|  |  |
| --- | --- |
|  | ООО «Круиз.онлайн»  ИНН 6315008371  КПП 631501001 |

Программа для ЭВМ «Kruiz.Online Docking Admin»

Информация, необходимая для установки и эксплуатации программы

Листов 34

2025

**Оглавление**

[1 Установка и запуск Системы 3](#_Toc201847623)

[1.1 Установка браузера 3](#_Toc201847624)

[Установка Google Chrome 4](#_Toc201847625)

[Установка Яндекс Браузера 5](#_Toc201847626)

[1.2 Установка Docker Desktop на операционную систему Windows 6](#_Toc201847627)

[Шаг 1: Включение WSL2 6](#_Toc201847628)

[Шаг 2: Загрузка и установка Docker Desktop 6](#_Toc201847629)

[Шаг 3: Запуск KODA Admin Panel через Docker Compose 7](#_Toc201847630)

[1.3 Вход в систему 9](#_Toc201847631)

[1.4 Требования к рабочему месту пользователя 9](#_Toc201847632)

[1.5 Возможные проблемы и способы их устранения 9](#_Toc201847633)

[1.6 Вспомогательные материалы 10](#_Toc201847634)

[2 Описание функциональных разделов 10](#_Toc201847635)

[2.1 Общие сведения о функциональных возможностях 11](#_Toc201847636)

[2.2 Функциональный раздел «Пользователи» 11](#_Toc201847637)

[Назначение 12](#_Toc201847638)

[Возможности 12](#_Toc201847639)

[Уровни доступа по ролям 13](#_Toc201847640)

[2.3 Функциональный раздел «Круизы» 14](#_Toc201847641)

[Назначение 14](#_Toc201847642)

[Возможности 14](#_Toc201847643)

[Пример данных: 15](#_Toc201847644)

[Уровни доступа по ролям 15](#_Toc201847645)

[2.4 Функциональный раздел «Теплоходы» 16](#_Toc201847646)

[Назначение 16](#_Toc201847647)

[Возможности 16](#_Toc201847648)

[Пример данных 17](#_Toc201847649)

[Уровни доступа по ролям 18](#_Toc201847650)

[2.5 Функциональный раздел «Отчеты» 18](#_Toc201847651)

[Назначение 18](#_Toc201847652)

[Возможности 19](#_Toc201847653)

[Пример отчета 20](#_Toc201847654)

[Уровни доступа по ролям 20](#_Toc201847655)

[2.6 Дополнительные возможности 20](#_Toc201847656)

[2.6.1 Авторизация и управление ролями 20](#_Toc201847657)

[2.6.2 Интеграция с REST API 21](#_Toc201847658)

[2.6.3 Поддержка JSON-формата 21](#_Toc201847659)

[2.6.4 Коды ошибок 21](#_Toc201847660)

[3 Развертывание на сервере 22](#_Toc201847661)

[3.1 Общие положения 22](#_Toc201847662)

[3.2 Требования к серверу 22](#_Toc201847663)

[3.3 Подготовка сервера 23](#_Toc201847664)

[Шаг 1: Обновление системы 23](#_Toc201847665)

[Шаг 2: Установка Docker 23](#_Toc201847666)

[3.4 Настройка домена и SSL-сертификата 23](#_Toc201847667)

[Привязка домена 23](#_Toc201847668)

[3.5 Развертывание приложения 23](#_Toc201847669)

[Шаг 1: Получение репозитория 23](#_Toc201847670)

[Шаг 2: Запуск контейнеров 25](#_Toc201847671)

[3.6 Открытие портов 25](#_Toc201847672)

[3.7 Бэкапы и восстановление данных 25](#_Toc201847673)

[3.8 Мониторинг состояния системы 25](#_Toc201847674)

[3.9 Возможные проблемы и способы их устранения 26](#_Toc201847675)

[3.10 Вспомогательные материалы 26](#_Toc201847676)

[4 Лицензирование и юридические аспекты 27](#_Toc201847677)

[4.1 Общие положения 27](#_Toc201847678)

[4.2 Использованные open-source технологии и их лицензии 27](#_Toc201847679)

[4.3 Лицензия на программное обеспечение KODA Admin Panel 28](#_Toc201847680)

[5 Описание REST API 29](#_Toc201847681)

[5.1 Общая информация 29](#_Toc201847682)

[5.2 Методы и эндпоинты 29](#_Toc201847683)

[5.3 Формат ответов 29](#_Toc201847684)

[5.4 Коды ошибок 30](#_Toc201847685)

[5.5 Параметры запросов 30](#_Toc201847686)

[5.6 Примеры использования 33](#_Toc201847687)

[5.7 Ограничения 34](#_Toc201847688)

# **1 Установка и запуск Системы**

## **Установка браузера**

Для корректной работы с системой необходимо использовать один из следующих современных браузеров:

* Google Chrome версии 118 и выше.
* Яндекс Браузер версии 23 и выше.

Эти браузеры обеспечивают полную совместимость с пользовательским интерфейсом административной панели, а также корректное отображение всех элементов веб-интерфейса, включая формы ввода, таблицы, модальные окна и графические элементы.

### **Установка Google Chrome**

1. Откройте любой браузер на вашем компьютере.
2. Перейдите по ссылке: <https://www.google.com/intl/ru_ru/chrome/>

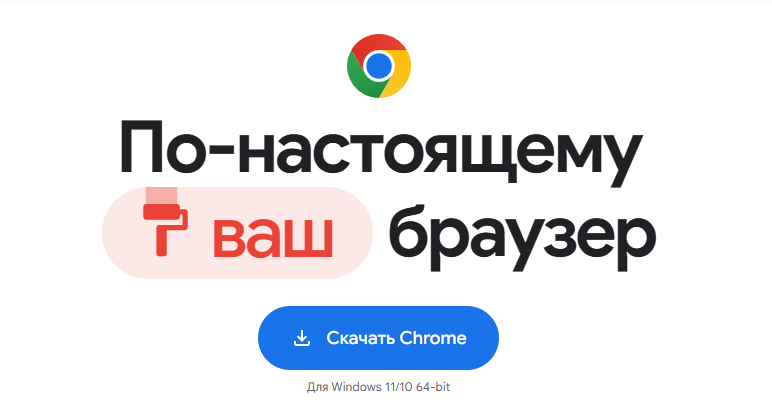


Рисунок 1 – Страница скачивания браузера.

1. Нажмите кнопку «Скачать Chrome».
2. В открывшемся окне выберите необходимую версию установщика (обычно это .exe файл для Windows).
3. После загрузки файла запустите его двойным кликом.
4. Следуйте указаниям мастера установки:

* Примите лицензионное соглашение.
* Выберите папку установки (по умолчанию).
* Установите галочку, если хотите сделать Chrome браузером по умолчанию.
* Нажмите «Установить».

1. По завершении установки нажмите «Запустить».
2. Для обновления браузера до актуальной версии:

* Откройте меню Chrome → Настройки → О программе Google Chrome.
* При наличии обновления оно будет установлено автоматически.

Рекомендуется не отключать автоматическое обновление браузера, чтобы обеспечить максимальную безопасность и стабильность работы.

### **Установка Яндекс Браузера**

1. Откройте любой браузер на вашем компьютере.
2. Перейдите по ссылке: <https://browser.yandex.ru/>

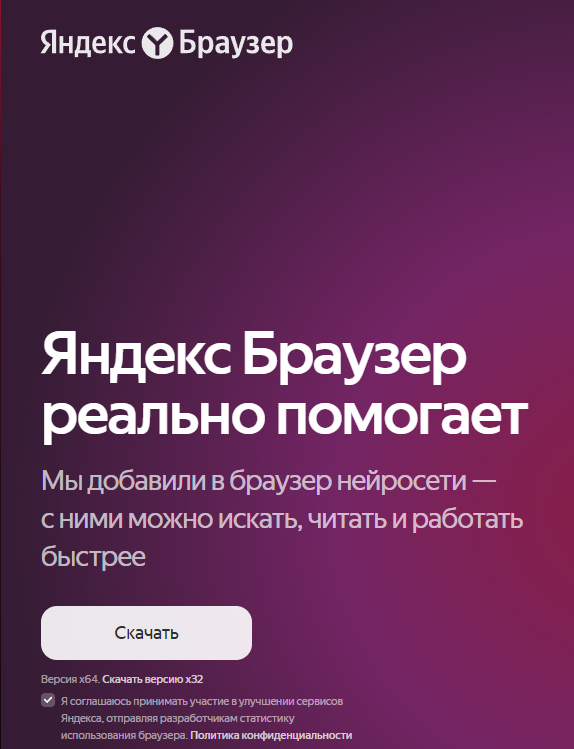


Рисунок 2 – Страница скачивания браузера.

1. Нажмите кнопку «Скачать».
2. В открывшемся окне выберите необходимую версию установщика (обычно это .exe файл для Windows).
3. После загрузки файла запустите его двойным кликом.
4. Следуйте указаниям мастера установки:

* Примите лицензионное соглашение.
* Выберите папку установки (по умолчанию).
* Установите галочки, если хотите сделать Яндекс Браузер браузером по умолчанию и добавить его на рабочий стол.
* Нажмите «Установить».

1. По завершении установки нажмите «Запустить».
2. Для обновления браузера:

* Откройте меню Яндекс Браузера → Настройки → О браузере.
* При наличии обновления оно будет установлено автоматически.

Яндекс Браузер имеет встроенную защиту от вредоносных сайтов и рекламы, что повышает уровень безопасности при работе с административной панелью.

## **Установка Docker Desktop на операционную систему Windows**

Docker является ключевым компонентом для локального развертывания KODA Admin Panel, поскольку система поставляется в виде контейнеров, которые можно запускать на различных платформах.

⚠️ Важно: перед установкой Docker необходимо активировать Windows Subsystem for Linux (WSL), так как Docker Desktop для Windows использует WSL2 для запуска контейнеров.

### **Шаг 1: Включение WSL2**

1. Откройте PowerShell от имени администратора.
2. Введите следующую команду и нажмите Enter:

«wsl –install»

1. Дождитесь завершения установки WSL и перезагрузите компьютер.

