --> github.com/docentre/docentre/controllers.GetUsersByUsername (3 handlers)
[GIN-debug] POST  /document        --> github.com/docentre/docentre/controllers.CreateDocument (3 handlers)
[GIN-debug] PUT   /document/update --> github.com/docentre/docentre/controllers.UpdateDocument (3 handlers)
[GIN-debug] POST  /document/add/viewer --> github.com/docentre/docentre/controllers.AddViewer (3 handlers)
[GIN-debug] POST  /documents/author --> github.com/docentre/docentre/controllers.GetAuthorDocuments (3 handlers)
[GIN-debug] POST  /documents/approver --> github.com/docentre/docentre/controllers.GetApproverDocuments (3 handlers)
[GIN-debug] POST  /documents/verify  --> github.com/docentre/docentre/controllers.GetVerifyDocuments (3 handlers)
[GIN-debug] PUT   /document/update/status --> github.com/docentre/docentre/controllers.SetDocumentStatus (3 handlers)
[GIN-debug] POST  /document/histories --> github.com/docentre/docentre/controllers.GetDocumentHistories (3 handlers)
[GIN-debug] DELETE /document       --> github.com/docentre/docentre/controllers.DeleteDocument (3 handlers)
[GIN-debug] POST  /document/content --> github.com/docentre/docentre/controllers.GetDocumentContent (3 handlers)
[GIN-debug] GET   /swagger/*any    --> github.com/swaggo/gin-swagger.customWrapHandler.func1 (3 handlers)
[GIN] 2024/06/02 - 14:34:54 | 200 |     86.875µs |             | GET      "/health"
--- PASS: TestCheckHealth (0.00s)
PASS
ok    github.com/docentre/docentre    0.827s
-test.shuffle 1717310093679496000
*** RUN  TestUser
--- PASS: TestUser (0.00s)
--- PASS: TestUserLogin (0.00s)
--- PASS: TestDocumentHistory (0.00s)
--- PASS: TestDocumentHistory (0.00s)
*** RUN  TestDocument
--- PASS: TestDocument (0.00s)
*** RUN  TestDocumentViewer
--- PASS: TestDocumentViewer (0.00s)
PASS
ok    github.com/docentre/docentre/models   0.142s
-test.shuffle 1717310093803043000
*** RUN  TestUserLogin
--- PASS: TestUserLogin (0.00s)
*** RUN  TestGetUsersByUsername
--- PASS: TestGetUsersByUsername (0.00s)
*** RUN  TestGetApproverDocuments
```

CI & CD

```
Run linter
1 ► Run make vet && make lint
4 go vet github.com/docentre/docentre github.com/docentre/docentre/controllers
github.com/docentre/docentre/docs github.com/docentre/docentre/models
github.com/docentre/docentre/repositories github.com/docentre/docentre/router
github.com/docentre/docentre/services
5 staticcheck is not installed. Installing...
6 staticcheck github.com/docentre/docentre github.com/docentre/docentre/controllers
github.com/docentre/docentre/docs github.com/docentre/docentre/models
github.com/docentre/docentre/repositories github.com/docentre/docentre/router
github.com/docentre/docentre/services
7 Error: services/document_service.go:120:38: error strings should not be capitalized (ST1005)
8 make: *** [Makefile:61: lint] Error 1
9 Error: Process completed with exit code 2.
```

All checks have passed

3 successful checks

- | | | | | |
|--|--|--------------------------------|-------------------|-------------------------|
| | | Deploy / docker (push) | Successful in 40s | Details |
| | | Run Tests / lint (push) | Successful in 31s | Details |
| | | Deploy / k8s (push) | Successful in 21s | Details |

PRE-COMMIT

First Commit:
Auto fixed



Next Commit:
Passed!

