

GitHub 开源推荐 | 一个轻量级、高性能的 C++ Web 框架

编程学习基地 今天

以下文章来源于高效程序员，作者Waleon



高效程序员

聚焦程序人生，践行终身成长。专注分享 IT 技术「C++/Python/Linux/Qt 等」、学习资料、职场经验、热点资讯，有趣、好玩、靠谱！ ...



点击蓝字 ··· 关注我们

Github 开源推荐

专注分享 GitHub 上有趣、好玩的开源项目，帮助大家提高编程技巧，找到编程乐趣。

说起 Web 开发，大多数人会想到 Java、Python、Golang ... 因为它们的主流 Web 框架有很多，Java 有非常知名的 Spring 全家桶，Python 有大而全的 Django、小而精的 Flask、高性能的 Tornado，Golang 也有快速灵活的 Gin、Echo 等框架。

相比之下，C/C++ 阵营则逊色不少。之前，我曾写过一篇《C++ Web (HTTP) 开发 10 大利器》，其中介绍了一些 C/C++ Web 框架。有一个名为 Oat++ 的很不错，轻量、跨平台、高性能、完全零依赖，非常值得学习！



Oat++

An Open Source C++ Web Framework

Get Started

Version: 1.2.5

License: Apache-2.0 License

高程序员



Oat++ 介绍

要深入了解 Oat++，离不开这几个网址：

- Oat++ 主页: <https://oatpp.io>
- Oat++ 文档: <https://oatpp.io/docs/start>
- GitHub 地址: <https://github.com/oatpp/oatpp>

其主要特性有：

- 随处运行

Oat++ 没有任何依赖性，可以很容易地移植到各种支持的平台上（Linux、MacOS、Windows）。

- 构建健壮的api

使用 Oat++ Simple-API，构建灵活而健壮的 API 既简单又有趣。

- 处理 500 万个连接

使用 Oat++ Async-API，可以在单个服务器上处理超过 500 万个并发连接。

- 访问数据库

Oat++ ORM 提供了一种简单而统一的方式来访问数据库

- 保持代码一致

Oat++ 在整个代码库中依靠对象映射来确保 API 和数据模型的一致性

- 生成 API 文档

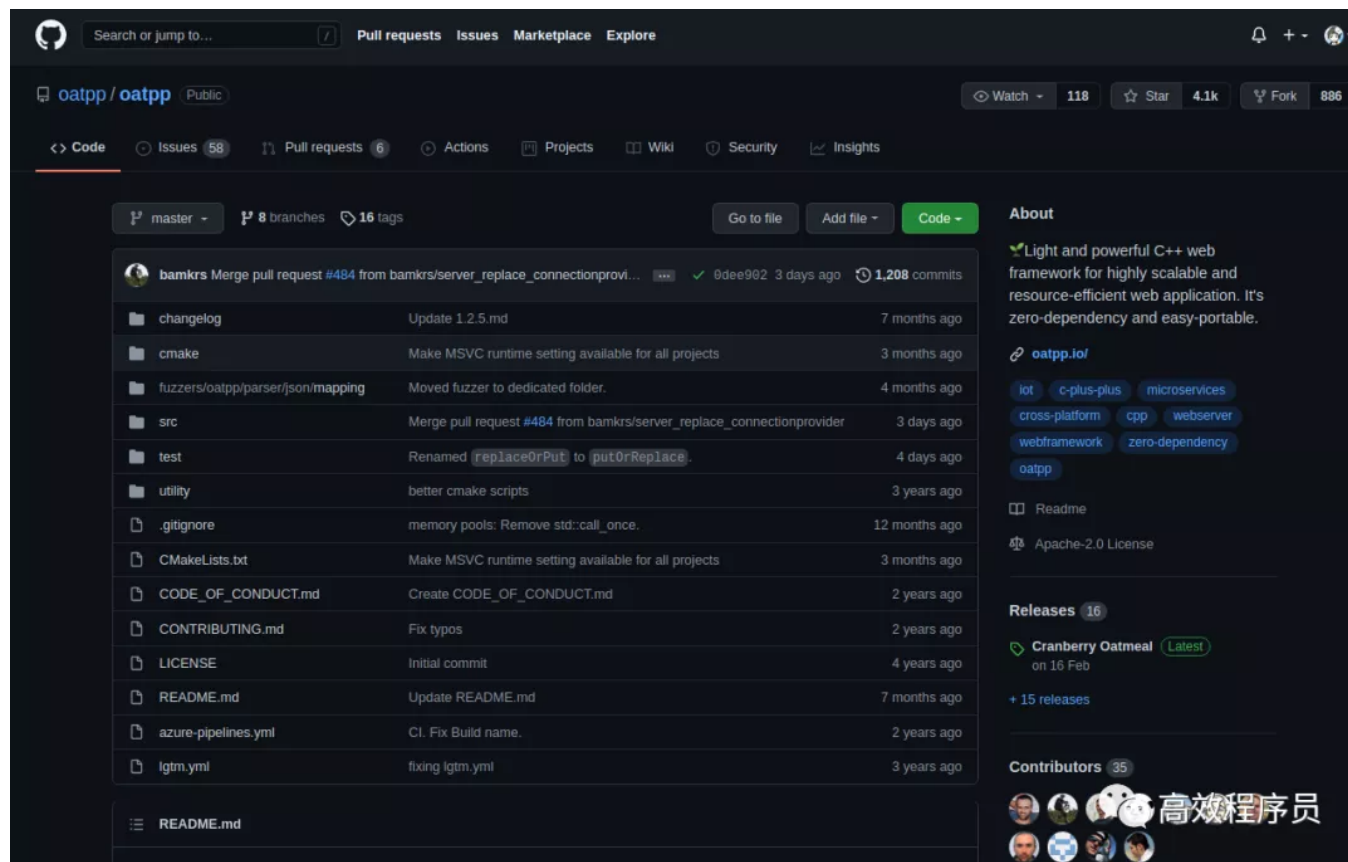
使用 Swagger-UI 和 OpenAPI 3.0.0 自动记录 endpoints

