

Frontend

Stores(pinia)

task.js

status.js

toast.js

theme.js

limitTask.js

filter.js

Composables

task-api.js

status-api.js

limit-api.js

Composables

provides several benefits

- Reusability
- Separation of Concerns
- Maintainability
- Testability
- Consistency
- Enhanced Code Readability



*เปลี่ยนจาก libs มาใช้ composables
เนื่องจากต้องการเรียกใช้ toast เมื่อมีการ
call api เกิดขึ้น -> เปลี่ยนรูปแบบการ
export เป็น closure function

Fetch data when mounted(tasks, statuses, limitTask)

{path: '/task'}

{path: '/status'}

TaskView.vue

StatusView.vue

TaskTable.vue

StatusTable.vue

ConfirmDelete.vue ([mode, object, number])

ConfirmDelete.vue ([mode, object, number])

([closeModal])

([closeModal])

StatusSetting.vue

StatusSetting.vue

([close])

([close])

FilterModal.vue

StatusModal.vue

([close], [applyFilter])

{path: '/task'}

{path: '/:taskId'}

{path: '/:taskId/edit'}

{path: '/add'}

{path: '/status'}

{path: '/:statusId'}

{path: '/:statusId/edit'}

{path: '/add'}

TaskModal.vue

Backend

Entity

LimitTask

- id (Integer)
- limit (Boolean)
- limitMaximumTask (Integer)

TaskV2

- id (Integer)
- title(String)
- description (Integer)
- assignees (String)
- createdOn (Date)
- updatedOn (Date)
- Status (Status)

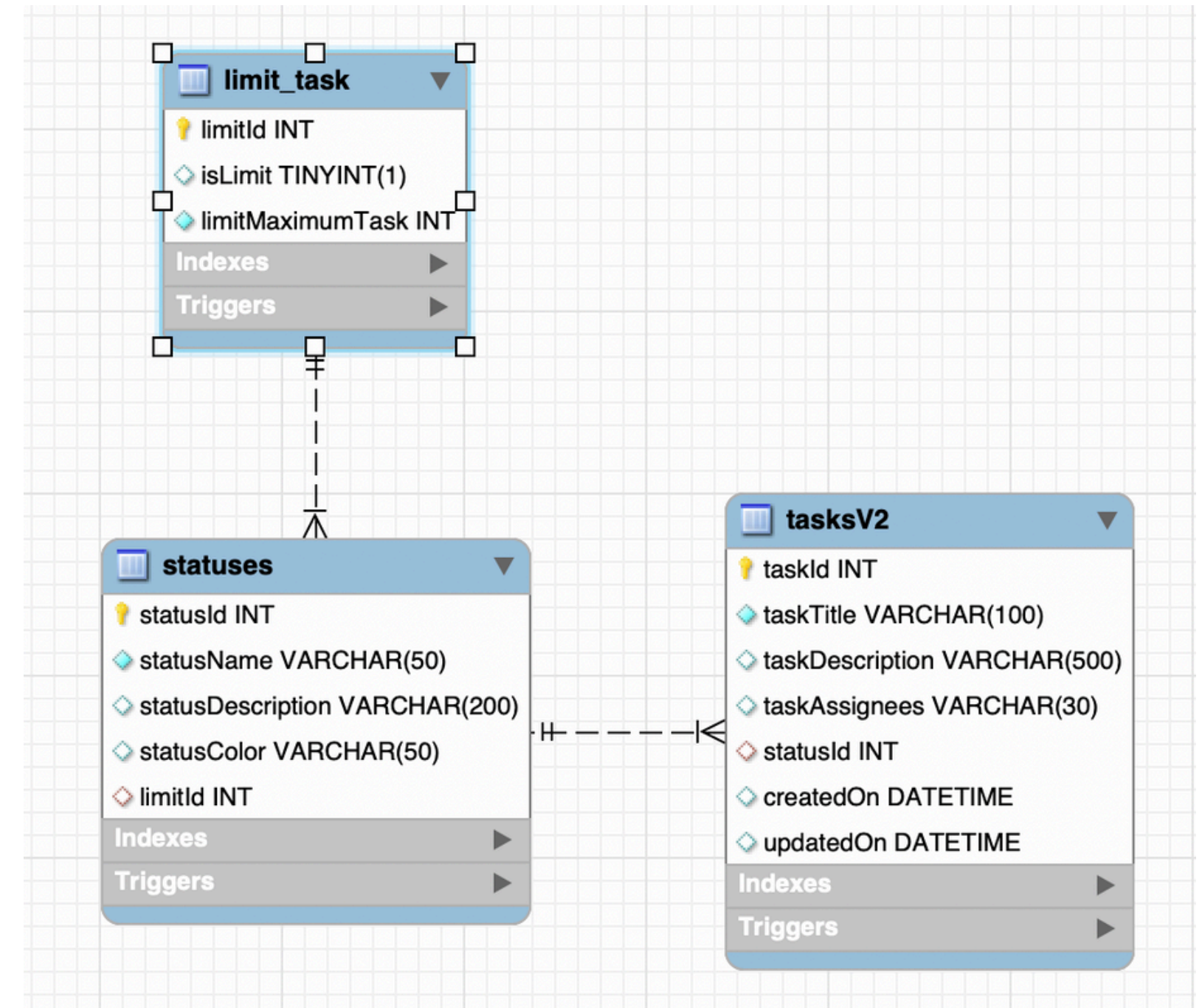
Status

- id (Integer)
- name (String)
- description (String)
- status (color)
- noOfTasks (List)

