

Руководство администратора BaiMuras Platform v2.0.0

Обзор платформы

BaiMuras Platform v2.0.0 - это комплексная веб-платформа для мебельного бизнеса, включающая:

- **Основной сайт** с портфолио и формами заявок
 - **CRM-систему** для управления лидами и проектами
 - **Админ-панель** для управления контентом
 - **Автоматизацию** бизнес-процессов через n8n
 - **API** для интеграций
-

Архитектура и готовность к продакшену

Техническая архитектура

Основные компоненты:

- **Flask** (Python 3.11+) - веб-фреймворк
- **PostgreSQL 15** - основная база данных
- **Redis 7** - кэширование и очереди задач
- **n8n** - автоматизация бизнес-процессов
- **Celery** - фоновые задачи
- **Nginx** - веб-сервер и прокси
- **Gunicorn** - WSGI сервер

Контейнеризация:

```
# Сервисы в docker-compose.yml:  
- web (Flask приложение)  
- db (PostgreSQL)  
- redis (Redis)  
- n8n (автоматизация)  
- worker (Celery worker)  
- scheduler (Celery beat)  
- nginx (веб-сервер)
```

Готовность к продакшену

Безопасность:

- Pylint Score: 9.21/10
- Все критические уязвимости устранены (Bandit scan)
- CSRF защита включена
- Rate limiting настроен
- Secure headers настроены
- JWT аутентификация для API

Производительность:

- Connection pooling для БД
- Redis кэширование
- Lazy loading изображений
- Оптимизированные SQL запросы

Мониторинг:

- Health check эндпоинты
- Структурированное логирование
- Автоматические backup

Масштабируемость:

- Горизонтальное масштабирование через Docker
- Отдельные контейнеры для разных сервисов
- Асинхронная обработка задач

Стилистические решения и дизайн

Цветовая схема

Основная палитра:

```
:root {
  --primary-blue: #6b70ba;      /* Основной синий */
  --wood-tone: #8d8056;         /* Древесный тон */
  --light-wood: #d4de54;        /* Светлое дерево */
  --soft-blue: #a3c2d3;         /* Мягкий синий */
  --neutral-gray: #848e79;      /* Нейтральный серый */
  --pure-white: #ffffff;        /* Чистый белый */
  --dark-gray: #2c2c2c;         /* Темно-серый */
}
```

Дополнительные цвета:

- --light-gray: #f8f9fa - фоны секций
- --medium-gray: #e9ecef - границы
- --text-dark: #1a1a1a - основной текст
- --text-muted: #4a5568 - вторичный текст

Типографика

Шрифты:

- **Основной:** Inter (Google Fonts)
- **Заголовки:** Inter, веса 600-700
- **Текст:** Inter, веса 400-500

Размеры:

- H1: 2.5rem (40px)
- H2: 2rem (32px)
- H3: 1.5rem (24px)
- H4: 1.25rem (20px)
- Основной текст: 1rem (16px)

Логотипы и брендинг

Файлы логотипов:

- `baimuras_logo_main.png` - основной цветной логотип
- `baimuras_logo_mono.png` - монохромная версия
- `logo.png` - стандартный логотип
- `baimuras_favicon.png` - фавикон

Фоновые элементы:

- `coral_accent_bg.png` - коралловый акцент
- `blue_splash_bg.png` - синий всплеск

Дизайн-система

Отступы:

```
--spacing-xs: 0.5rem; /* 8px */
--spacing-sm: 1rem; /* 16px */
--spacing-md: 1.5rem; /* 24px */
--spacing-lg: 2rem; /* 32px */
--spacing-xl: 3rem; /* 48px */
--spacing-xxl: 4rem; /* 64px */
```

Тени:

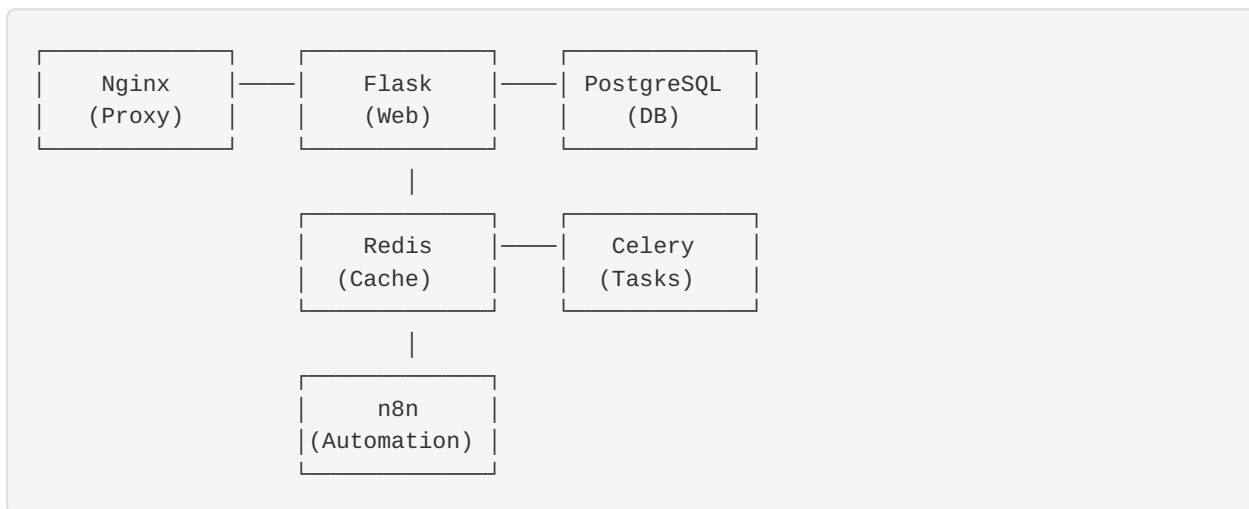
```
--shadow-sm: 0 1px 3px rgba(0, 0, 0, 0.1);
--shadow-md: 0 4px 6px rgba(0, 0, 0, 0.1);
--shadow-lg: 0 10px 25px rgba(0, 0, 0, 0.1);
```

Скругления:

```
--radius-sm: 4px;
--radius-md: 8px;
--radius-lg: 12px;
```

Интеграция сервисов и взаимодействие

Схема взаимодействия сервисов



Flask приложение

Основные модули:

- `src/main.py` - точка входа
- `src/routes/` - маршруты и контроллеры
- `src/models/` - модели данных
- `src/utils/` - утилиты и хелперы

Ключевые маршруты:

- `/` - главная страница
- `/contact` - форма обратной связи
- `/dashboard` - админ-панель
- `/api/v1/` - API эндпоинты
- `/webhooks/` - webhook обработки

PostgreSQL база данных

Основные таблицы:

- `user` - пользователи системы
- `lead` - лиды (потенциальные клиенты)
- `project` - проекты
- `consultation_request` - запросы на консультацию
- `order` - заказы
- `service` - услуги

Связи:

- Lead → Project (конвертация лида в проект)
- User → Lead (назначение ответственного)
- Project → Order (создание заказа по проекту)

Redis кэширование

Использование:

- Кэширование сессий пользователей
- Rate limiting для API

- Очереди Celery задач
- Временное хранение данных форм

Конфигурация:

```
Host: localhost:6379
Password: baimuras2025
Databases:
  0 - Celery broker
  1 - Rate limiting
  2 - Sessions
```

n8n автоматизация

Доступ:

- URL: `http://95.140.153.181:5678`
- Логин: `admin@baimuras.space`
- Пароль: `Admin123!`

Автоматические процессы:

1. **Новый лид** → Уведомление менеджеру → Создание задачи
2. **Консультация** → Напоминание клиенту → Подготовка материалов
3. **Проект завершен** → Отправка счета → Запрос отзыва
4. **Еженедельные отчеты** → Сбор данных → Отправка руководству

Webhook эндпоинты:

- `/webhooks/n8n` - основной webhook
- `/webhooks/lead-automation` - автоматизация лидов
- `/webhooks/consultation-reminder` - напоминания
- `/webhooks/project-status` - обновления проектов

Celery фоновые задачи

Типы задач:

- `email_tasks.py` - отправка email
- `analytics_tasks.py` - сбор аналитики
- `backup_tasks.py` - резервное копирование
- `notification_tasks.py` - уведомления

Запуск:

```
# Worker
celery -A automation.celery_app worker --loglevel=info

# Scheduler
celery -A automation.celery_app beat --loglevel=info
```

