**Содержание**

[Задание 3](#_Toc57920040)

[Справочная информация 4](#_Toc57920041)

[Текст программы на языке С++ 5](#_Toc57920043)

[Файл main.cpp 5](#_Toc57920044)

[Файл MyController.hpp 7](#_Toc57920045)

[Файл MyController.cpp 10](#_Toc57920045)

[Файл SwaggerComponent.hpp 11](#_Toc57920046)

[Файл CountryDto.hpp 14](#_Toc57920046)

[Файл index.html 15](#_Toc57920046)

[Разработка интерфейса пользователя 18](#_Toc57920047)

[Список литературы 28](#_Toc57920048)

# **Задание**

Необходимо разработать приложение для работы со справочником по странам с использованием фреймворка Oat++.

В приложении должен быть реализован хотя бы один API контроллер, который включает в себя методы для работы со справочником (получить все записи, получить одну запись по id, добавить новую запись, отредактировать запись, удалить запись).

Записи должны храниться в БД SQLite (или PostgreSQL). База данных должна создаваться с помощью миграции (миграция выполняется с помощью соответствующего инструмента фреймворка Oat++). При создании базы данных необходимо описать таблицу, в которой будут храниться записи справочника, который выдан в задании.

В приложении должна присутствовать система авторизации. Необходимо завести несколько пользователей (можно усложнить задачу и создать отдельную таблицу в БД, а также предусмотреть отдельный API контроллер и сервис; можно этого не делать и обойтись условными операторами).

Реализовать возможность определенным пользователям только просматривать записи справочника (запрос всех записей и запрос одной записи), а другим пользователям – управлять записями и просматривать (дополнительно создание, обновление и удаление).

Все запросы должны быть задокументированы с использованием Swagger UI.

# **Справочная информация**

1. **Swagger** – это язык описания интерфейса для описания RESTful API, выраженных с помощью JSON. Swagger используется вместе с набором программных средств с открытым исходным кодом для проектирования, создания, документирования и использования веб-служб RESTful.
2. **API** (Application Programming Interface) – описание способов взаимодействия одной компьютерной программы с другими. Обычно входит в описание какого-либо интернет-протокола (например, SCIM), программного каркаса (фреймворка) или стандарта вызовов функций операционной системы.
3. **База данных** – совокупность данных, хранимых в соответствии со схемой данных, манипулирование которыми выполняют в соответствии с правилами средств моделирования данных.
4. **PostgreSQL** –свободная объектно-реляционная система управления базами данных. Существует в реализациях для множества UNIX-подобных платформ, включая AIX, различные BSD-системы, HP-UX, IRIX, Linux, macOS, Solaris/OpenSolaris, Tru64, QNX, а также для Microsoft Windows.
5. **Объект Передачи Данных (DTO)** – шаблон проектирования, используемый для передачи данных между программными системами или уровнями в программной архитектуре. Это простой объект содержащий набор данных, обычно в форме свойств или полей, которые представляют определённый объект или концепцию в системе.
6. **Oat ++** – это современный веб-фреймворк для C ++. Он полностью загружен и содержит все необходимые компоненты для эффективной разработки на производственном уровне. Он также легкий и имеет небольшой объем памяти.

# **Текст программы на языке С++**

**main.cpp**

#include "oatpp/network/Server.hpp"

#include "oatpp/web/server/api/ApiController.hpp"

#include "oatpp/parser/json/mapping/ObjectMapper.hpp"

#include "oatpp-swagger/Controller.hpp"

#include "MyController.hpp"

#include "SwaggerComponent.hpp"

void run() {

oatpp::base::Environment::init();

auto objectMapper = oatpp::parser::json::mapping::ObjectMapper::createShared();

auto router = oatpp::web::server::HttpRouter::createShared();

auto myController = std::makeshared<MyController>(objectMapper);

myController->addEndpointsToRouter(router);

auto swaggerComponent = std::make\_shared<SwaggerComponent>();

oatpp::swagger::Controller::addEndpointsToRouter(router);

oatpp::network::Address address("tcp://0.0.0.0:8000");

auto server = oatpp::network::Server::createShared(router, 1);

pqxx::connection conn("dbname=mydb user=postgres password=1234 hostaddr=127.0.0.1 port=5432");

server->run();

oatpp::base::Environment::destroy();