### **Шаг 2: Загрузка и установка Docker Desktop**

1. Перейдите на официальный сайт Docker: <https://www.docker.com/products/docker-desktop/>

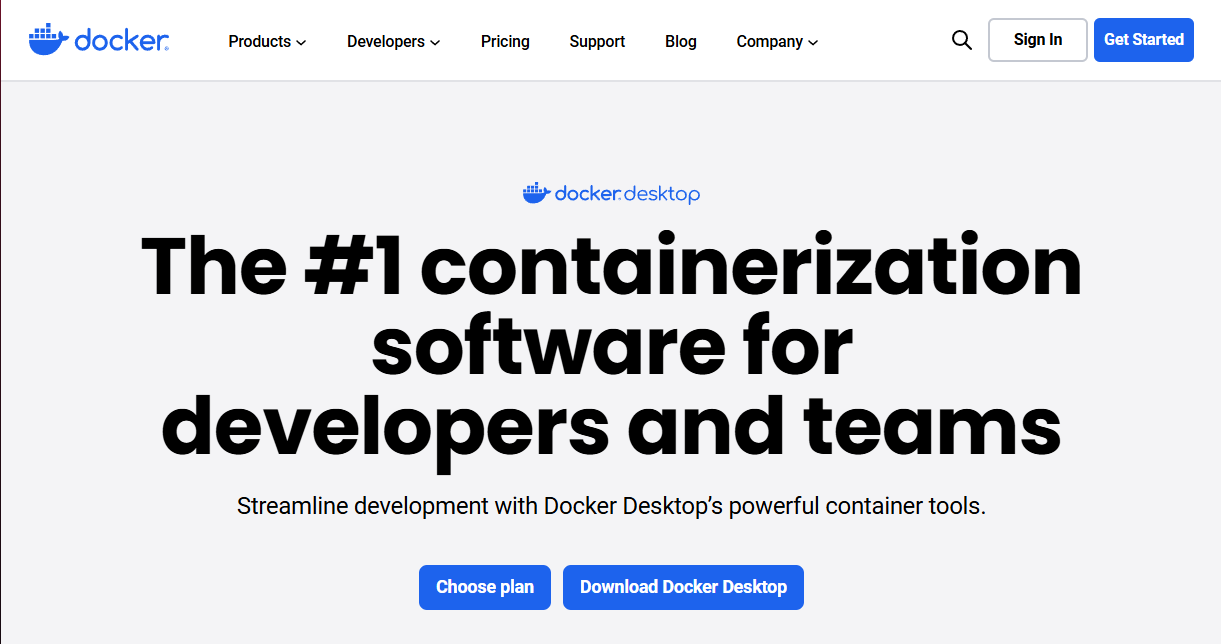


Рисунок 3 – Страница скачивания Docker.

1. Нажмите кнопку Download Docker Desktop for Windows.
2. После загрузки файла запустите его двойным кликом.
3. Следуйте указаниям мастера установки:

* Примите лицензионное соглашение.
* Установите галочку напротив пункта "Use the WSL2 based engine".
* Укажите путь установки (по умолчанию).
* Подтвердите установку дополнительных компонентов.

1. По завершении установки нажмите Close and restart.
2. Перезагрузите компьютер.

### **Шаг 3: Запуск KODA Admin Panel через Docker Compose**

После установки Docker Desktop, необходимо подготовить проект к запуску:

1. Убедитесь, что Docker Desktop запущен.
2. Скопируйте репозиторий проекта или папку с koda-master в удобное место на диске, например: D:\koda-master
3. Запустите **install.bat** (только для Windows). В модальном окне предупреждения необходимо разрешить доступ для этого приложения.
4. Дождитесь завершения процесса сборки контейнеров.
5. После успешной установки будет выведена краткая информация по запуску программы

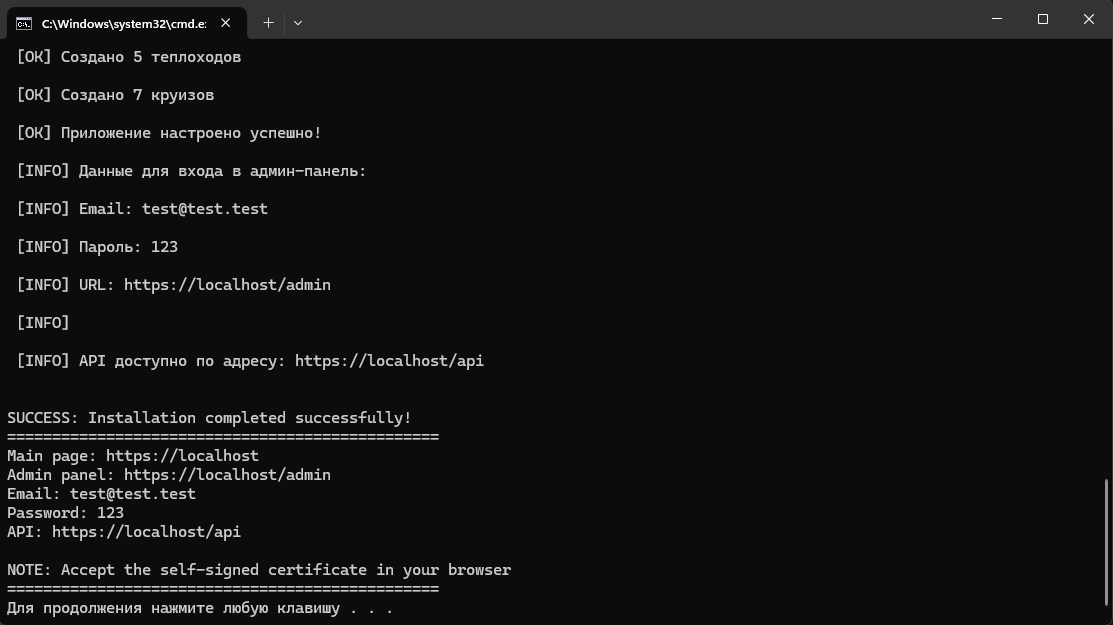


Рисунок 4 - Краткая информация по запуску программы.

Для управления и проверки статуса используйте докер десктоп.

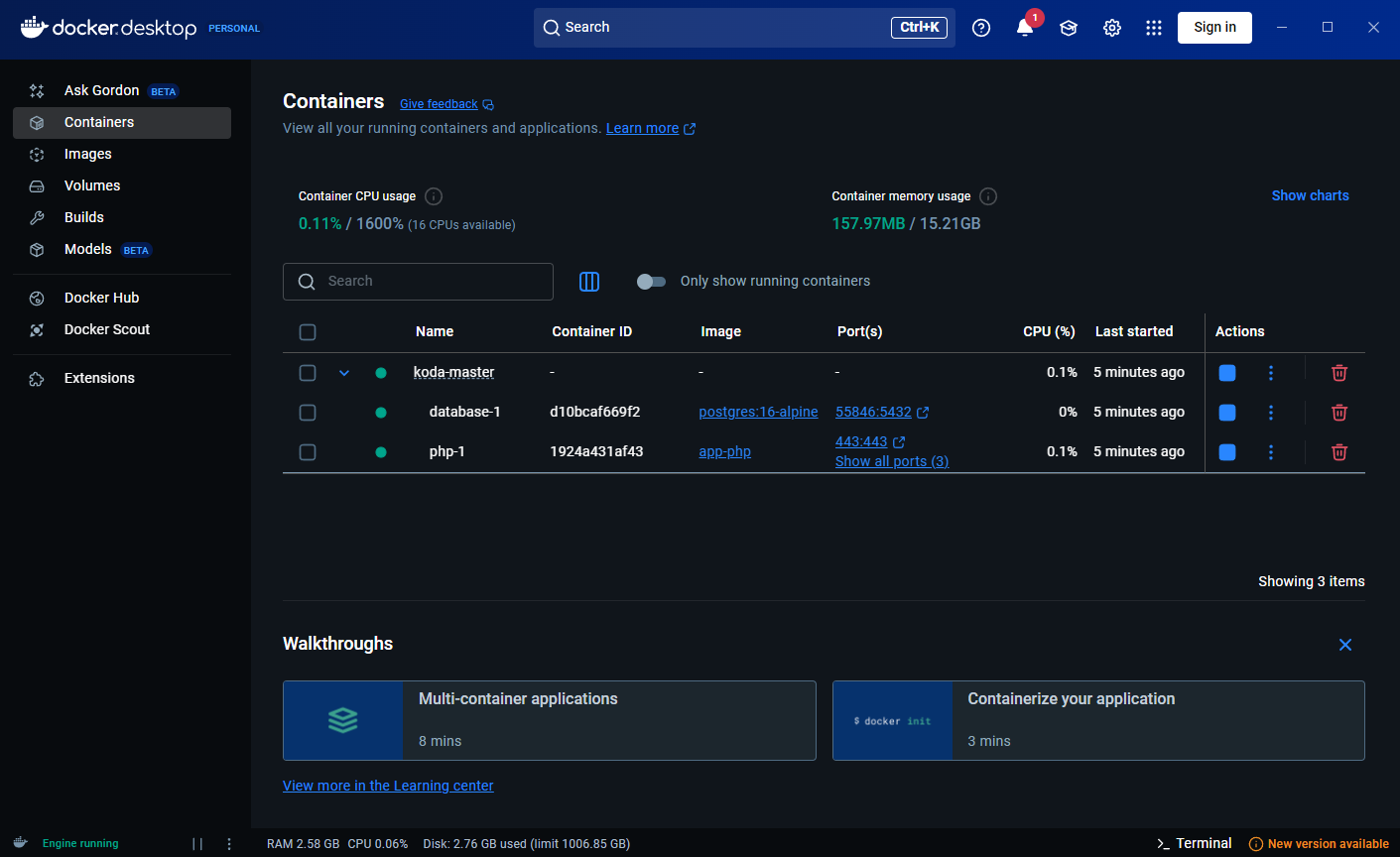


Рисунок 5 – Проверка статуса программы.

## **Вход в систему**

После успешного запуска контейнеров, вы можете получить доступ к административной панели через веб-браузер.

1. Откройте ранее установленный браузер (Google Chrome или Яндекс Браузер).
2. В адресной строке введите:

http://localhost/

1. На странице авторизации введите ваш email и пароль.
2. Нажмите кнопку «Войти».