Процесс сбора лидов

Источники лидов

1. Контактная форма на сайте (`/contact`)

Поля формы:

- Имя (обязательно)
- Телефон (обязательно)
- Email (опционально)
- Тип услуги (обязательно)
- Сообщение (обязательно)

2. API эндпоинт (/api/v1/leads)

```
POST /api/v1/leads
{
  "name": "Анна Смирнова",
  "email": "anna@example.com",
  "phone": "+7 777 111 22 33",
  "furniture_type": "kitchen",
  "budget_range": "500000-800000",
  "description": "Нужна кухня в современном стиле",
  "source": "website"
}
```

3. Консультации (/api/consultation)

```
POST /api/consultation
{
  "name": "Мария Петрова",
  "phone": "+7 777 999 88 77",
  "service_type": "children_room",
  "consultation_type": "home_visit",
  "preferred_date": "2025-06-25",
  "message": "Нужна детская комната Монтессори"
}
```

Обработка лидов

1. Валидация данных

```
# B src/routes/main_routes.py
required_fields = ['name', 'phone', 'service_type', 'message']
for field in required_fields:
    if not form_data.get(field, '').strip():
        flash(f"Поле '{field}' обязательно для заполнения", "error")
```

2. Сохранение в базу данных

```
# Создание лида
lead = Lead(
    name=data.get("name"),
    email=data.get("email"),
    phone=data.get("phone"),
    furniture_type=data.get("furniture_type"),
    budget_range=data.get("budget_range"),
    description=data.get("description"),
    status=LeadStatus.NEW,
    source=data.get("source", "website")
)
db.session.add(lead)
db.session.commit()
```

3. Автоматизация

```
# Отправка в n8n для автоматизации
lead_data = {
    'id': lead.id,
    'name': lead.name,
    'email': lead.email,
    'phone': lead.phone,
    'service_type': lead.service_type,
    'created_at': lead.created_at.isoformat()
}
send_lead_to_automation(lead_data)

# Уведомление менеджеров
manager_emails = ['manager@baimuras.space']
send_lead_notification.delay(lead_data, manager_emails)
```

Статусы лидов

```
class LeadStatus(Enum):
    NEW = "new" # Новый лид
    CONTACTED = "contacted" # Связались с клиентом
    QUALIFIED = "qualified" # Квалифицированный лид
    PROPOSAL = "proposal" # Отправлено предложение
    NEGOTIATION = "negotiation" # Переговоры
    CONVERTED = "converted" # Конвертирован в проект
    LOST = "lost" # Потерянный лид
```

Куда попадают данные

1. База данных PostgreSQL

- Таблица `lead` - основная информация
- Таблица `consultation_request` - запросы на консультацию

2. n8n автоматизация

- Webhook `/webhooks/lead-automation`
- Создание задач для менеджеров
- Отправка уведомлений

3. Email уведомления

- Менеджерам: новый лид
- Клиенту: подтверждение получения заявки
- Руководству: еженедельная сводка

4. CRM система

- Админ-панель /dashboard/leads
- Возможность назначения ответственного
- Отслеживание статуса и прогресса

Функциональность админ-панели

Доступ к админ-панели

URL: http://localhost:5000/dashboard (локально)

Продакшн: https://hub.baimuras.space

Учетные данные:

- Email: admin@baimuras.space
- Пароль: admin123

Структура админ-панели

1. Главная панель (/dashboard)

Основные метрики:

- Общее количество лидов за месяц
- Конверсия лидов в проекты (%)
- Активные проекты в работе
- Выручка за месяц (₹)
- Средний чек проекта

Быстрые действия:

- Создать новый лид
- Добавить проект
- Запланировать консультацию
- Отправить email рассылку

Последние активности:

- Новые лиды за 24 часа
- Обновления проектов
- Предстоящие консультации
- Системные уведомления

2. Управление лидами (/dashboard/leads)

Функции:

- Просмотр всех лидов с фильтрацией
- Изменение статуса лида
- Назначение ответственного менеджера
- Добавление заметок и комментариев
- Конвертация лида в проект
- Планирование звонков и встреч

Фильтры:

```
# По статусу
LeadStatus.NEW, CONTACTED, QUALIFIED, CONVERTED, LOST

# По источнику
website, social_media, referral, advertising, direct

# По типу мебели
FurnitureType.KITCHEN, BEDROOM, CHILDREN_ROOM, MONTESSORI
```