}

**MyController.hpp**

#pragma once

#include "oatpp/web/server/api/ApiController.hpp"

#include "oatpp/core/macro/codegen.hpp"

#include "MyController.hpp"

#include <libpq-fe.h>

#include <pqxx/pqxx>

#include OATPP\_CODEGEN\_BEGIN(ApiController)

class MyController : public oatpp::web::server::api::ApiController {

public:

MyController(const std::shared\_ptr<ObjectMapper>& objectMapper)

: oatpp::web::server::api::ApiController(objectMapper)

{}

ENDPOINT("GET", "/", root) {

return createResponse(Status::CODE\_200, "text/html", getRootHtml());

}

ENDPOINT("POST", "/country", createCountry,

BODY\_STRING(String, body)) {

auto dto = MyDto::createShared();

oatpp::parser::json::mapping::ObjectMapper::parseJson(body, dto);

pqxx::work txn(conn);

pqxx::result res = txn.exec(

"INSERT INTO yname (name, capital, population, gdp, leader, currency) "

"VALUES ($1, $2, $3, $4, $5, $6)",

dto->getCountry(),

dto->getCapital(),

dto->getPopulation(),

dto->getGdp(),

dto->getLeader(),

dto->getCurrency()

);

txn.commit();

return createResponse(Status::CODE\_200, "text/plain", "Data saved to PostgreSQL");

}

ENDPOINT("GET", "/countries", getCountries) {

try {

pqxx::work txn(conn);

pqxx::result res = txn.exec("SELECT name, capital, population, gdp, leader, currency FROM uname");

oatpp::data::stream::ChunkedBuffer buffer;

for (auto row : res) {

buffer.write("");

buffer.write("Страна: " + row[0].c\_str());

buffer.write("");

buffer.write("Столица: " + row[1].c\_str());

buffer.write("");

buffer.write("Население: " + std::to\_string(row[2].as<int>()));

buffer.write("");

buffer.write("ВВП: " + std::to\_string(row[3].as<double>()));

buffer.write("");

buffer.write("Лидер: " + row[4].c\_str());

buffer.write("");

buffer.write("Валюта: " + row[5].c\_str());

buffer.write("");

}

}

catch (err) {

return createResponse(Status::CODE\_200, "Таблицы не существует", buffer.toString());

}

}

ENDPOINT("GET", "/edit", getCountries) {

try {

pqxx::work txn(conn);

pqxx::result res = txn.exec("UPDATE uname SET type = 'edit' WHERE name = 'country'");

oatpp::data::stream::ChunkedBuffer buffer;

}

ENDPOINT("GET", "/delete", getCountries) {

pqxx::work txn(conn);

pqxx::result res = txn.exec("DELETE FROM uname WHERE name = 'country'");

oatpp::data::stream::ChunkedBuffer buffer;

}

private:

static pqxx::connection conn;

oatpp::String getRootHtml() {

std::ifstream inFile("index.html");

std::stringstream strStream;

strStream << inFile.rdbuf();

return strStream.str().c\_str();

}

};

#include OATPP\_CODEGEN\_END(ApiController)

# **MyController.cpp**

# #include "MyController.hpp"

# #include "oatpp/web/server/api/ApiController.hpp"

# #include "oatpp/core/macro/codegen.hpp"

# #include "MyController.hpp"

# #include <libpq-fe.h>

# #include <pqxx/pqxx>

# pqxx::connection MyController::conn("dbname=oatpp user=postgres password='' hostaddr=127.0.0.1 port=5432");

# **SwaggerComponent.hpp**

# #pragma once

# #include "oatpp-swagger/Controller.hpp"

# #include <iostream>

# class SwaggerComponent {

# public:

# std::sharedptr; <oatpp::swagger::Document> getSwaggerDocument() {

# OATPPLOGD("SwaggerComponent", "getSwaggerDocument");

# auto document = oatpp::swagger::Document::createShared();

# document->info.title = "My API";

# document->info.description = "API for creating country info form";

# document->info.version = "1.0";

# auto myController = std::makeshared<MyController>();

# auto endpoints = myController->getEndpoints();

# for (auto& endpoint : endpoints) {

# auto operation = oatpp::swagger::Controller::createSwaggerOperation(endpoint.method, endpoint.path, endpoint.description);