```
~/Desktop/NYCU/112-2/Cloud-Native-Dev/DoCentre feature-add-dockerfile-and-expose-pr
> git commit -am "bad format :("
trim trailing whitespace.....Passed
fix end of files.....Failed
- hook id: end-of-file-fixer
- exit code: 1
- files were modified by this hook
```

Fixing router/router.go

```
check yaml.....(no files to check)Skipped
check for added large files.....Passed
check for merge conflicts.....Passed
detect private key.....Passed
check that scripts with shebangs are executable.....Passed
format code.....Passed
```

```
~/Desktop/NYCU/112-2/Cloud-Native-Dev/DoCentre feature-add-dockerfile-and-expose-pr
> git commit -am "bad format fixed! :)"
trim trailing whitespace.....(no files to check)Skipped
fix end of files.....(no files to check)Skipped
check yaml.....(no files to check)Skipped
check for added large files.....(no files to check)Skipped
check for merge conflicts.....(no files to check)Skipped
detect private key.....(no files to check)Skipped
check that scripts with shebangs are executable.....(no files to check)Skipped
format code.....Passed
```

```
On branch feature-add-dockerfile-and-expose-prometheus-metrics
Your branch is up to date with 'origin/feature-add-dockerfile-and-expose-prometheus'

nothing to commit, working tree clean
```

STATIC CODE ANALYSE: SONARQUBE

Local setup
sonar scan &
sonarqube server



localhost:9000/dashboard?id=docentre&codeScope=overall

sonarqube Projects Issues Rules Quality Profiles Quality Gates Administration More ? A

docentre / main ?

Overview Issues Security Hotspots Measures Code Activity Project Settings Project Information

main 3.1k Lines of Code • Version 1.0 • Set as homepage Last analysis 2 minutes ago

Quality Gate Status ? Passed

New Code Overall Code

Security	Reliability	Maintainability
0 Open issues	0 Open issues	15 Open issues
0 H 0 M 0 L	0 H 0 M 0 L	8 H 7 M 0 L

Accepted issues 0 Coverage 0.0% On 467 lines to cover.

Duplications Security Hotspots

Enjoy your sparkling clean code!

This screenshot shows the SonarQube dashboard for the project 'docentre'. The main status is 'Passed' with a green checkmark icon. The dashboard provides an overview of code quality across three dimensions: Security, Reliability, and Maintainability. In Security, there are 0 open issues. In Reliability, there are 0 open issues. In Maintainability, there are 15 open issues, with 8 high priority, 7 medium priority, and 0 low priority. The 'Accepted issues' section shows 0 valid issues that were not fixed. Coverage is currently at 0.0%, with 467 lines still needed to be covered. The 'Duplications' and 'Security Hotspots' sections are also visible at the bottom of the dashboard.

STATIC CODE ANALYSE: SONARQUBE

The screenshot shows the SonarQube interface for a project named "docentre". The "Security Hotspots" tab is selected. On the left, there's a summary card showing 0.0% Security Hotspots Reviewed. Below it, there are four buttons: To review, Acknowledged, Fixed, and Safe. A message indicates there are 2 Security Hotspots. The review priority is set to Medium, and the category is Permission. The assignee is Not assigned.

Security Hotspot Details:

- Message:** Adding files recursively might inadvertently add sensitive data to the container. Make sure it is safe here.
- Review priority:** Medium
- Category:** Permission
- Assignee:** Not assigned

Code Snippet:

```
FROM golang:alpine AS build-env
ADD . /src
```

A red box highlights the warning message above the code snippet.

Code Block Buttons:

- Where is the risk?
- What's the risk?
- Assess the risk
- How can I fix it?
- Activity

Bottom Status Bar:

localhost:9000



Monitor Metrics

Golang Prometheus Default metrics (Expose from Backend)