Карточка лида:

```

[ ] Анна Смирнова [ ]
[ ] anna@example.com [ ]
[ ] +7 777 111 22 33 [ ]
[ ] [ ] [ ]
[ ] Источник: Сайт [ ]
[ ] Интерес: Дизайн кухни [ ]
[ ] Статус: Новый [ ]
[ ] Дата: 21.06.2025 09:00 [ ]
[ ] [ ] [ ]
[ ] Заметки: Интересуется современным [ ]
[ ] стилем, бюджет до 800,000 [ ]
[ ] [ ] [ ]
[ ] [Изменить статус] [Конвертировать] [ ]
[ ] [ ] [ ]

```

3. Управление проектами (/dashboard/projects)**Статусы проектов:**

```
class ProjectStatus(Enum):
    CONSULTATION = "consultation" # Консультация
    MEASUREMENT = "measurement" # Замер
    DESIGN = "design" # Дизайн
    APPROVAL = "approval" # Согласование
    PRODUCTION = "production" # Производство
    INSTALLATION = "installation" # Монтаж
    COMPLETED = "completed" # Завершен
```

Карточка проекта:

Кухня для семьи Ивановых

Клиент: Иван Иванов

+7 777 123 45 67

Тип: Дизайн и изготовление кухни

Бюджет: 800,000 ₹

Статус: В производстве

Прогресс: 65%

Дедлайн: 15.08.2025

Ответственный: Менеджер Петров

[Обновить]

[Файлы]

[Финансы]

Управление проектом:

- Обновление статуса и прогресса
- Загрузка файлов (чертежи, фото)
- Планирование задач и этапов
- Учет времени и ресурсов
- Коммуникация с клиентом
- Финансовый учет

4. Аналитика (/dashboard/analytics)

Дашборд аналитики:

- График конверсии лидов по месяцам
- Источники лидов (круговая диаграмма)
- Воронка продаж по этапам
- Выручка по проектам (столбчатая диаграмма)
- Производительность менеджеров

Отчеты:

- Отчет по лидам за период
- Отчет по проектам и статусам
- Финансовый отчет по выручке
- Отчет по конверсии источников
- Отчет по производительности команды

Экспорт данных:

- Excel (.xlsx) - детальный анализ
- PDF - презентации
- CSV - импорт в другие системы

5. Управление пользователями (/dashboard/users)

Роли пользователей:

```
class UserRole:
    ADMIN = "admin"      # Полный доступ
    MANAGER = "manager"  # CRM, проекты, отчеты
    USER = "user"        # Просмотр назначенных проектов
```

Функции:

- Создание новых пользователей
- Назначение ролей и разрешений
- Управление доступом к разделам
- Блокировка/разблокировка аккаунтов

6. Консультации (/dashboard/consultations)**Календарь консультаций:**

- Месячный вид с событиями
- Недельный вид с деталями
- Дневной вид с расписанием

Планирование консультации:

Клиент: Мария Петрова Тип: Первичная консультация Дата: 25.06.2025 Время: 14:00 - 15:30 Место: Офис / Онлайн Ответственный: Дизайнер Сидоров <hr/> Заметки: Обсуждение дизайна детской комнаты Монтессори <hr/> [Подтвердить] [Перенести] [Отменить]
--

Автоматические напоминания:

- Email уведомления за 24 часа
- SMS напоминания за 2 часа
- Push уведомления в админ-панели

7. Настройки системы (/dashboard/settings)**Общие настройки:**

- Название компании и контакты
- Логотип и брендинг
- Часовой пояс и локализация
- Валюта по умолчанию

Email настройки:

- SMTP сервер конфигурация
- Шаблоны писем для автоматизации
- Подписи и контактная информация

Интеграции:

- n8n автоматизация - настройка webhook
- CRM системы - синхронизация данных
- Платежные системы - прием оплаты
- Аналитика - Google Analytics, Яндекс.Метрика

Редактирование контента

Управление страницами

Основные страницы:

- `index.html` - главная страница
- `about.html` - о компании
- `services.html` - услуги
- `portfolio.html` - портфолио
- `contact.html` - контакты

Специализированные страницы услуг:

- `services_kitchens.html` - кухни
- `services_children_rooms.html` - детские комнаты
- `services_montessori.html` - Монтессори мебель
- `services_custom_furniture.html` - мебель на заказ
- `services_design_bureau.html` - дизайн-бюро
- `services_academy.html` - академия

Управление блогом

Структура блога:

- `blog.html` - список статей
- Динамическое создание статей через админ-панель
- Категории и теги
- SEO оптимизация статей

Функции редактирования:

- WYSIWYG редактор для контента
- Загрузка изображений
- Настройка meta-тегов
- Планирование публикации

Управление портфолио

Проекты портфолио:

- Загрузка фотографий проектов
- Описание проектов
- Категоризация по типам мебели
- Lazy loading для оптимизации

Галерея изображений:

- Массовая загрузка фотографий
- Автоматическое сжатие изображений
- Водяные знаки
- Alt-теги для SEO

Типовые сценарии использования

Сценарий 1: Обработка нового лида

Шаги:

1. **Получение лида** - клиент заполняет форму на сайте

2. **Автоматическое уведомление** - менеджер получает email
3. **Назначение ответственного** - в админ-панели назначается менеджер
4. **Первичный контакт** - звонок клиенту в течение 1 часа
5. **Квалификация** - определение потребностей и бюджета
6. **Планирование консультации** - назначение встречи
7. **Конвертация в проект** - при положительном решении

Автоматизация:

Новый лид → n8n webhook → Создание задачи →
Email менеджеру → SMS клиенту → Напоминание через 1 час

Сценарий 2: Управление проектом

Этапы проекта:

1. **Консультация** (10% прогресса)
 - Встреча с клиентом
 - Обсуждение требований
 - Предварительная смета
1. **Замер** (20% прогресса)
 - Выезд замерщика
 - Точные размеры
 - Фотофиксация
2. **Дизайн** (40% прогресса)
 - Создание 3D модели
 - Подбор материалов
 - Согласование с клиентом
3. **Согласование** (50% прогресса)
 - Финальное утверждение
 - Подписание договора
 - Внесение предоплаты
4. **Производство** (80% прогресса)
 - Изготовление мебели
 - Контроль качества
 - Подготовка к доставке
5. **Монтаж** (95% прогресса)
 - Доставка и установка
 - Проверка работы
 - Обучение клиента
6. **Завершение** (100% прогресса)
 - Подписание акта
 - Окончательный расчет
 - Запрос отзыва

Сценарий 3: Еженедельная отчетность

Автоматический процесс:

1. Сбор данных (каждый понедельник 09:00)

- Новые лиды за неделю
- Конвертированные проекты
- Выручка и прибыль
- Производительность менеджеров

1. Формирование отчета

- Создание PDF отчета
- Графики и диаграммы
- Сравнение с предыдущей неделей

2. Рассылка отчета

- Email руководству
- Уведомление в Telegram
- Сохранение в архив

Сценарий 4: Управление консультациями

Процесс планирования:

1. **Запрос консультации** - клиент оставляет заявку
2. **Обработка заявки** - менеджер связывается с клиентом
3. **Назначение времени** - согласование удобного времени
4. **Подтверждение** - отправка календарного приглашения
5. **Напоминания** - автоматические уведомления
6. **Проведение консультации** - встреча с клиентом
7. **Фиксация результатов** - заметки и следующие шаги

Автоматизация напоминаний:

```
3а 24 часа: Email клиенту + SMS
3а 2 часа: SMS напоминание
3а 30 минут: Push уведомление консультанту
```

Сценарий 5: Резервное копирование

Автоматический backup:

1. База данных - каждые 6 часов

```
bash
```

```
pg_dump baimuras_db > backup_$(date +%Y%m%d_%H%M%S).sql
```

1. Файлы проектов - ежедневно в 02:00

```
bash
```

```
tar -czf uploads_backup_$(date +%Y%m%d).tar.gz uploads/
```

2. Конфигурация - еженедельно

```
bash
```

```
tar -czf config_backup_$(date +%Y%m%d).tar.gz .env nginx/ docker-compose.yml
```

3. Очистка старых backup - хранение 30 дней

```
bash
```

```
find /backups -name "*.sql" -mtime +30 -delete
```