## **Требования к рабочему месту пользователя**

Для корректной работы с KODA Admin Panel необходимо соблюдение следующих минимальных технических требований:

Таблица 1 – Технические требования к системе.

|  |  |
| --- | --- |
| Характеристика | Минимальные требования |
| Операционная система | Windows 10 / 11 (x64), Linux (Ubuntu 20.04+), MacOS (ограниченная поддержка) |
| Процессор | Intel Core i3 / AMD A6 и выше |
| Оперативная память | 4 ГБ |
| Свободное место на диске | 10 ГБ |
| Браузер | Google Chrome 118+, Яндекс Браузер 23+ |
| Интернет-соединение | Да (для первоначальной загрузки образов Docker и обновлений) |

## **Возможные проблемы и способы их устранения**

Таблица 2 – Возможные проблемы и способы их устранения

|  |  |  |
| --- | --- | --- |
| Проблема | Причина | Решение |
| Не открывается страница  http://localhost/admin | Docker не запущен или контейнер не собран | Запустите Docker Desktop и выполните  docker-compose up -d |
| Контейнеры не запускаются | Конфликт портов | Проверьте, свободны ли порты 80 и 5432 на вашем компьютере |
| Не проходит авторизация | Неверный email или пароль | Проверьте данные входа или обратитесь к администратору |
| Не отображаются картинки или CSS | Проблема с путями в контейнере | Выполните миграции базы данных и очистите кэш (php bin/console cache:clear) |
| Ошибка при запуске Docker | WSL2 не установлен или отключен | Активируйте WSL2 через PowerShell или переустановите Docker |

## **Вспомогательные материалы**

* [Официальная документация Docker](https://docs.docker.com/)
* [Инструкция по настройке WSL2 на Windows](https://docs.microsoft.com/ru-ru/windows/wsl/install)
* [Zip](https://docs.microsoft.com/ru-ru/windows/wsl/install" \t "_blank) архив с приложением

# **2 Описание функциональных разделов**

## **2.1 Общие сведения о функциональных возможностях**

Административная панель KODA Admin Panel предназначена для управления данными круизов, теплоходов и пользователей. Она предоставляет широкий набор инструментов для администрирования, редактирования и анализа данных.

В системе реализованы следующие функциональные разделы:

* Пользователи
* Круизы
* Теплоходы
* Отчеты

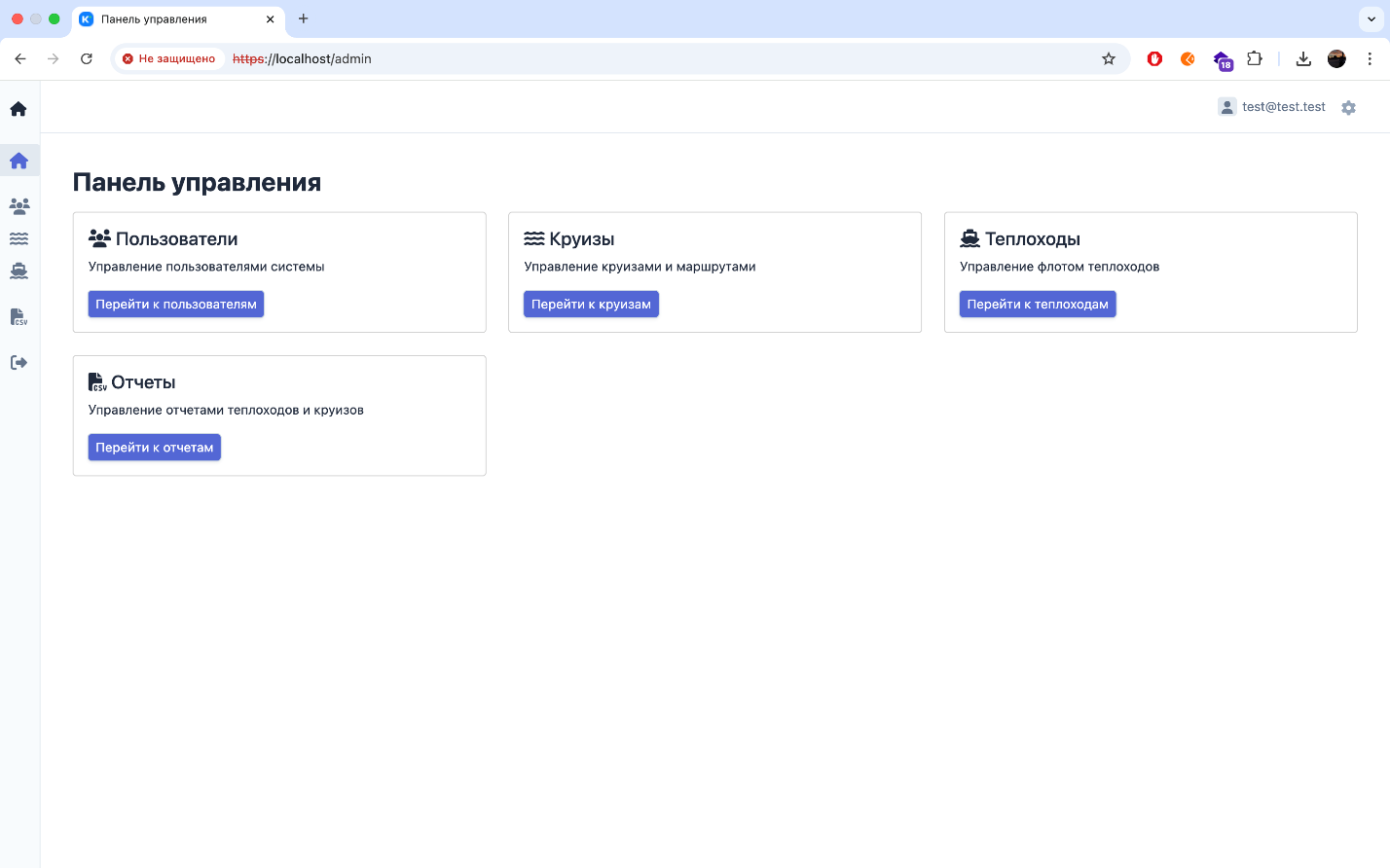


Рисунок 6 – Главная страница KODA

Каждый из этих разделов позволяет выполнять операции добавления, редактирования, просмотра и удаления данных (в зависимости от роли пользователя). Все данные хранятся в базе PostgreSQL и доступны через REST API.

## **2.2 Функциональный раздел «Пользователи»**

### **Назначение**

Раздел «Пользователи» предназначен для управления учетными записями сотрудников, администраторов и менеджеров системы. Здесь можно создавать, редактировать и удалять пользователей, а также назначать им роли.

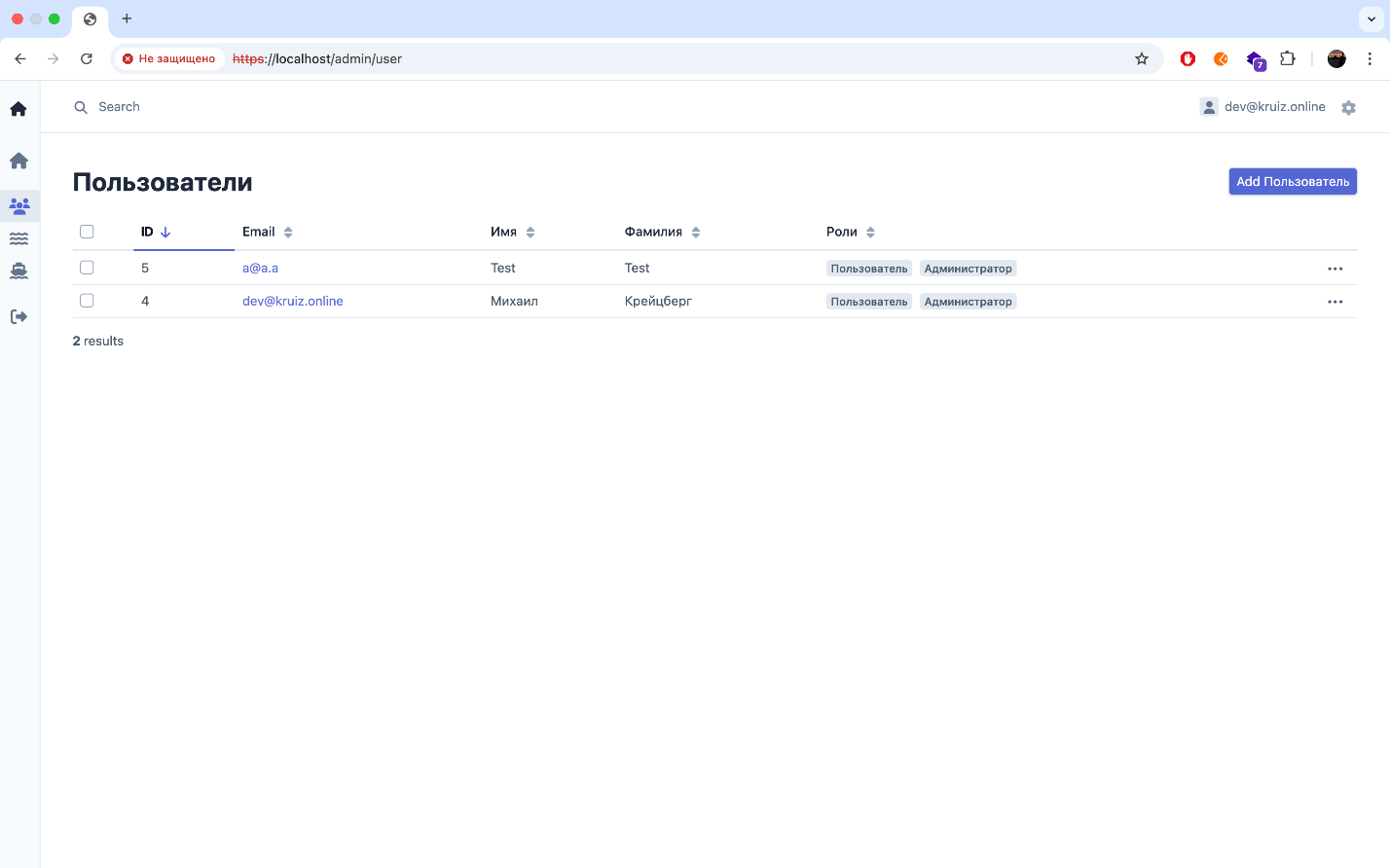


Рисунок 7 – Главная страница раздела «Пользователи».

### **Возможности**

* Добавление нового пользователя:
* Ввод email.
* Установка пароля.
* Выбор роли: Администратор, Менеджер.
* Редактирование информации:
* Изменение email.
* Изменение пароля.
* Изменение роли.