```
# HELP go_gc_duration_seconds A summary of the pause duration of garbage collection cycles.
# TYPE go_gc_duration_seconds summary
go_gc_duration_seconds{quantile="0"} 0.000140349
go_gc_duration_seconds{quantile="0.25"} 0.000160711
go_gc_duration_seconds{quantile="0.5"} 0.000172352
go_gc_duration_seconds{quantile="0.75"} 0.000185403
go_gc_duration_seconds{quantile="1"} 0.000431624
go_gc_duration_seconds_sum 1.2820405959999999
go_gc_duration_seconds_count 7102
# HELP go_goroutines Number of goroutines that currently exist.
# TYPE go_goroutines gauge
go_goroutines 9
# HELP go_info Information about the Go environment.
# TYPE go_info gauge
go_info{version="go1.22.3"} 1
# HELP go_memstats_alloc_bytes Number of bytes allocated and still in use.
# TYPE go_memstats_alloc_bytes gauge
go_memstats_alloc_bytes 1.0793576e+07
# HELP go_memstats_alloc_bytes_total Total number of bytes allocated, even if freed.
# TYPE go_memstats_alloc_bytes_total counter
go_memstats_alloc_bytes_total 1.162943912e+09
# HELP go_memstats_buck_hash_sys_bytes Number of bytes used by the profiling bucket hash table.
# TYPE go_memstats_buck_hash_sys_bytes gauge
go_memstats_buck_hash_sys_bytes 8828
# HELP go_memstats_frees_total Total number of frees.
# TYPE go_memstats_frees_total counter
go_memstats_frees_total 1.270675e+06
# HELP go_memstats_gc_sys_bytes Number of bytes used for garbage collection system metadata.
# TYPE go_memstats_gc_sys_bytes gauge
go_memstats_gc_sys_bytes 3.45952e+06
# HELP go_memstats_heap_alloc_bytes Number of heap bytes allocated and still in use.
# TYPE go_memstats_heap_alloc_bytes gauge
go_memstats_heap_alloc_bytes 1.0793576e+07
# HELP go_memstats_heap_idle_bytes Number of heap bytes waiting to be used.
# TYPE go_memstats_heap_idle_bytes gauge
go_memstats_heap_idle_bytes 4.0845312e+07
# HELP go_memstats_heap_inuse_bytes Number of heap bytes that are in use.
# TYPE go_memstats_heap_inuse_bytes gauge
go_memstats_heap_inuse_bytes 1.318912e+07
# HELP go_memstats_heap_objects Number of allocated objects.
# TYPE go_memstats_heap_objects gauge
go_memstats_heap_objects 13307
# HELP go_memstats_heap_released_bytes Number of heap bytes released to OS.
# TYPE go_memstats_heap_released_bytes gauge
go_memstats_heap_released_bytes 3.7855232e+07
# HELP go_memstats_heap_sys_bytes Number of heap bytes obtained from system.
# TYPE go_memstats_heap_sys_bytes gauge
go_memstats_heap_sys_bytes 5.4034432e+07
# HELP go_memstats_last_gc_time_seconds Number of seconds since 1970 of last garbage collection.
# TYPE go_memstats_last_gc_time_seconds gauge
go_memstats_last_gc_time_seconds 1.7173161546543138e+09
```