Техническое обслуживание

Мониторинг системы

Health Check эндпоинты:

- GET /health - статус приложения
- GET /api/health - статус API
- GET /webhooks/health - статус webhook системы

Системные метрики:

```
# Использование ресурсов
docker stats

# Логи приложения
docker-compose logs -f web

# Статус сервисов
docker-compose ps
```

Логирование

Файлы логов:

- logs/baimuras.log - основные логи приложения
- logs/error.log - ошибки
- logs/celery.log - логи фоновых задач
- /var/log/nginx/baimuras_access.log - логи Nginx

Уровни логирования:

- DEBUG - детальная отладочная информация
- INFO - общая информация о работе
- WARNING - предупреждения
- ERROR - ошибки
- CRITICAL - критические ошибки

Обновление системы

Процедура обновления:

1. Создание backup

```
bash
./scripts/backup.sh
```

1. Остановка сервисов

```
bash
docker-compose down
```

2. Обновление кода

```
bash
git pull origin main
```

3. Обновление зависимостей

```
bash
pip install -r requirements.txt
```

4. Миграции базы данных

```
bash
flask db upgrade
```

5. Запуск сервисов

```
bash
docker-compose up -d
```

6. Проверка работоспособности

```
bash
curl http://localhost/health
```

Безопасность

Регулярные проверки:

- Обновление зависимостей
- Сканирование уязвимостей
- Проверка логов на подозрительную активность
- Ротация паролей и ключей

Команды для проверки:

```
# Проверка уязвимостей Python пакетов
pip audit

# Статический анализ кода
pylint src/

# Сканирование безопасности
bandit -r src/
```

Поддержка и контакты

Техническая поддержка

Контакты:

- Email: support@baimuras.space
- Телефон: +7 777 123 45 67
- Telegram: @baimuras_support

Разработчик:

- Email: admin@baimuras.space
- GitHub: <https://github.com/ardakchapaev/baimuras.space>

Часто задаваемые вопросы

Q: Как сбросить пароль администратора?

A: Используйте команду:

```
python -c "from src.models.user import User; User.reset_admin_password()"
```


Q: Как добавить нового менеджера?

A: Перейдите в “Пользователи” → “Создать пользователя” → Роль “manager”

Q: Как настроить автоматические уведомления?

A: В “Настройки” → “Email настройки” → Включите нужные типы

Q: Как экспортировать данные?

A: В любом разделе нажмите “Экспорт” и выберите формат

Q: Как восстановить удаленный лид?

A: “Лиды” → “Корзина” → “Восстановить” (хранится 30 дней)

Полезные команды**Docker управление:**

```
# Просмотр логов
docker-compose logs -f [service_name]

# Перезапуск сервиса
docker-compose restart [service_name]

# Обновление образов
docker-compose pull && docker-compose up -d

# Очистка неиспользуемых ресурсов
docker system prune -f
```

База данных:

```
# Подключение к БД
docker-compose exec db psql -U baimuras_user -d baimuras_db

# Создание backup
docker-compose exec db pg_dump -U baimuras_user baimuras_db > backup.sql

# Восстановление из backup
docker-compose exec -T db psql -U baimuras_user baimuras_db < backup.sql
```

Celery задачи:

```
# Просмотр активных задач
celery -A automation.celery_app inspect active

# Очистка очереди
celery -A automation.celery_app purge

# Мониторинг задач
celery -A automation.celery_app flower
```

Заключение

BaiMuras Platform v2.0.0 представляет собой полнофункциональную систему управления мебельным бизнесом с современной архитектурой, надежной безопасностью и удобным интерфейсом администрирования.

Ключевые преимущества:

- Готовность к продакшн деплою
- Комплексная автоматизация процессов
- Интуитивная админ-панель
- Масштабируемая архитектура
- Высокий уровень безопасности

Рекомендации по использованию:

1. Регулярно создавайте резервные копии
2. Мониторьте системные метрики
3. Обновляйте зависимости для безопасности
4. Обучайте персонал работе с системой
5. Используйте аналитику для принятия решений

Система готова к полноценному использованию в продакшн среде и может эффективно поддерживать все бизнес-процессы мебельной компании.

Последнее обновление: 21 июня 2025 г.

Версия платформы: 2.0.0

Автор руководства: Система анализа BaiMuras Platform