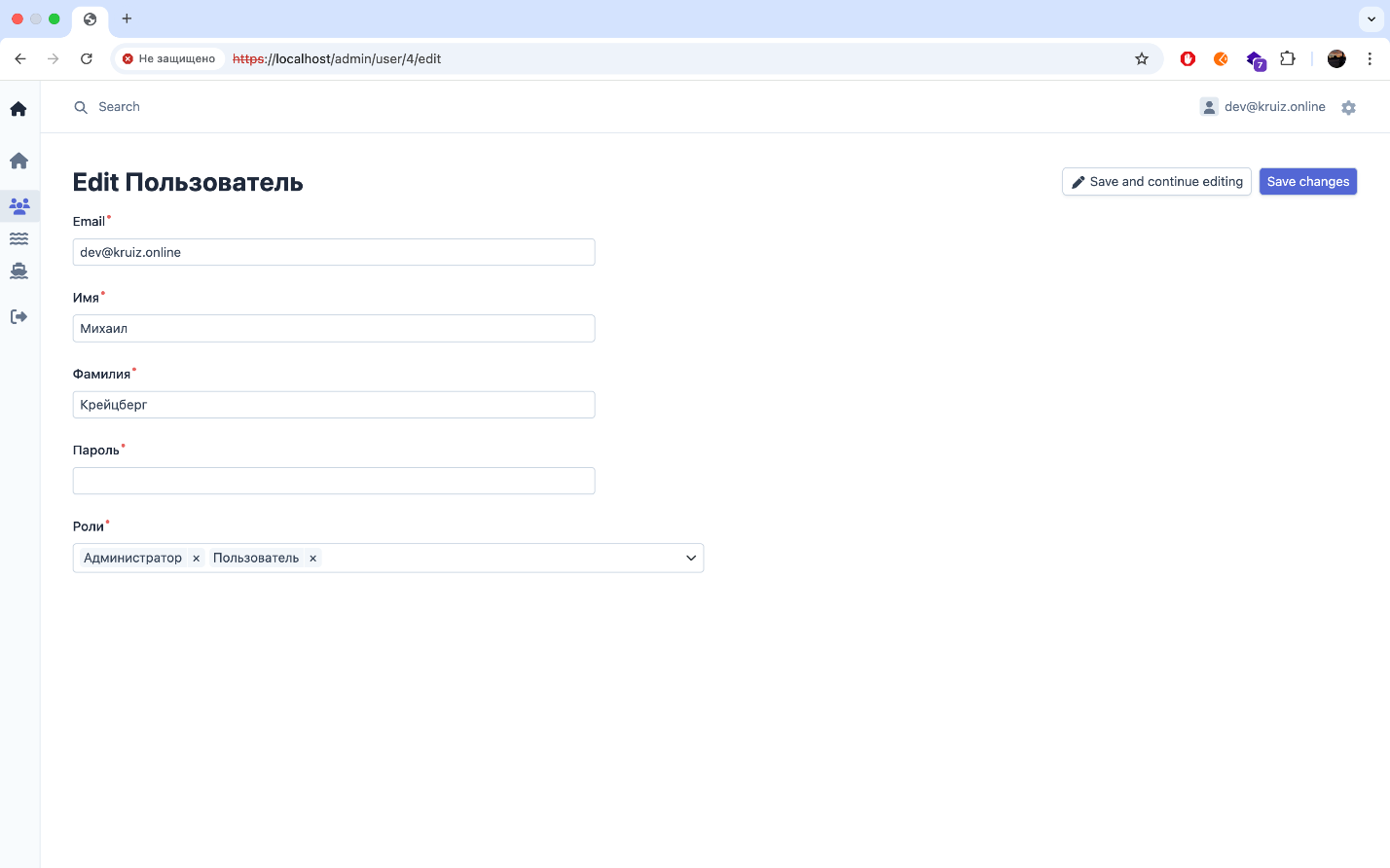


Рисунок 8 – Интерфейс редактирования пользователя.

* Удаление пользователей (только для Администратора).
* Поиск и фильтрация пользователей по имени, email, роли.
* Отображение статуса активности пользователя.

### **Уровни доступа по ролям**

Таблица 3 – Уровень доступа к функциональности страницы «Пользователи» у роли.

|  |  |  |
| --- | --- | --- |
| Функция | Администратор | Менеджер |
| Добавление пользователей | Да | Нет |
| Редактирование пользователей | Да | Нет |
| Удаление пользователей | Да | Нет |
| Просмотр списка пользователей | Да | Нет |

*Примечание - Пароли хранятся в зашифрованном виде с использованием алгоритма bcrypt.*

## **2.3 Функциональный раздел «Круизы»**

### **Назначение**

Раздел «Круизы» используется для добавления, редактирования и просмотра информации о круизах, которые предлагаются клиентам. Каждый круиз связан с определенным теплоходом и имеет свои параметры маршрута, дат, цен и т.д.

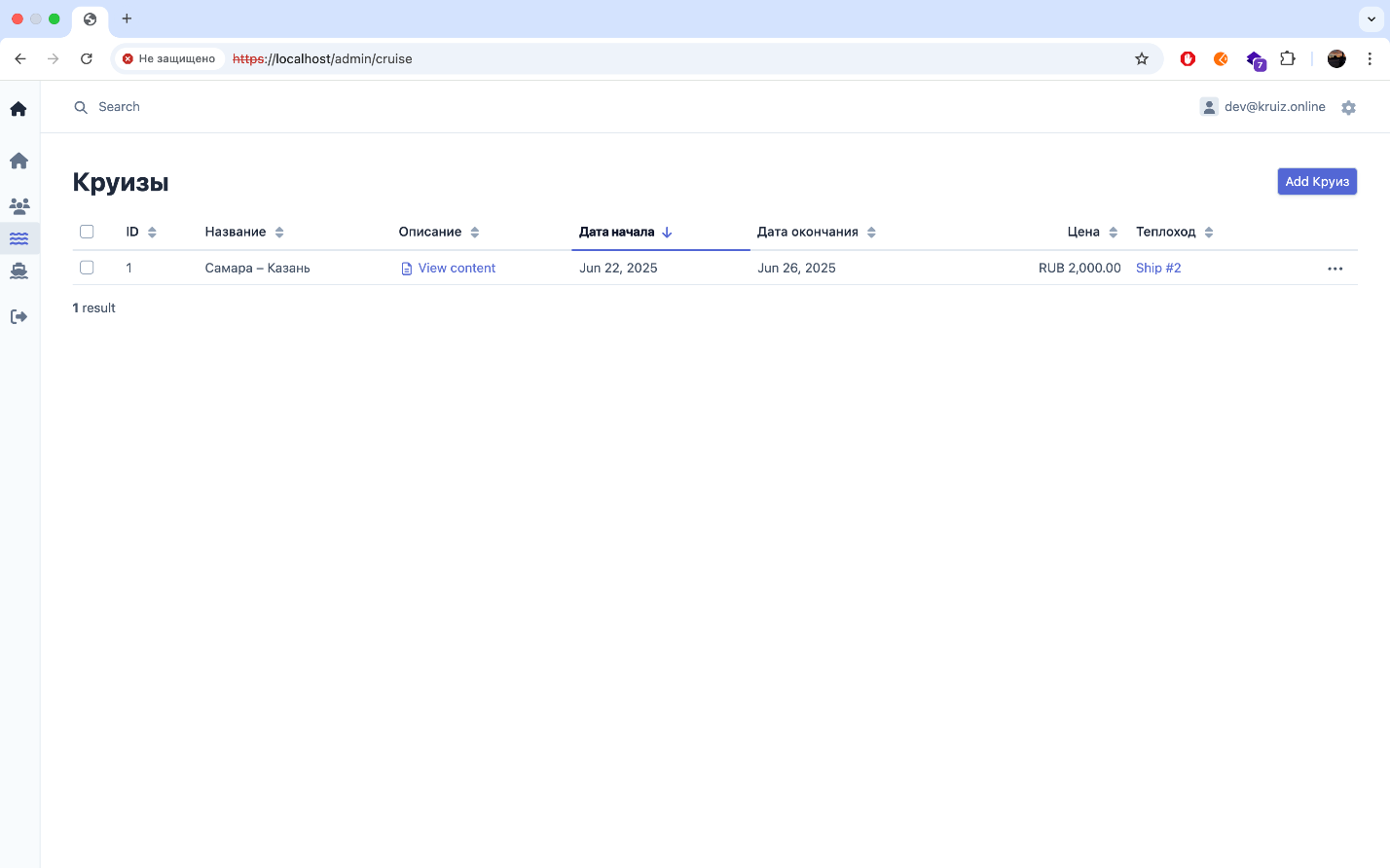


Рисунок 9 – Главная страница раздела «Круизы».

### **Возможности**

* Добавление нового круиза:
* Название круиза.
* Описание.
* Дата начала и окончания.
* Цена.
* Привязка к теплоходу.
* Редактирование круиза:
* Изменение названия, описания, дат, цены
* Перепривязка к другому теплоходу
* Удаление круиза (только для Администратора)