最吸引我的是 HTTP/HTTPS、文件上传/下载、以及强大的 Swagger API 功能。



编译 Oat++

进入 Oat++ 的 GitHub 页面，你会发现 Star 多达 4K+，贡献者有 30 多个，且最近几天还有代码提交，所以 **不用担心热度和活跃度，这个框架一直有人在积极地维护。**



环境要求

Oat++ 的编译过程很简单，只需要有基本的开发环境就行了：

- Git

- 编译器支持的 C++ 版本 ≥ 11
- Make
- CMake 版本 ≥ 3.1

如果没有的话，按照下述步骤安装，以 Ubuntu 为例：

```
$ sudo apt install git
$ sudo apt install cmake
$ sudo apt install build-essential
```

编译安装

下载 Oat++ 源码：

```
$ git clone https://github.com/oatpp/oatpp.git
```

随后，执行编译安装四部曲：

```
$ cd oatpp/
$ mkdir build && cd build
$ cmake ..
$ sudo make && sudo make install
```

示例程序

为了演示 Oat++，我们从最简单的“Hello, World!”开始！

创建一个 CMake 项目，CMakeLists.txt 配置如下：

```
cmake_minimum_required(VERSION 3.1)
project(helloworld)

set(CMAKE_CXX_STANDARD 11)
set(SOURCE_FILES main.cpp handler.h)

# 查找 oatpp 依赖
find_package(oatpp REQUIRED)

add_executable(${PROJECT_NAME} ${SOURCE_FILES})

# 将目标文件与库文件进行链接
target_link_libraries(${PROJECT_NAME} oatpp::oatpp)
```

默认情况下，Oat++ 会对客户端请求响应 404，除此之外什么都不做。

若要添加自定义响应，必须实现 `HttpRequestHandler`：

```
// handler.h
#ifndef HANDLER_H
#define HANDLER_H

#include "oatpp/web/server/HttpRequestHandler.hpp"

#define O_UNUSED(x) (void)x;
```

```
// 自定义请求处理程序
class Handler : public oatpp::web::server::HttpRequestHandler
{
public:
    // 处理传入的请求，并返回响应
    std::shared_ptr<OutgoingResponse> handle(const std::shared_ptr<IncomingRequest>& request) override {
        O_UNUSED(request);

        return ResponseFactory::createResponse(Status::CODE_200, "Hello, World!");
    }
};

#endif // HANDLER_H
```