Task

- id (Integer)
- title(String)
- description (Integer)
- assignees (String)
- createdOn (Date)
- updatedOn (Date)
- Status (String)

Relationnal



DTO

StatusInputDTO

- id (Integer)
- name (String)
- description (String)
- status (color)
- noOfTasks (Integer)

Input →

Status

- id (Integer)
- name (String)
- description (String)
- status (color)
- noOfTasks (List)

Output →

StatusOutputDTO

- id (Integer)
- name (String)
- description (String)
- status (color)
- noOfTasks (Integer)

TaskInputDTO

- id (Integer)
- title(String)
- assignees (String)
- Status (Status)

Input →

TaskV2

- id (Integer)
- title(String)
- description (String)
- assignees (String)
- createdOn (Date)
- updatedOn (Date)
- Status (Status)

Output →

TaskOutputAllFieldDTO

- id (Integer)
- title(String)
- description (String)
- assignees (String)
- createdOn (Date)
- updatedOn (Date)
- Status (StatusOutputDTO)

Task

- id (Integer)
- title(String)
- description (Integer)
- assignees (String)
- createdOn (Date)
- updatedOn (Date)
- Status (String)

Input →

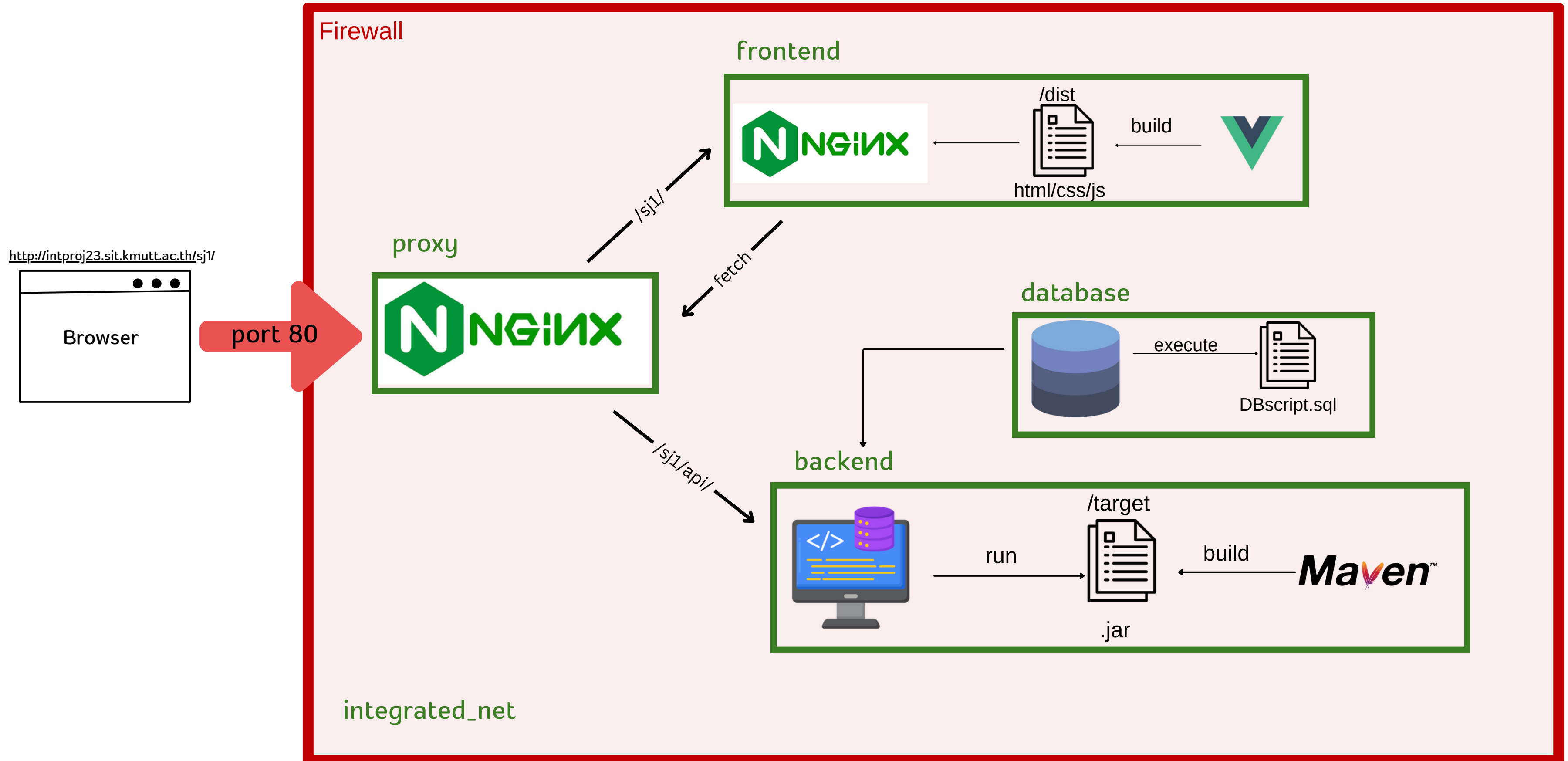
TaskOutputDTO

- id (Integer)
- title(String)
- assignees (String)
- Status (Status)

Output →

Integration

Infrastructure



Frontend

.env.production

```
integrated-1-frontend > $ .env.production
1 # VITE_BASE_URL=http://ip23sj1.sit.kmutt.ac.th:8080/v2
2 VITE_BASE_URL=/sj1/api/v2
3 VITE_BASE=/sj1
```

Fetch API

	✓ composables
JS	limit-api.js
JS	status-api.js
JS	task-api.js

Composables

provides several benefits

- Reusability
- Separation of Concerns
- Maintainability
- Testability
- Consistency
- Enhanced Code Readability

Dockerfile (build frontend)

```
FROM node:20.10.0 as build
COPY . .
RUN rm -rf node_modules/
RUN npm install
RUN npm run build

FROM nginx:alpine as run
COPY --from=build dist /usr/share/nginx/html
COPY nginx.conf /etc/nginx/nginx.conf

EXPOSE 80

CMD ["nginx", "-g", "daemon off;"]
```

Backend

Dockerfile (backend)

```
FROM maven:3.8.3-openjdk-17 AS build
COPY pom.xml .
COPY src ./src
RUN mvn clean package -DskipTests

FROM openjdk:17-jdk-alpine AS run