```
go_memstats_mspan_inuse_bytes 99360
# HELP go_memstats_mspan_sys_bytes Number of bytes used for mspan structures obtained from system.
# TYPE go_memstats_mspan_sys_bytes gauge
go_memstats_mspan_sys_bytes 261120
# HELP go_memstats_next_gc_bytes Number of heap bytes when next garbage collection will take place.
# TYPE go_memstats_next_gc_bytes gauge
go_memstats_next_gc_bytes 2.2124064e+07
# HELP go_memstats_other_sys_bytes Number of bytes used for other system allocations.
# TYPE go_memstats_other_sys_bytes gauge
go_memstats_other_sys_bytes 407268
# HELP go_memstats_stack_inuse_bytes Number of bytes in use by the stack allocator.
# TYPE go_memstats_stack_inuse_bytes gauge
go_memstats_stack_inuse_bytes 491520
# HELP go_memstats_stack_sys_bytes Number of bytes obtained from system for stack allocator.
# TYPE go_memstats_stack_sys_bytes gauge
go_memstats_stack_sys_bytes 491520
# HELP go_memstats_sys_bytes Number of bytes obtained from system.
# TYPE go_memstats_sys_bytes gauge
go_memstats_sys_bytes 5.8678288e+07
# HELP go_threads Number of OS threads created.
# TYPE go_threads gauge
go_threads 7
# HELP process_cpu_seconds_total Total user and system CPU time spent in seconds.
# TYPE process_cpu_seconds_total counter
process_cpu_seconds_total 41.58
# HELP process_max_fds Maximum number of open file descriptors.
# TYPE process_max_fds gauge
process_max_fds 1.048576e+06
# HELP process_open_fds Number of open file descriptors.
# TYPE process_open_fds gauge
process_open_fds 10
# HELP process_resident_memory_bytes Resident memory size in bytes.
# TYPE process_resident_memory_bytes gauge
process_resident_memory_bytes 6.2619648e+07
# HELP process_start_time_seconds Start time of the process since unix epoch in seconds.
# TYPE process_start_time_seconds gauge
process_start_time_seconds 1.71644548021e+09
# HELP process_virtual_memory_bytes Virtual memory size in bytes.
# TYPE process_virtual_memory_bytes gauge
process_virtual_memory_bytes 1.66033408e+09
# HELP process_virtual_memory_max_bytes Maximum amount of virtual memory available in bytes.
# TYPE process_virtual_memory_max_bytes gauge
process_virtual_memory_max_bytes 1.8446744073709552e+19
# HELP promhttp_metric_handler_requests_in_flight Current number of scrapes being served.
# TYPE promhttp_metric_handler_requests_in_flight gauge
promhttp_metric_handler_requests_in_flight 1
# HELP promhttp_metric_handler_requests_total Total number of scrapes by HTTP status code.
# TYPE promhttp_metric_handler_requests_total counter
promhttp_metric_handler_requests_total{code="200"} 726
promhttp_metric_handler_requests_total{code="500"} 0
promhttp_metric_handler_requests_total{code="503"} 0
```

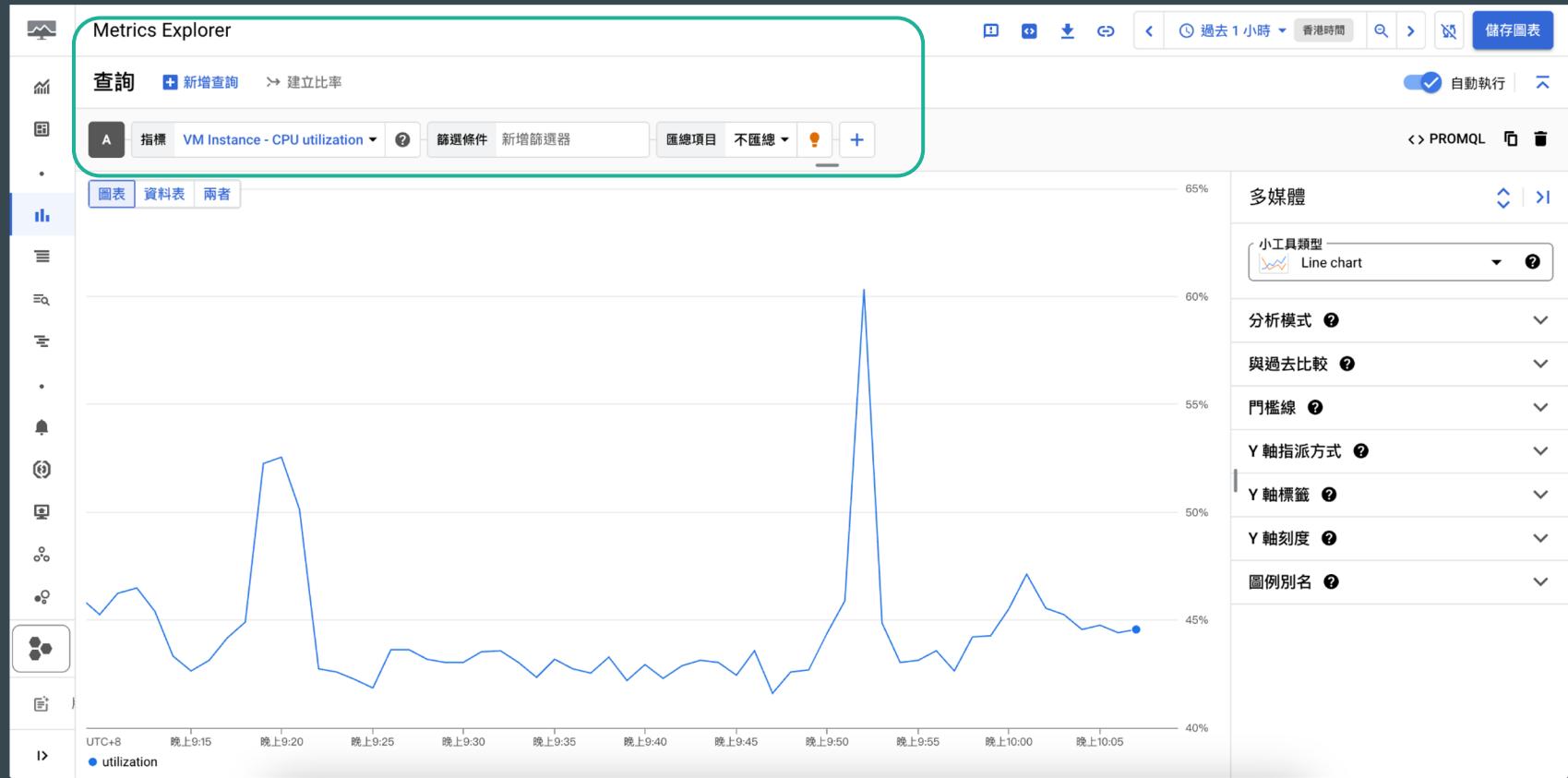
Prometheus Metrics - Dashboard Page

The screenshot shows a web browser window displaying a Prometheus Metrics dashboard. The URL is docentre.codex.tw/dashboard. The dashboard includes the following sections:

- System Status**:
 - Memory Usage: 30.98 MB / 1000 MB (progress bar: ~3%)
 - Process Time: 0.25 ms / 1 ms (progress bar: ~25%)
- HTTP Traffic**:
 - Total : 130
 - HTTP 200: 130
 - HTTP 500: 0
 - HTTP 503: 0
- SHOW DETAIL** button
- A summary of the pause duration of garbage collection cycles.**

Type	summary
go_gc_duration_seconds{quantile="0"}	0.000165669
go_gc_duration_seconds{quantile="0.25"}	0.000230736

GCP VM Instance - Metrics Explorer Example





Beyond Technical ... Collaboration

- Swagger
- Pull Request
 - Code Quality
- GitHub Issues

SWAGGER



Swagger UI

doc.json Explore

Docentre API

doc.json

This is the API documentation for Docentre.

Document

- POST** /document Create document
- DELETE** /document Delete document
- POST** /document/add/viewer Add viewer
- POST** /document/content Get document content
- POST** /document/histories Get document histories
- PUT** /document/update Update document
- PUT** /document/update/status Set document status
- POST** /documents/approver Get approver documents

Swagger UI

PUT /document/update Update document

Update a document that belongs to the author; the author has to be a existing user. if no approver yet, approver_id should be 0.

Parameters

Try it out

Name	Description
body <small>* required</small>	Example Value Model

