General information

This web server is a lightweight HTTP/1.1 server written in C.

Functionality

- basic HTTP protocol (methods GET, POST, DELETE)
- status codes: 200 OK, 404 - Not Found, 400 - Bad request, 403 - Forbidden, 501 - not implemented, 503 - service unavailable and return HTML pages in case of an error
- transfer big files (> 1GB size)
- hosting storage to save and load files to it.
- configurable: JSON config file provides with the following: configure server IP, port, max-clients, root directory, log-file.
- keep-alive option (can be configurable via config file)
- can handle multiple clients.
- supports logging and levels(FATAL, ERROR, WARN, INFO, DEBUG)
- covered with Unit Testing (positive, negative scenarios) using PyTest module

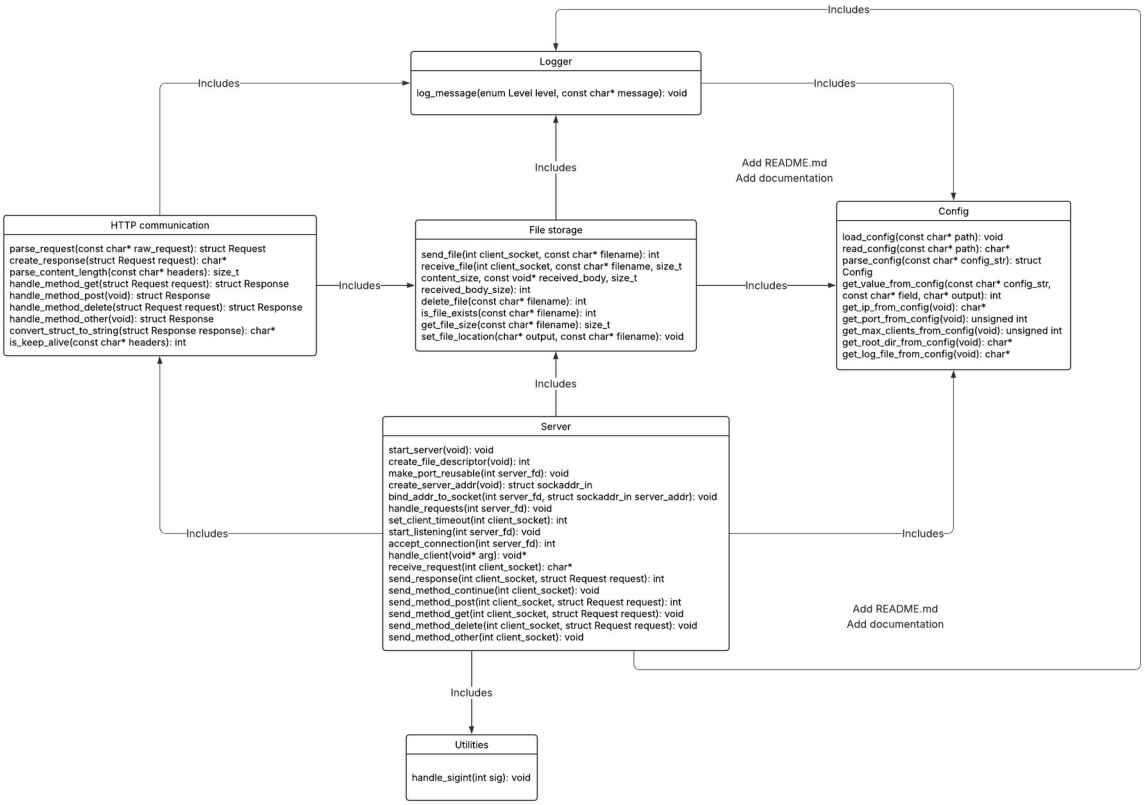
Architecture

Consists of these module:

Module name	Purpose
Config	Works with config.json: read, parse and store and use information from it.
File storage	Provides an interface for sending, receiving, deleting and checking files, as well as determining file size and resolving file paths relative to the server's configured root directory.

HTTP communication	Provides functionality for parsing HTTP requests, generating responses, and managing various HTTP methods such as GET, POST, and DELETE.
Logger	Simple logging into configured file with severity levels (FATAL, ERROR, WARN, INFO, DEBUG)
Server	Core part of Web Server, that provides functions for starting server, accept connections and handle clients.
Utilities	Provides functions, that doesn't belong to modules mentioned previously, such as safely exiting Web Server.

Module diagram:



Configuration

Web Server can be configured through using “config.json”, which has these fields:

- ip – used to set server’s IPv4 address.
- port – used to set server’s port.
- max_clients – used to set maximum number of concurrent clients.
- root_directory – used to set root storage directory of Web Server.
- log_file – used to set path and name of log file.

Default config file look:

```
{
    "ip": "127.0.0.1",
    "port": 8080,
    "max_clients": 3,
    "root_directory": "./storage",
    "log_file": "log.txt"
}
```

Note: if value of any field is not configured or configured with mistakes, Web Server will use default values.

Logging

Web Server uses logging to file, where log messages divided into 5 categories:

- DEBUG – used for debugging.
- INFO – general informational messages.
- WARN – warnings indicating potential issues.
- ERROR – error messages for failed operations.
- FATAL – critical errors that cause program exit.

Log file example:

[Sat Nov 1 18:48:37 2025] [INFO] Keep-Alive: waiting for next request on same connection
[Sat Nov 1 18:48:37 2025] [ERROR] Client disconnected or recv() error while reading headers
[Sat Nov 1 18:48:37 2025] [WARN] Client closed connection or invalid request
[Sat Nov 1 18:48:37 2025] [INFO] Client socket closed
[Sat Nov 1 18:48:37 2025] [INFO] Connection successfully accepted
[Sat Nov 1 18:48:37 2025] [INFO] Successfully set timeout time for client
[Sat Nov 1 18:48:37 2025] [ERROR] Client disconnected or recv() error while reading headers
[Sat Nov 1 18:48:37 2025] [WARN] Client closed connection or invalid request
[Sat Nov 1 18:48:37 2025] [ERROR] Client disconnected or recv() error while reading headers
[Sat Nov 1 18:48:37 2025] [WARN] Client closed connection or invalid request
[Sat Nov 1 18:48:37 2025] [INFO] Client socket closed