fastapi□□□□

```
.
├── main.py              # □□□□□
├── config.py            # □□□□
├── run.py               # □□□□
├── requirements.txt     # □□□□
├── .env                 # □□□□
├── README.md            # □□□□
├── routers/             # □□
├── models/              # □□□□
├── schemas/             # □□□□
├── repositories/        # □□□□□
├── services/            # □□□□
└── utils/               # □□□
```

□□□□□

```
# common_logger.py

import logging
import sys

__all__ = ["get_logger"]

# 1. □□□□□□
_INITIALIZED = False

# 2. □□□ + Windows □□
COLORS = {
    "DEBUG": "\033[36m",
    "INFO": "\033[32m",
    "WARNING": "\033[33m",
    "ERROR": "\033[31m",
    "CRITICAL": "\033[35m",
```

```
    "RESET": "\033[0m",
}


def _init_logger(level: int = logging.INFO):
    global _INITIALIZED
    # force=True □□□□
    if _INITIALIZED:
        return

    class ColorFormatter(logging.Formatter):
        def format(self, record):
            levelname = record.levelname
            record.levelname = (
                f"{COLORS.get(levelname, '')}{levelname}{COLORS['RESET']}"
            )
            return super().format(record)

    handler = logging.StreamHandler(sys.stdout)
    handler.setFormatter(
        ColorFormatter(
            fmt="%(asctime)s - %(name)s - %(levelname)s - %(message)s",
            datefmt="%Y-%m-%d %H:%M:%S",
        )
    )
    logging.basicConfig(level=level, handlers=[handler], force=True)
    # force=True □□□□
    _INITIALIZED = True


def get_logger(name: str, level: int = logging.INFO) -> logging.Logger:
    _init_logger(level)
    return logging.getLogger(name)
```

uvicorn □□□□□

```
```

```
#! /usr/bin/env python3
"""
FastAPI □□□□□□
"""

import uvicorn
from config import settings
from common_logger import get_logger

log = get_logger(__name__)

LOGGING_CONFIG = {
    "version": 1,
    "disable_existing_loggers": False,
    "formatters": {
        "default": {
            "fmt": "%(asctime)s - %(name)s - %(levelname)s - %(message)s",
        },
        "access": {
            "fmt": "%(asctime)s - %(name)s - %(levelname)s - %(message)s",
        },
    },
}



if __name__ == "__main__":
    log.info(f"□□ {settings.app_name} v{settings.app_version}")
    log.info(f"□□□□□: http://{settings.host}:{settings.port}")
    # print(f"API □□: http://{settings.host}:{settings.port}/docs")
    log.info("-" * 50)

    uvicorn.run(
        "main:app",
        host=settings.host,
        port=settings.port,
        reload=settings.reload,
        log_level=settings.log_level.lower(),
        log_config=LOGGING_CONFIG,
    )
```

□□□ #4
□ □□□ □□□ 13 □□ 2025 04:00:44
□ □□□ □□□ 18 □□ 2025 03:45:38