# for (auto& header : endpoint.headers) {

# operation->addResponseHeader(header.name, header.description);

# }

# for (auto& tag : endpoint.tags) {

# operation->addTag(tag);

# }

# for (auto& param : endpoint.params) {

# auto swaggerParam = oatpp::swagger::Controller::createSwaggerParameter(param.in,

# param.name,

# oatpp::swagger::StringToSchema(param.type),

# param.description,

# param.required);

# operation->addParameter(swaggerParam);

# }

# auto responses = myController->getEndpointResponses(endpoint.method, endpoint.path);

# for (auto& response : responses) {

# operation->addResponse(response.statusCode, oatpp::swagger::StringToSchema(response.responseType), response.description);

# }

# document->addOperation(endpoint.path, operation);

# }

# ; return document;

# }

# };

**CountryDto.hpp**

#pragma once

#include "oatpp/core/Types.hpp"

#include "oatpp/core/macro/codegen.hpp"

#include OATPP\_CODEGEN\_BEGIN(DTO)

class CountryDto : public oatpp::DTO {

DTO\_INIT(CountryDto, DTO)

DTO\_FIELD(String, name);

DTO\_FIELD(String, capital);

DTO\_FIELD(Int32, population);

DTO\_FIELD(Int32, gdp);

DTO\_FIELD(String, leader);

DTO\_FIELD(String, currency);

};

#include OATPP\_CODEGEN\_END(DTO)

**Index.html**

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>Country Form</title>

</head>

<body>

<form action="/country" method="post">

<label for="yname">Твое имя:</label>

<input type="text" id="yname" name="yname"><br>

<label for="name">Название страны:</label>

<input type="text" id="name" name="name"><br>

<label for="capital">Столица:</label>

<input type="text" id="capital" name="capital"><br>

<label for="population">Население:</label>

<input type="text" id="population" name="population"><br>

<label for="gdp">ВВП:</label>

<input type="text" id="gdp" name="gdp"><br>

<label for="leader">Лидер:</label>

<input type="text" id="leader" name="leader"><br>

<label for="currency">Валюта:</label>

<input type="text" id="currency" name="currency"><br>

<button type="submit">Отправить</button>

</form>

<form action="/countries" method="post">

<label for="uname">Твое имя:</label>

<input type="text" id="uname" name="uname"><br>

<button type="submit">Посмотреть страны</button>

</form>

<form action="/id" method="post">

<label for="uname">Твое имя:</label>

<input type="text" id="uname" name="uname"><br>

<label for="country">ID:</label>

<input type="text" id="country" name="country"><br>

<button type="submit">Посмотреть страну</button>

</form>

<form action="/edit" method="post">

<label for="uname">Твое имя:</label>

<input type="text" id="uname" name="uname"><br>

<label for="country">Название страны:</label>

<input type="text" id="country" name="country"><br>

<label for="type">Какой параметр:</label>

<select id="type" name="type">

<option value="name">Название</option>

<option value="capital">Столица</option>

<option value="population">Население</option>

<option value="gdp">ВВП</option>

<option value="leader">Лидер</option>

<option value="currency">Валюта</option>

</select>

<br>

<label for="edit">Значение:</label>

<input type="text" id="edit" name="edit"><br>

<button type="submit">Редактировать страну</button>

</form>

<form action="/delete" method="post">

<label for="uname">Твое имя:</label>

<input type="text" id="uname" name="uname"><br>

<label for="country">Название страны:</label>

<input type="text" id="country" name="country"><br>

<button type="submit">Удалить страну</button>

</form>

</body>

</html>

# **Разработка интерфейса пользователя**

На рисунке 1 представлено основное меню программы. Управление осуществляется с помощью ввода данных в окошки и нажатия на кнопки. Так же на нем представлен пример заполнения информации о стране на имя «Раевских Максим». Путь <http://localhost:8000>. При нажатии на кнопку «Отправить» информация о стране будет отправлена в базу данных на PostgreSQL.