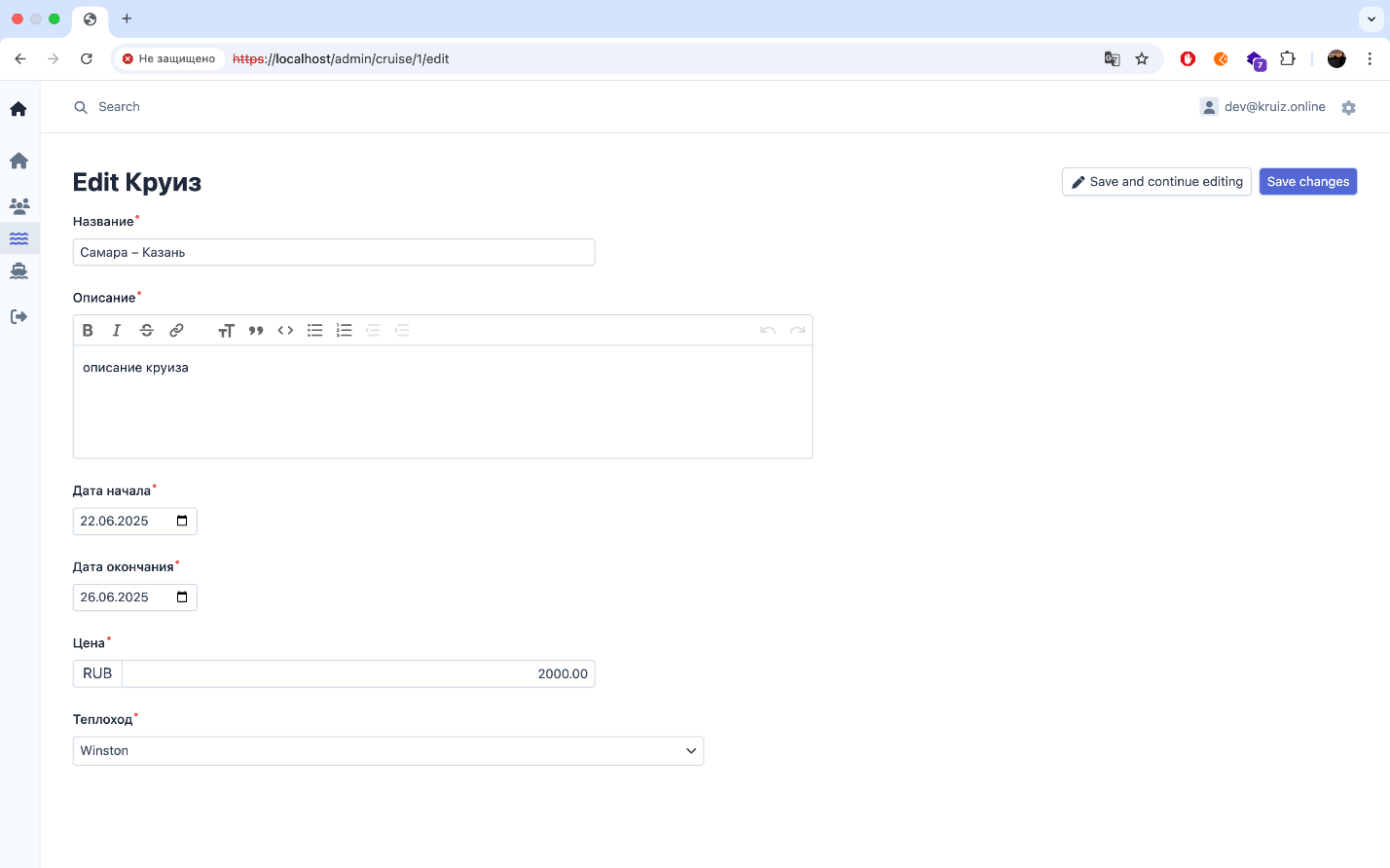


Рисунок 10 – Интерфейс редактирования круиза.

* Фильтрация круизов:
* По названию
* По дате
* По теплоходу
* Экспорт списка круизов в формат Excel

### **Пример** **данных:**

{

"name": "Круиз по Волге",

"description": "7-дневный круиз по реке Волга с посещением исторических городов.",

"startDate": "2025-06-01",

"endDate": "2025-06-07",

"price": 50000,

"ship": {

"id": 1,

"name": "Winston"

}

}

### **Уровни доступа по ролям**

Таблица 4 - Уровень доступа к функциональности страницы «Круизы» у роли.

|  |  |  |
| --- | --- | --- |
| Функция | Администратор | Менеджер |
| Добавление круизов | Да | Да |
| Редактирование круизов | Да | Да |
| Удаление круизов | Да | Да |
| Просмотр списка круизов | Да | Да |

## **2.4 Функциональный раздел «Теплоходы»**

### **Назначение**

Раздел «Теплоходы» предназначен для управления информацией о судах, на которых осуществляются круизы. Здесь можно добавлять новые теплоходы, загружать их фотографии, указывать вместимость и другие технические характеристики.

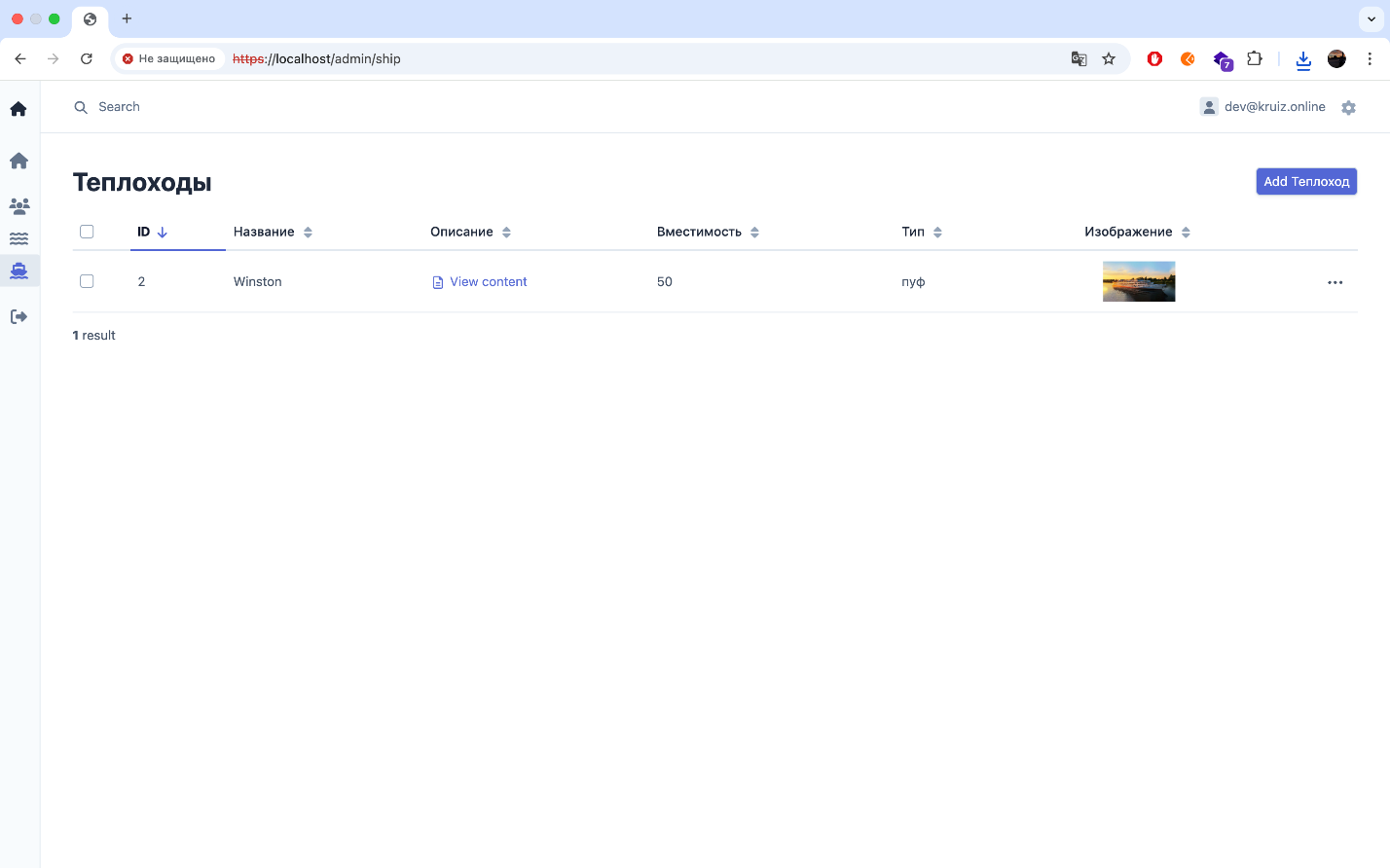


Рисунок 11 – Главная страница раздела «Теплоходы».

### **Возможности**

* Добавление нового теплохода:
* Название
* Описание
* Вместимость
* Тип теплохода
* Загрузка фото
* Редактирование теплохода:
* Изменение всех параметров
* Обновление фотографии
* Удаление теплохода (только для Администратора)

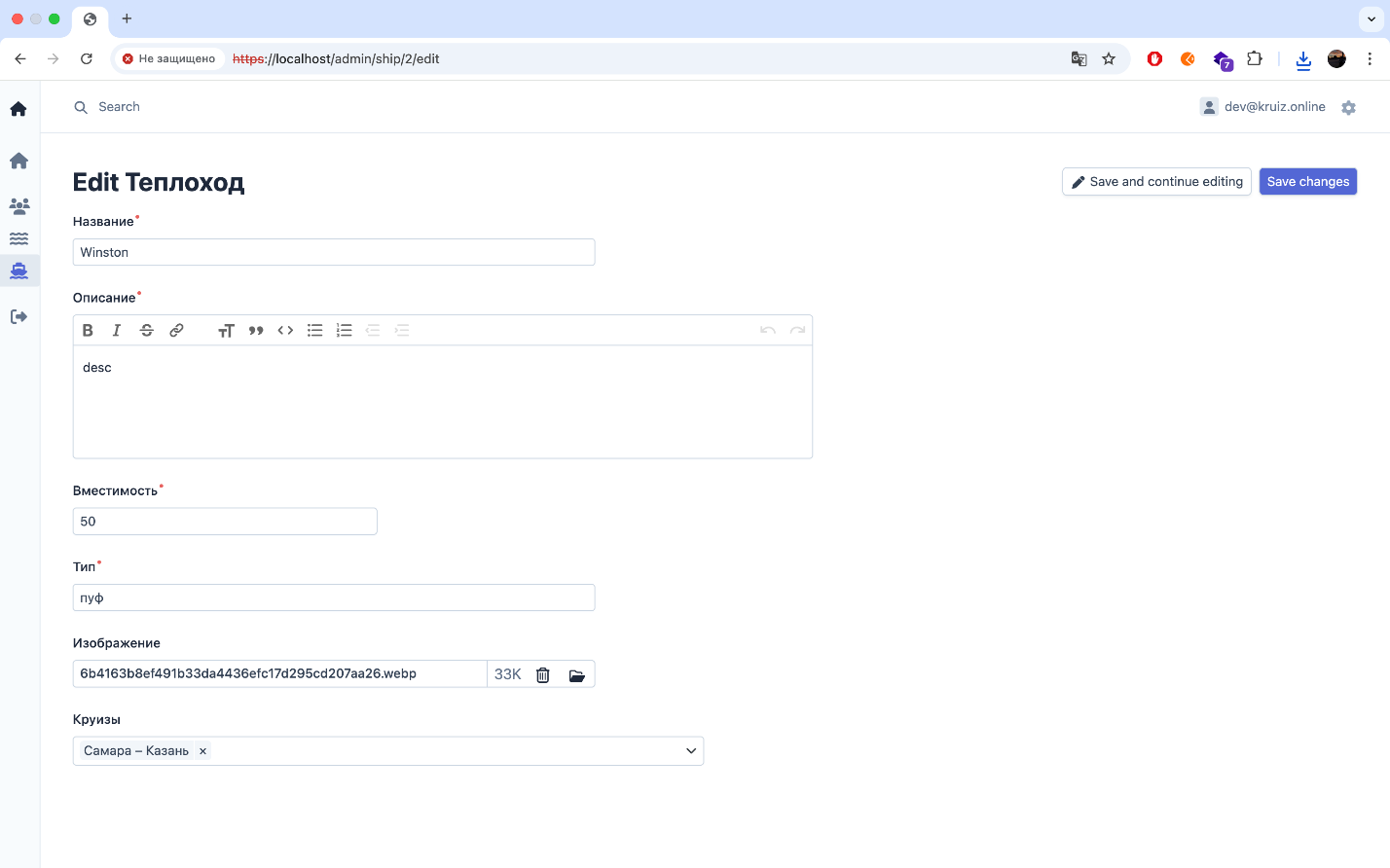


Рисунок 12 – Интерфейс редактирования теплохода.