有了处理程序之后，需要通过 Router 将请求路由到它：

```
// main.cpp
#include "oatpp/web/server/HttpConnectionHandler.hpp"
#include "oatpp/network/tcp/server/ConnectionProvider.hpp"
#include "oatpp/network/Server.hpp"
#include "handler.h"

void run()
{
    // 为 HTTP 请求创建路由器
    auto router = oatpp::web::server::HttpRouter::createShared();

    // 路由 GET - "/hello" 请求到处理程序
    router->route("GET", "/hello", std::make_shared<Handler>());

    // 创建 HTTP 连接处理程序
    auto connectionHandler = oatpp::web::server::HttpConnectionHandler::createShared(router);

    // 创建 TCP 连接提供者
    auto connectionProvider = oatpp::network::tcp::server::ConnectionProvider::createShared({"localhost", 8000, oatpp::network::tcp::server::ConnectionProvider::createShared()});

    // 创建服务器，它接受提供的 TCP 连接并将其传递给 HTTP 连接处理程序
    auto server = oatpp::network::Server::createShared(connectionProvider, connectionHandler);
    server->start();
}
```

```
oatpp::network::Server server(connectionProvider, connectionHandler);

// 打印服务器端口
OATPP_LOGI("MyApp", "Server running on port %s", connectionProvider->getProperty("port").getData());

// 运行服务器
server.run();
}

int main()
{
    // 初始化 oatpp 环境
    oatpp::base::Environment::init();

    // 运行应用
    run();

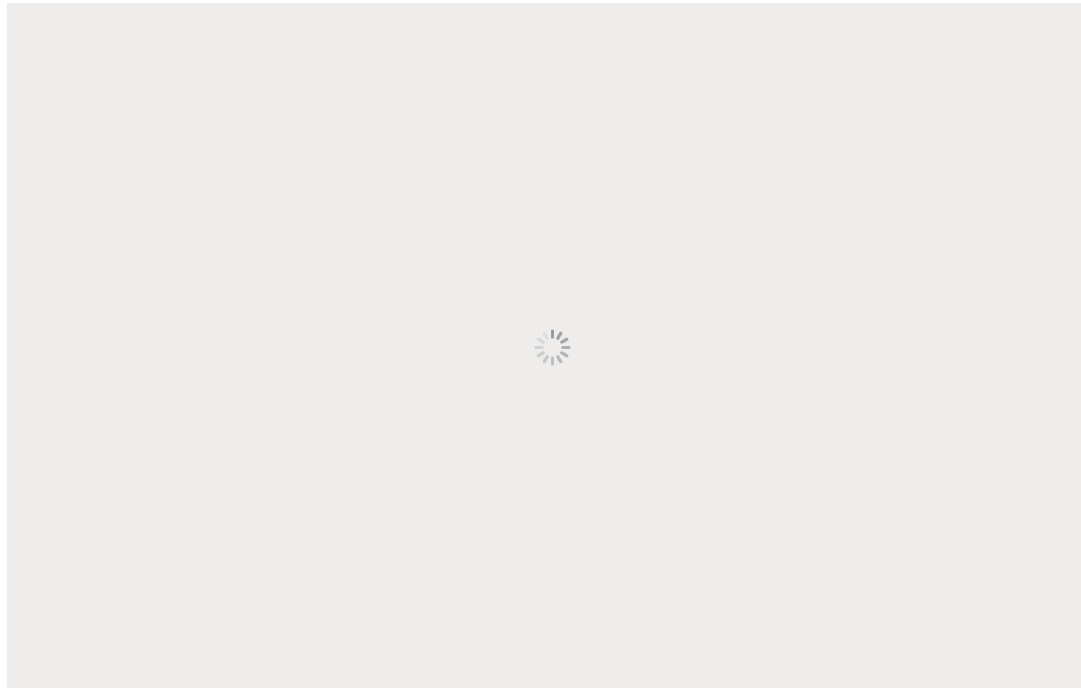
    // 销毁 oatpp 环境
    oatpp::base::Environment::destroy();

    return 0;
}
```



请求验证

运行程序，在浏览器中访问 <http://localhost:8000/hello>，就会显示“Hello, World!”信息了：

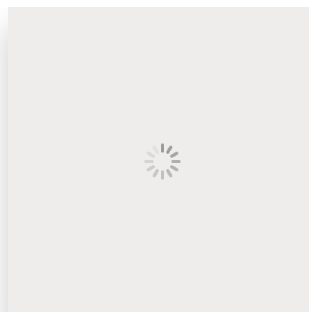


或者使用 curl 请求 <http://127.0.0.1:8000/hello>，效果一样：

```
$ curl http://127.0.0.1:8000/hello
Hello, World!
```

.....END.....

你们的在看就是对我最大的肯定，
点个在看好吗~



编程学习基地

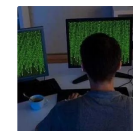
常回基地看看

[阅读原文](#)

喜欢此内容的人还喜欢

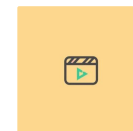
C/C++ 和其他语言的最大区别是什么？

C语言Plus



PyTorch vs LibTorch：网络推理速度谁更快？

程序员大白



推荐两个项目！

Coding这件小事