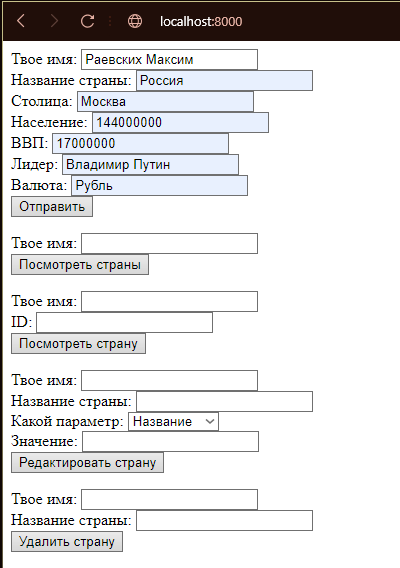


Рисунок 1 ­­­– Основное меню программы

На рисунке 2 представлен пример заполнения формы для просмотра существующей информации о странах. При нажатии кнопки «Посмотреть страны» происходит переход на другую страницу (рисунок 3).

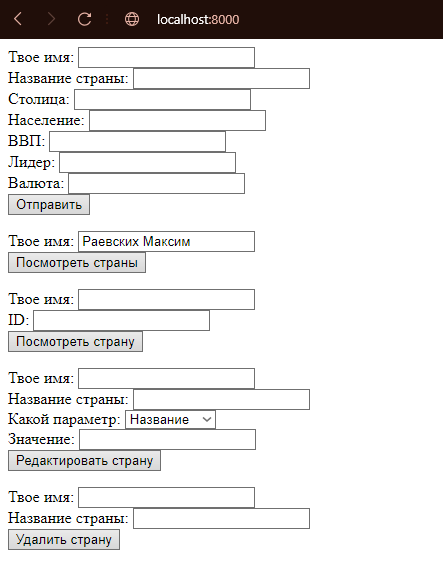


Рисунок 2 ­­­– Пример заполнения формы для просмотра существующей информации о странах.

На рисунке 3 представлен вывод всей информации о странах на имя «Раевских Максим». Путь http://localhost:8000/countries.

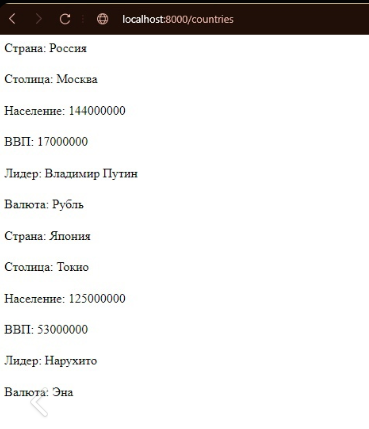


Рисунок 3 ­­­– Вывод всей информации о странах.

На рисунке 4 представлен пример заполнения формы для просмотра информации о стране по ее id. При нажатии на кнопку «Посмотреть страну» выводит информации о стране по ее id (рисунок 5).

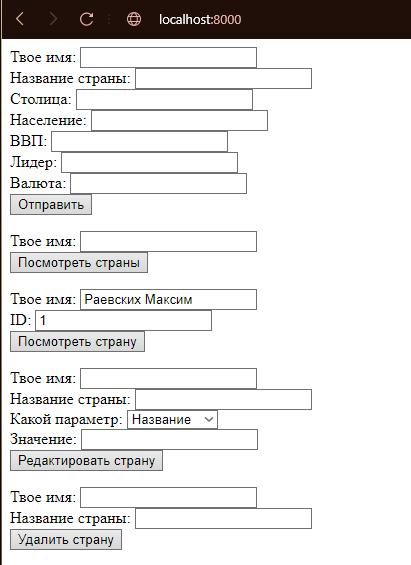


Рисунок 4 ­­­– Пример заполнения формы для просмотра информации по стране по ее id.

На рисунке 5 представлена страница с выводом информации по стране с id равном «1» на имя «Раевских Максим». Путь <http://localhost:8000/id>.

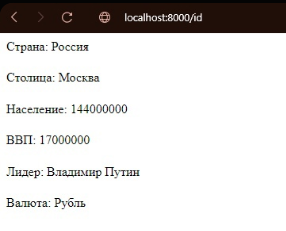


Рисунок 5 ­­­– Вывод информации о стране по ее id.