* Поиск и фильтрация теплоходов:
* По названию
* По типу
* По вместимости
* Просмотр детальной информации о теплоходе

### **Пример данных**

{

"name": "Winston",

"description": "Прогулочный теплоход повышенной комфортности.",

"capacity": 60,

"type": "прогулочный",

"imageUrl": "/uploads/ships/winston.jpg"

}

### **Уровни доступа по ролям**

Таблица 5 – Уровень доступа к функциональности страницы «Теплоходы» у роли.

|  |  |  |
| --- | --- | --- |
| Функция | Администратор | Менеджер |
| Добавление теплоходов | Да | Да |
| Редактирование теплоходов | Да | Да |
| Удаление теплоходов | Да | Да |
| Просмотр списка теплоходов | Да | Да |

## **2.5 Функциональный раздел «Отчеты»**

### **Назначение**

Раздел «Отчеты» создан для автоматической генерации отчетов по круизам и теплоходам по фильтрам. Он позволяет анализировать эффективность работы компании, выявлять популярные направления и планировать бюджет.

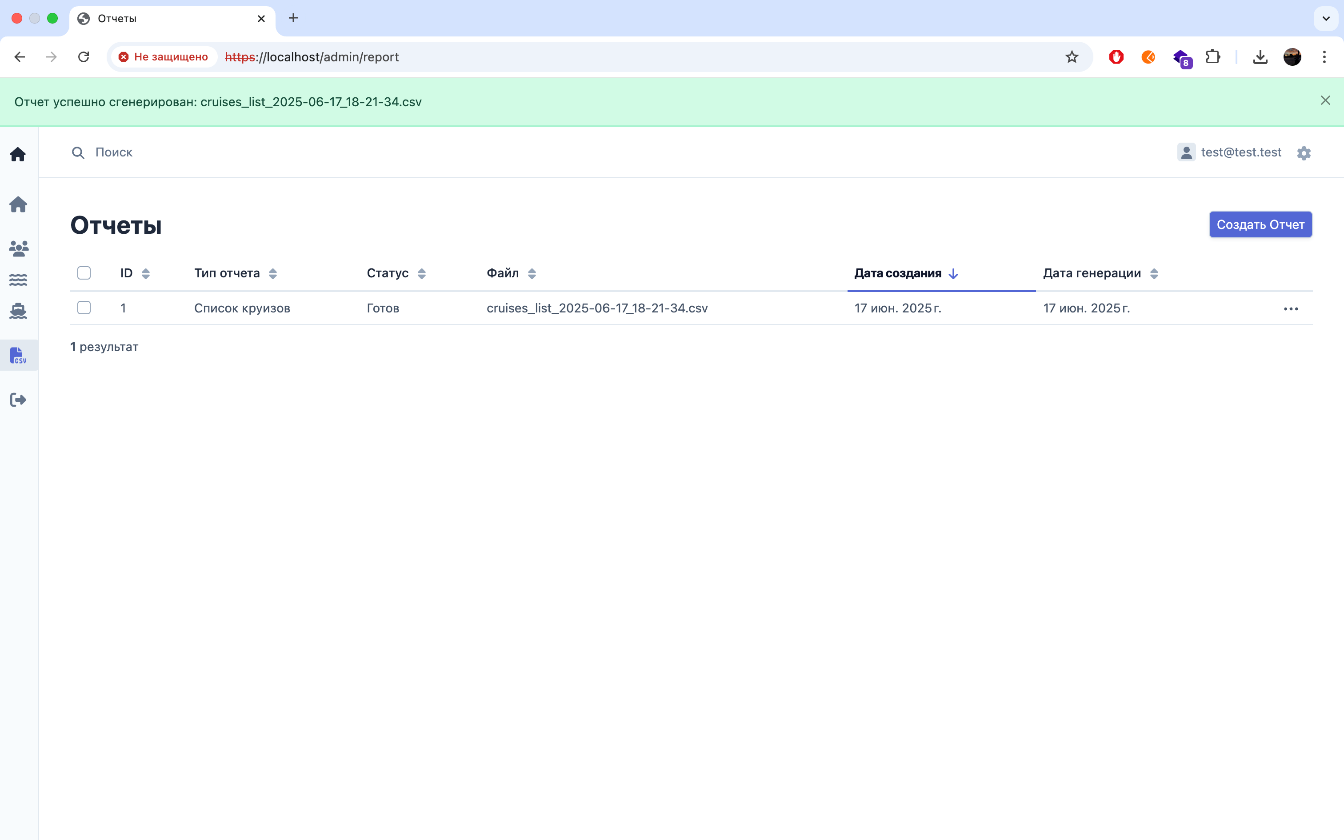


Рисунок 13 – Главная страница раздела «Отчеты».

### **Возможности**

* Формирование отчетов:

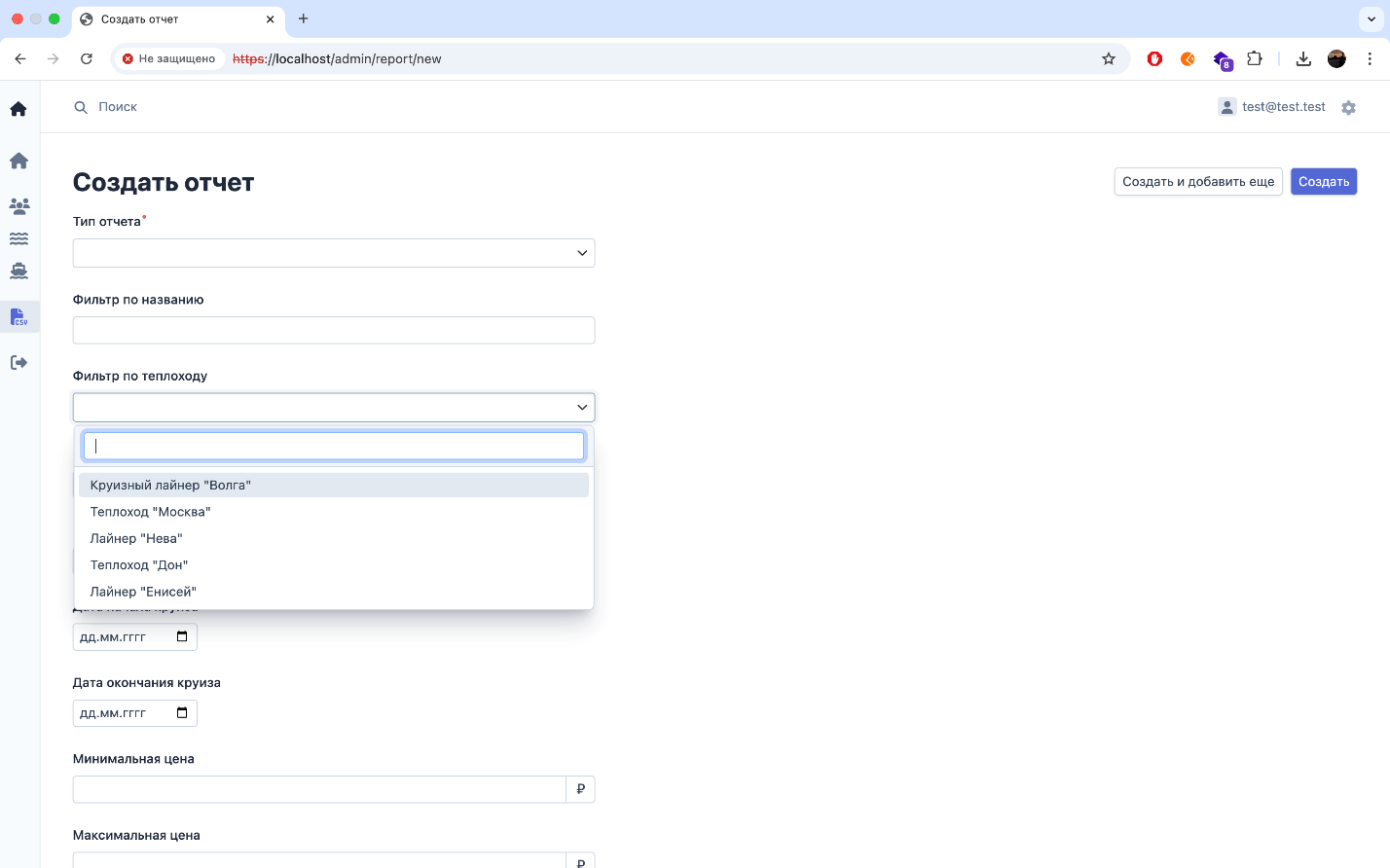


Рисунок 14 – Интерфейс редактирования отчета.

* По круизам:
* Список всех круизов
* Финансовые показатели
* Сравнение по датам и маршрутам
* По теплоходам:
* Общее количество рейсов
* Число перевезенных пассажиров
* Доход по каждому теплоходу
* Комбинированные отчеты:
* По дате
* По названию
* По статусу
* Экспорт отчетов в формат Excel
* Просмотр деталей отчета.