COPY --from=build target/*.jar app.jar
EXPOSE 8080
CMD ["java", "-jar", "app.jar"]
```

Database

Dockerfile (Database)

```
FROM mysql:latest
ENV MYSQL_USER=dev
ENV MYSQL_PASSWORD=Mysql@sit123
ENV MYSQL_ROOT_PASSWORD=Mysql@sit123
ENV MYSQL_DATABASE=integrated
ENV MYSQL_CHARSET=utf8mb4
ENV MYSQL_COLLATION=utf8mb4_unicode_ci
EXPOSE 3306
COPY scripts/ /docker-entrypoint-initdb.d/
COPY dbscripts/ /dbscripts/

COPY my.cnf /etc/my.cnf
```

Proxy

Dockerfile (Proxy)

```
FROM nginx:alpine as run
COPY default.conf /etc/nginx/conf.d/default.conf

EXPOSE 80

CMD ["nginx", "-g", "daemon off;"]
```

default.conf

```
server {
    listen      80 default_server;
    server_name ip23sj1.sit.kmutt.ac.th;

    location / {
        proxy_pass http://frontend;
    }
    location /api/ {
        proxy_pass http://backend:8080/;
    }
    location /sj1/ {
        proxy_pass http://frontend;
    }
    location /sj1/api/ {
        proxy_pass http://backend:8080/;
    }
}
```

Compose.yaml

service (database)

```
database:
  build: ./integrated-1-database
  container_name: database
  restart: unless-stopped
  networks:
    - integrated_net
  volumes:
    - myapp:/var/lib/mysql
  healthcheck:
    test: ["CMD", "mysqladmin", "ping", "-h", "localhost"]
    interval: 5s
    timeout: 5s
    retries: 5
```

service (frontend)

```
frontend:
  build: ./integrated-1-frontend
  container_name: frontend
  restart: unless-stopped
  networks:
    - integrated_net
  depends_on:
    - backend
```

service (backend)

```
backend:
  build: ./integrated-1-backend
  container_name: backend
  restart: unless-stopped
  networks:
    - integrated_net
  depends_on:
    - database
  environment:
    - DB_HOST=database
    - DB_PORT=3306
    - DB_NAME=integrated
    - DB_USER=dev
    - DB_PASS=Mysql@sit123
```

service (proxy)

```
proxy:
  build: ./integrated-1-proxy
  container_name: proxy
  restart: unless-stopped
  ports:
    - 80:80
  networks:
    - integrated_net
  depends_on:
    - frontend
    - backend
    - database
```

networks & volumes

```
networks:
  integrated_net:

volumes:
  myapp:
```