На рисунке 6 представлен пример заполнения формы на изменение данных о существующей стране. При нажатии на кнопку «Редактировать страну» изменяет данные о стране (рисунок 7)

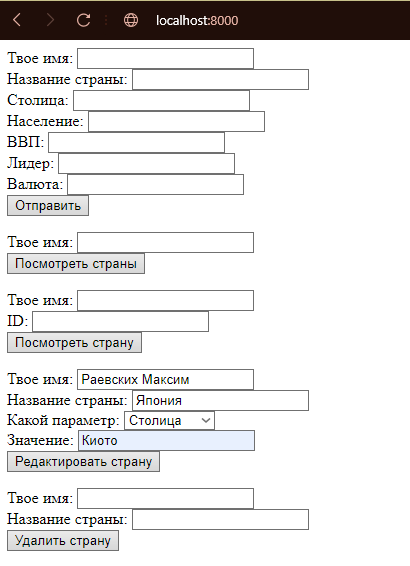


Рисунок 6 – Пример заполнения формы на изменение параметров страны.

На рисунке 7 представлено изменение столицы на «Киото» страны «Япония» на имя «Раевских Максим».

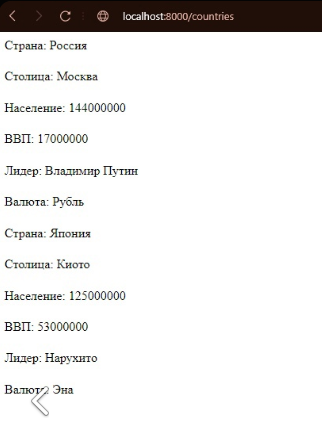


Рисунок 7 – Изменение столицы Японии.

На рисунке 8 представлен пример заполнения формы на удаление страны из списка для пользователя. При нажатии на кнопку «Удалить страну» удаляет страну из списка для пользователя (рисунок 9).

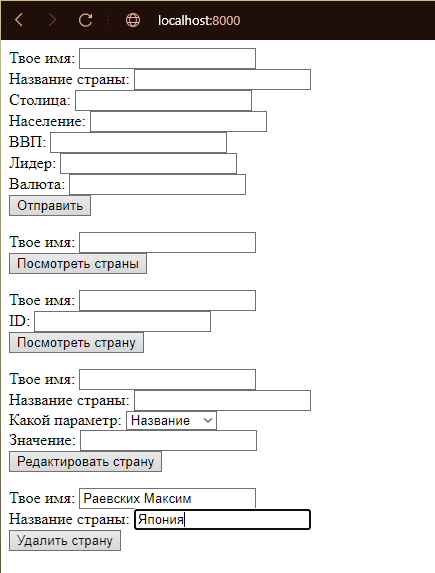


Рисунок 8 – Пример заполнения формы на удаление страны.

На рисунке 9 представлен пример удаления страны «Япония» для пользователя «Раевских Максим».

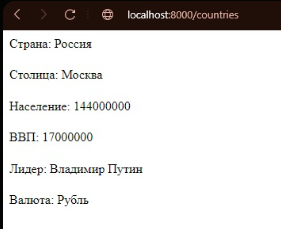


Рисунок 9 – Пример удаления «Японии».

На рисунках 10 и 11 представлено доказательство работы базы данных PostgreSQL.

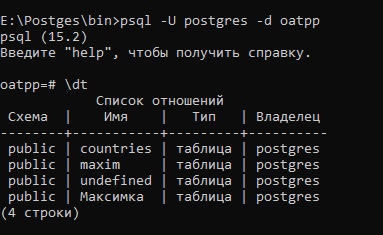


Рисунок 10 – Существующие отношения в базе данных.

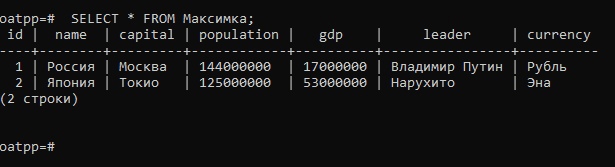


Рисунок 11 – Существующие столбцы в базе данных на имя «Раевских Максим».

На рисунке 12 представлена страница Swagger UI.