```
{  
    "appendix": "",  
    "approver_id": 0,  
    "author_id": 1,  
    "content": "Hello, world!",  
    "document_id": 1,  
    "status": "EDIT",  
    "title": "Hello"  
}
```

Parameter content type
application/json

Responses

Response content type application/json

Code	Description
200	OK

<http://docentre.codex.tw/api/swagger/index.html>

PULL REQUEST



Issues 1 Pull requests Actions Projects Wiki Security Insights Settings

Filters is:pr is:closed Labels 9 Milestones 0 New pull request

Clear current search query, filters, and sorts

Author	Label	Projects	Milestones	Reviews	Assignee	Sort
0 Open	12 Closed					
Fix missing record on status change ✓ #13 by Lai-YT was merged 12 hours ago • Approved						
Send email when status is set to approve ✓ enhancement #12 by Lai-YT was merged yesterday • Approved						
feat: add dockerfile and expose prometheus default metrics ✓ #11 by lohsuan was merged yesterday • Approved						
Implement DeleteDocument and GetDocumentContent APIs ✓ #10 by eytodo was merged 2 weeks ago • Approved						
Get document histories ✓ enhancement #9 by Lai-YT was merged 3 weeks ago • Approved						
Set status of document & record in history ✓ #8 by Lai-YT was merged 3 weeks ago • Approved						
Revise response status and body ✓ #6 by Lai-YT was merged 3 weeks ago • Approved						
implement get documents relates API: GetAuthorDocuments, AddViewer, GetViewerDocuments ✓ #5 by lohsuan was merged 3 weeks ago • Approved						
Implement UpdateDocument API ✓ #4 by lohsuan was merged last month • Approved						

CODE QUALITY



Merged

Implement UpdateDocument API #4

lohsuan merged 4 commits into [main](#) from [feature-update-document](#) last month

services/document_service.go Outdated

Hide resolved