### **Пример отчета**

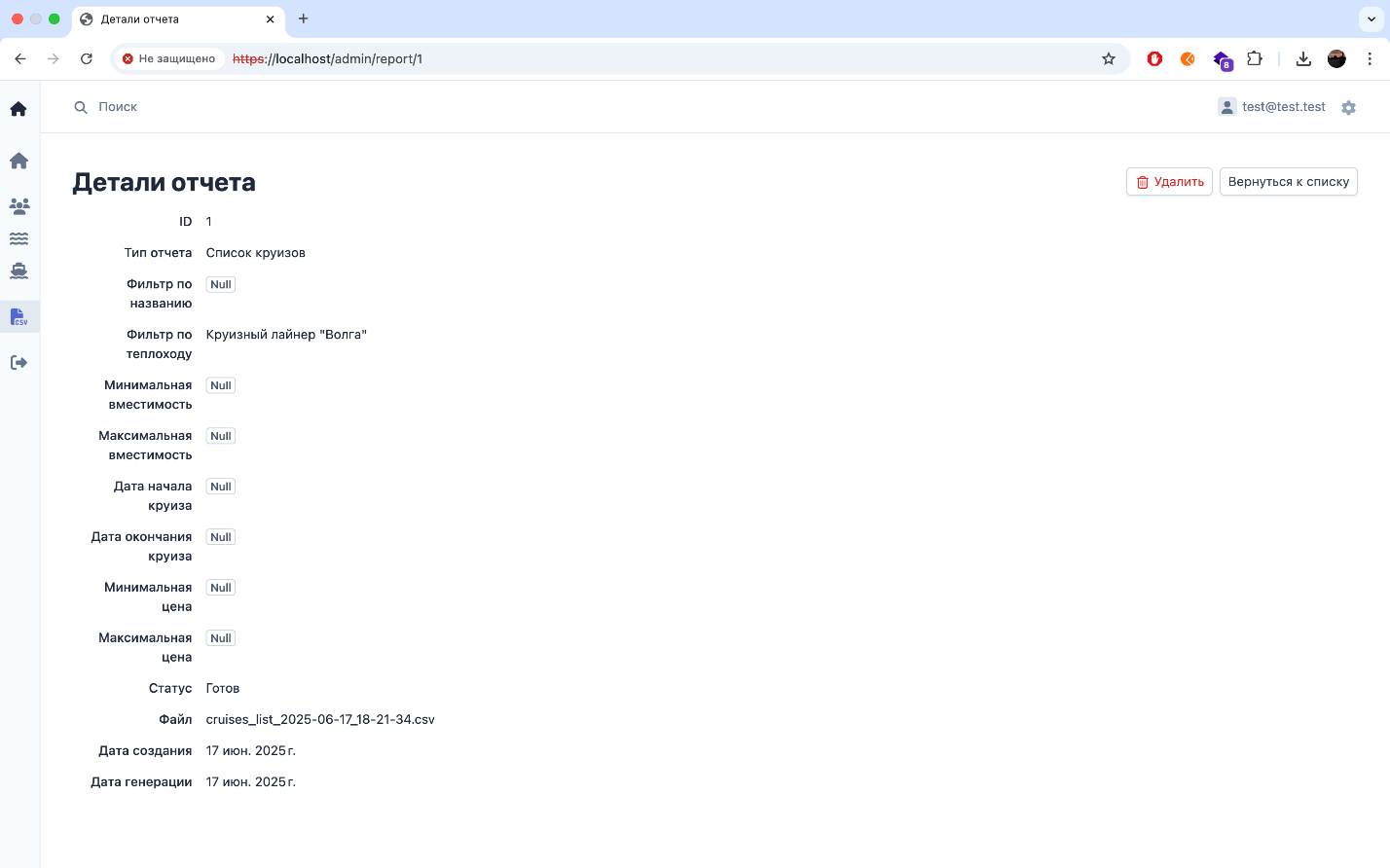


Рисунок 15 – Пример отчета.

### **Уровни доступа по ролям**

Таблица 6 – Уровень доступа к функциональности страницы «Отчет» у роли.

|  |  |  |
| --- | --- | --- |
| Функция | Администратор | Менеджер |
| Генерация отчетов | Да | Да |
| Экспорт в Excel | Да | Да |
| Просмотр деталей | Да | Да |
| Удаление отчетов | Да | Да |

## **2.6 Дополнительные возможности**

### **2.6.1 Авторизация и управление ролями**

* Авторизация происходит по email и паролю.
* Реализованы три роли:
* Администратор: полный доступ ко всем разделам
* Менеджер: ограниченный доступ, возможность редактирования данных
* Все права и роли настраиваются через файл конфигурации security.yaml и управляются вручную в базе данных.

### **2.6.2 Интеграция с REST API**

Система предоставляет REST API для внешних сервисов:

* GET /api/ships — получить список теплоходов.
* GET /api/ships/{id} — получить теплоход по ID.
* GET /api/cruises — получить список круизов.
* GET /api/cruises/{id} — получить круиз по ID.

Пример использования:

Javascript

fetch('/api/cruises?page=1&limit=10')

.then(response => response.json())

.then(data => console.log(data));

### **2.6.3 Поддержка JSON-формата**

Все ответы сервера возвращаются в формате JSON с кодировкой UTF-8.

### **2.6.4 Коды ошибок**

* 200 – успешный запрос.
* 400 – неверный запрос.
* 401 – не авторизован.
* 403 – доступ запрещен.
* 404 – ресурс не найден.
* 405 – метод не разрешен.
* 422 – ошибка валидации.
* 500 – внутренняя ошибка сервера.

# **3 Развертывание на сервере**

## **3.1 Общие положения**

Для полноценного использования административной панели KODA Admin Panel в условиях предприятия или организации, рекомендуется развернуть систему на внешнем сервере (VPS/VDS) или в облаке. Это позволит обеспечить доступ к системе из любой точки, а также организовать централизованное хранение и обработку данных.

Система поставляется в виде контейнеров Docker и может быть развернута как локально, так и на удаленном сервере с использованием docker-compose.

## **3.2 Требования к серверу**

Для корректной работы системы на сервере должны быть выполнены следующие минимальные технические требования:

Таблица 7 – Системные требования к серверу.

|  |  |
| --- | --- |
| Характеристика | Минимальные требования |
| Операционная система | Ubuntu 20.04 LTS / Debian 11+ |
| Процессор | 2 ядра |
| Оперативная память | 4 ГБ |
| Свободное место на диске | 20 ГБ |
| Доступ в интернет | Да |
| Установленный Docker и Docker Compose | Да |

*Примечание - рекомендуется использовать SSD-диск для ускорения работы базы данных и повышения отзывчивости системы.*

Требования к Администратору KODA:

Основными требованиями к квалификации Администратора являются:

• наличие знаний в области администрирования операционных систем рабочих станций и серверного оборудования, а также знаний в области администрирования СУБД, используемой в KODA;

• наличие знаний в области принципов и методов резервного копирования и аварийного восстановления данных, конфигурирования и настройки программно-технических средств подсистемы, а также навыков инсталляции и настройки клиентских приложений.

## **3.3 Подготовка сервера**

### **Шаг 1: Обновление системы**

* Подключитесь к серверу через SSH:

ssh username@your\_server\_ip

* Обновите список пакетов и установите актуальные версии программ:

sudo apt update && sudo apt upgrade -y

### **Шаг 2: Установка Docker**

Установите Docker и плагин Docker Compose для вашей операционной системы согласно документации с оффициальнгого сайта https://docs.docker.com/engine/install/

## **3.4 Настройка домена и SSL-сертификата**

Для обеспечения безопасного HTTPS соединения необходимо настроить домен и получить SSL-сертификат.

### **Привязка домена**

Зарегистрируйте домен (например, koda-admin-panel.ru).

В настройках DNS добавьте запись A, указывающую на IP-адрес вашего сервера.

## **3.5 Развертывание приложения**

### **Шаг 1: Получение репозитория**

* Клонируйте репозиторий с проектом или скачайте архив с приложением KODA.
* Откройте файл .env, отредактируйте переменные
* SERVER\_NAME: **реальное\_имя\_домена.ru**
* MERCURE\_PUBLISHER\_JWT\_KEY: ChangeThis
* MERCURE\_SUBSCRIBER\_JWT\_KEY: ChangeThis
* DATABASE\_URL: postgresql://${POSTGRES\_USER}:${POSTGRES\_PASSWORD}@database:5432/${POSTGRES\_DB}?serverVersion=${POSTGRES\_VERSION}&charset=${POSTGRES\_CHARSET}
* POSTGRES\_VERSION:16 (**версия постгрес)**
* POSTGRES\_DB: app **(имя базы данных)**
* POSTGRES\_PASSWORD: !ChangeMe! **(пароль от базы данных)**
* POSTGRES\_USER: app **(пользователь базы данных)**
* POSTGRES\_CHARSET:utf8
* MERCURE\_URL: http://php/.well-known/mercure
* MERCURE\_PUBLIC\_URL:https://${SERVER\_NAME}:${HTTPS\_PORT}/.well-known/mercure}
* MERCURE\_JWT\_SECRET: ChangeThis
* SYMFONY\_VERSION:
* STABILITY: stable
* HTTP\_PORT:80
* HTTPS\_PORT:443
* HTTP3\_PORT:443
* APP\_ENV=prod
* APP\_SECRET=**your\_secret\_key**
* CADDY\_MERCURE\_PUBLIC\_URL=https://${SERVER\_NAME}/.well-known/mercure
* CADDY\_MERCURE\_URL=http://${SERVER\_NAME}/.well-known/mercure

**Важно:** указывайте **реальный домен** и **настрой DNS**, чтобы он указывал на IP сервера.

* Откройте файл compose.yaml, отредактируйте значения по умолчанию в переменных окружения

*Пример*:

* POSTGRES\_PASSWORD: ${POSTGRES\_PASSWORD:-!ChangeMe!}

POSTGRES\_PASSWORD – имя\_переменной

${POSTGRES\_PASSWORD:- } - команда для установки значения не из файла переменных .env, а из строки.

!ChangeMe! – значение переменной по умолчанию, нужно изменить на собственное.