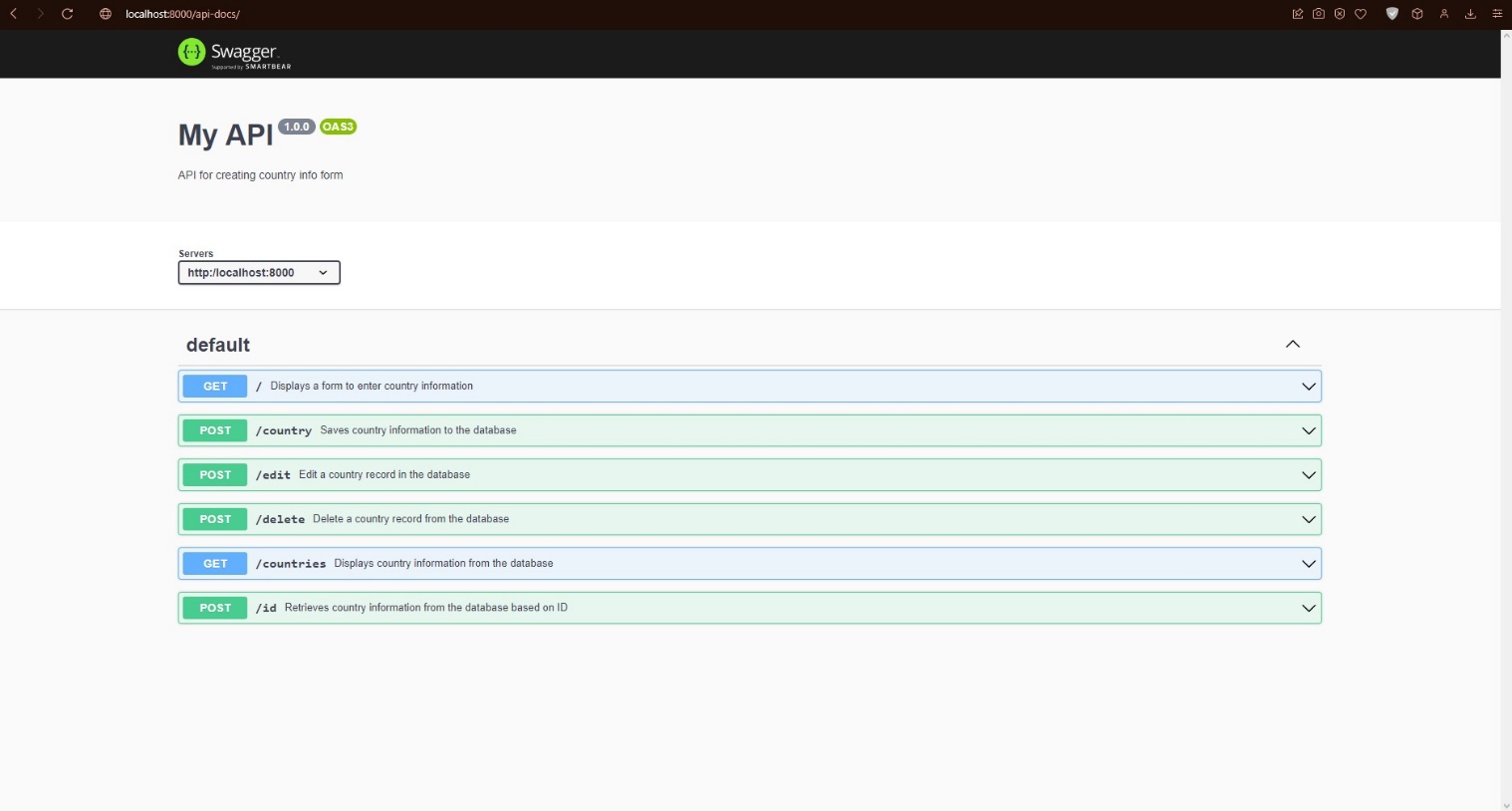


Рисунок 12 – Swagger.

# **Список литературы**

1. Введение в C/C++ программирование консоли - Доля П.Г. Харьковский Национальный Университет механико-математический факультет, 2015 г.
2. Программирование на языке СИ. Методические указания к лабораторным работам – Шафеева О.П. Государственное образовательное учреждение высшего профессионального образования «Омский государственный технический университет», 2008 г.