```
13    11      if result.Error != nil {  
14    12          return models.Document{}, result.Error  
15    13      }  
16    14  
17    15      return doc, nil  
18    16  }  
17  +  
18  + func UpdateDocument(documentID uint, authorID uint, title string, content string, appendix str
```

Lai-YT last month

Member ...

Instead of accepting a large number of parameters, consider encapsulating them within a `struct`. This approach helps prevent users from inadvertently swapping the order of parameters, especially when they share common types.



lohsuan last month

Member Author ...

I'm not pretty sure. But this is controller's responsibility to make the object type into the flatten primitive type. Otherwise, I only need to pass the entire request body inside Service?



Lai-YT last month

Member ...

Since the `service` is where our business logic resides, it has the authority to define the structure it prefers, which doesn't necessarily have to consist of flattened primitives. This structure is recognized by the service, making it resilient to changes in the actual request body. The controller's responsibility is to convert the request body into this structure.



Lai-YT last month

Member ...

The structure can currently be identical to the request body. While the conversion may seem redundant, I believe it's necessary to provide independence.



GITHUB ISSUES (TRACK PROGRESS)

The image displays two side-by-side screenshots of GitHub issue pages, illustrating the tracking of progress for different projects.

Left Screenshot (APIs Issue #7):

- Title:** APIs #7
- Status:** Open (13 tasks done)
- Author:** Lai-YT opened this issue 29 days ago · 0 comments
- Comment by Lai-YT:** Lai-YT commented 29 days ago · edited (Member) ...
This issue outlines the APIs to support, along with related issues:
- User API Tasks:**
 - Create user (/user)
 - Login (/login)
 - Get users by username, as the username is not unique (/users)
 - The username is partially matched by the database
 - Login with OAuth (properly handled by frontend)
- Document API Tasks:**
 - Create document (/document)
 - Update the content/status of a document (/document/update)
 - Add viewer to a document (/document/add/viewer)
 - Get documents of a viewer (/documents/viewer)
 - Get the documents of an author (/documents/author)
 - Set the status of a document (/document/update/status)

Right Screenshot (Features Issue #1):

- Title:** Features #1
- Status:** Open (25 tasks done)
- Author:** AU2A opened this issue 28 days ago · 0 comments
- Comment by AU2A:** AU2A commented 28 days ago · edited (Member) ...
- Login/Logout (Aura) API Tasks:**
 - Login page
 - Registration page
 - Google OAuth
 - Logout
- Doc edit API Tasks:**
 - Edit doc List page (Aura)
 - Add new doc (Aura)
 - Click to edit doc content (Aura)
 - Edit doc Page (Johnny)
 - route to edit doc page (Aura)
 - Submit API (Johnny)
- Doc approve API Tasks:**



Live Demo

Any Questions?

GitHub: <https://github.com/DoCentre>

Website: <http://docentre.codex.tw>