Чтобы переменные применились из файла .env удалите все строки **:${значение}**

**Важно:** не оставляйте значения по умолчанию!

### **Шаг 2: Запуск контейнеров**

Выполните команду

docker compose up -d --build

из каталога в котором находятся исходные коды KODA

После запуска система будет доступна по адресу:

https://реальное\_имя\_домена.ru/admin

также KODA автоматически получит сертификаты SSL от Let's Encrypt

## **3.6 Открытие портов**

Для внешнего доступа откройте следующие порты на вашем сервере:

Таблица 8 – Порты сервера.

|  |  |
| --- | --- |
| Порт | Назначение |
| 80 | HTTP (для перенаправления на HTTPS) |
| 443 | HTTPS (безопасное соединение) |
| 22 | SSH (для управления сервером) |

Настройка фаервола:

sudo ufw allow OpenSSH

sudo ufw allow 80

sudo ufw allow 443

sudo ufw enable

## **3.7 Бэкапы и восстановление данных**

Регулярно создавайте резервные копии базы данных PostgreSQL:

docker exec -t koda-db pg\_dump -U postgres -Fc app > backup\_$(date +%F).dump

Для восстановления:

pg\_restore -U postgres -d app -Fc -v backup\_2025-04-05.dump

## **3.8 Мониторинг состояния системы**

Используйте инструменты мониторинга, такие как:

* Netdata
* Prometheus + Grafana
* Portainer (для управления контейнерами)

Пример установки Portainer:

docker volume create portainer\_data

docker run -d -p 9443:9443 --name portainer --restart always -v /var/run/docker.sock:/var/run/docker.sock -v portainer\_data:/data portainer/portainer-ce:latest

## **3.9 Возможные проблемы и способы их устранения**

Таблица 9 - Возможные проблемы и способы их устранения

|  |  |  |
| --- | --- | --- |
| Проблема | Причина | Решение |
| Не открывается сайт | Сертификат не выдан | Проверьте настройки домена и Caddy |
| База данных не запускается | Конфликт портов | Убедитесь, что порт 5432 свободен |
| Контейнеры не работают | Ошибки вdocker-compose.yml | Проверьте синтаксис и пути |
| Не работает авторизация | Неверный секретный ключ | Измените APP\_SECRETв.env |
| Ошибка 500 | Проблема с миграциями | Выполните php bin/console doctrine:migrations:migrate внутри контейнера |

## **3.10 Вспомогательные материалы**

[Официальная документация Docker](https://docs.docker.com/)

(<https://docs.docker.com/>)

[Docker Compose Reference](https://docs.docker.com/compose/reference/)

(https://docs.docker.com/compose/reference/)

[Caddy Server Documentation](https://caddyserver.com/docs/)

(https://caddyserver.com/docs/)

[Let's Encrypt Free SSL Certificates](https://letsencrypt.org/)

(https://letsencrypt.org/)

# **4 Лицензирование и юридические аспекты**

## **Общие положения**

Для обеспечения законности использования, распространения и защиты интеллектуальной собственности, разработчик системы ООО «Круиз.онлайн» предоставляет программное обеспечение KODA Admin Panel с четко определенной лицензией и указанием используемых компонентов.

В данном документе описаны:

Лицензии на используемые open-source технологии

Лицензия на само ПО (KODA Admin Panel)

Юридическая информация компании

Указание на соответствие требованиям реестра отечественного ПО

## **Использованные open-source технологии и их лицензии**

KODA Admin Panel разработан с использованием современных open-source технологий. Ниже приведен список основных библиотек, фреймворков и сервисов, а также их лицензии:

Таблица 10 - Использованные open-source технологии

|  |  |  |
| --- | --- | --- |
| Компонент | Версия | Лицензия |
| PHP | 8.x | PHP License |
| Symfony | 6.x | MIT |
| PostgreSQL | 15 | PostgreSQL License |
| Docker | 20.10 | Apache 2.0 |
| Docker Compose | 2.x | Apache 2.0 |
| EasyAdmin | 3.x | MIT |
| Doctrine ORM | ~3.0 | MIT |
| Twig | ~3.0 | BSD-3-Clause |
| bcrypt | built-in | Open Source |
| FrankenPHP / Caddy | latest | Apache 2.0 |
| Bootstrap | 5.x | MIT |
| jQuery | 3.x | MIT |
| Axios | latest | MIT |

*Примечание - Все перечисленные технологии используются в рамках разрешенных условий их лицензий, без модификации исходного кода ядра, за исключением пользовательских кастомизаций интерфейса и бизнес-логики.*

## **Лицензия на программное обеспечение KODA Admin Panel**

KODA Admin Panel v1.0 является оригинальной разработкой ООО «Круиз.онлайн», предназначенной для управления данными круизов, теплоходов и пользователей.

Тип лицензии: Proprietary (проприетарная)

Это означает, что:

* Программа не является свободным или открытым ПО.
* Распространение программы возможно только по договору с правообладателем.
* Копирование, изменение, использование программы без письменного разрешения запрещено.
* Авторские права принадлежат ООО «Круиз.онлайн».

# **5 Описание REST API**

## **5.1 Общая информация**

KODA Admin Panel предоставляет интерфейс программирования приложений (REST API) для взаимодействия с системой из внешних сервисов, таких как CRM, мобильные приложения, сторонние сайты и т.п.

API реализован в виде набора HTTP-запросов, возвращающих данные в формате JSON с кодировкой UTF-8.

Все запросы отправляются на базовый URL:

https://localhost/api

*Примечание - Для большинства операций требуется авторизация, за исключением чтения данных, если это разрешено политикой безопасности.*

## **5.2 Методы и эндпоинты**

Таблица 11 - Список доступных эндпоинтов

|  |  |  |
| --- | --- | --- |
| Метод | URL | Описание |
| GET | /api/ships | Получить список теплоходов |
| GET | /api/ships/{id} | Получить теплоход по ID |
| GET | /api/cruises | Получить список круизов |
| GET | /api/cruises/{id} | Получить круиз по ID |

## **5.3 Формат ответов**

Все ответы системы возвращаются в формате JSON и содержат следующие поля:

{

"data": [...],

"pagination": {

"page": 1,

"limit": 10,

"total": 100,

"pages": 10,

"has\_next": true,

"has\_prev": false

}

}

В случае ошибки:

{

"error": {

"code": 404,

"message": "Ресурс не найден"

}

}

## **5.4 Коды ошибок**

Таблица 12 – Коды ошибок.

|  |  |
| --- | --- |
| Код | Описание |
| 200 | Успешный запрос |
| 400 | Некорректный запрос |
| 401 | Не авторизован |
| 403 | Доступ запрещен |
| 404 | Ресурс не найден |
| 405 | Метод не разрешен |
| 422 | Ошибка валидации |
| 500 | Внутренняя ошибка сервера |

## **5.5 Параметры запросов**

**GET /api/ships**

Получает список теплоходов с возможностью пагинации.

Параметры:

* page — номер страницы (по умолчанию: 1)
* limit — количество записей на странице (по умолчанию: 10, максимум: 50)

Пример запроса:

GET /api/ships?page=1&limit=5

Пример ответа:

{

"data": [

{

"id": 1,

"name": "Winston",

"description": "Современный прогулочный теплоход",

"capacity": 60,

"type": "прогулочный",

"imageUrl": "/uploads/ships/winston.jpg"

}

],

"pagination": {

"page": 1,

"limit": 5,

"total": 10,

"pages": 2,

"has\_next": true,

"has\_prev": false

}

}

**GET /api/ships/{id}**

Получает детальную информацию о теплоходе по его ID.

Пример запроса:

GET /api/ships/1

Пример ответа:

{

"id": 1,

"name": "Winston",

"description": "Современный прогулочный теплоход",

"capacity": 60,

"type": "прогулочный",

"imageUrl": "/uploads/ships/winston.jpg"

}

**GET /api/cruises**

Получает список круизов с возможностью пагинации.

Параметры:

* page — номер страницы (по умолчанию: 1)
* limit — количество записей на странице (по умолчанию: 10, максимум: 50)

Пример запроса:

GET /api/cruises?page=1&limit=5

Пример ответа:

{

"data": [

{

"id": 1,

"name": "Круиз по Волге",

"description": "7-дневный круиз по реке Волга",

"startDate": "2025-06-01",

"endDate": "2025-06-07",

"price": 50000.0,

"ship": {

"id": 1,

"name": "Winston",

"description": "Современный прогулочный теплоход",

"capacity": 60,

"type": "прогулочный",

"imageUrl": "/uploads/ships/winston.jpg"

}

}

],

"pagination": {

"page": 1,

"limit": 5,

"total": 15,

"pages": 3,

"has\_next": true,

"has\_prev": false

}

}

**GET /api/cruises/{id}**

Получает детальную информацию о круизе по его ID.

Пример запроса:

GET /api/cruises/1

Пример ответа:

{

"id": 1,

"name": "Круиз по Волге",

"description": "7-дневный круиз по реке Волга",

"startDate": "2025-06-01",

"endDate": "2025-06-07",

"price": 50000.0,

"ship": {

"id": 1,

"name": "Winston",

"description": "Современный прогулочный теплоход",

"capacity": 60,

"type": "прогулочный",

"imageUrl": "/uploads/ships/winston.jpg"

}

}

## **5.6 Примеры использования**

JavaScript (Fetch API)

Получение списка теплоходов:

fetch('/api/ships?page=1&limit=10')

.then(response => response.json())

.then(data => {

console.log("Теплоходы:", data.data);

console.log("Пагинация:", data.pagination);

});

Получение конкретного круиза:

fetch('/api/cruises/1')

.then(response => response.json())

.then(cruise => {

console.log("Круиз:", cruise);

});

cURL

Получение списка теплоходов:

curl -k -X GET "https://localhost/api/ships?page=1&limit=5"

Получение конкретного круиза:

curl -k -X GET "https://localhost/api/cruises/1"

Python (requests)

Получение списка теплоходов:

import requests

response = requests.get("https://localhost/api/ships?page=1&limit=5", verify=False)

data = response.json()

print("Теплоходы:", data['data'])

print("Пагинация:", data['pagination'])

Получение конкретного круиза:

response = requests.get("https://localhost/api/cruises/1", verify=False)

cruise = response.json()

print("Круиз:", cruise)

## **5.7 Ограничения**

* Максимальное количество записей на странице: 50
* Минимальное количество записей на странице: 1
* Сортировка: данные возвращаются отсортированными по возрастанию ID (новые записи в начале)
* SSL сертификаты: при локальной разработке можно использовать флаг -k или verify=False, но в production обязательно использовать действующий SSL-сертификат