

Система автоматизированного управления проектами



для BIPULSE 7.14.503 и выше

Руководство администратора
Версия 1.11

Санкт-Петербург
2008-2025

Оглавление

Система автоматизированного управления проектами.....	1
1. Что за документ.....	3
2. Функциональные характеристики.....	4
2.1. Возможности планирования.....	4
2.2. Возможности отображения.....	5
2.3. Иные возможности:.....	6
3. Руководство администратора.....	7
3.1. Развёртывание на сервер.....	7
3.2. Алгоритм развёртывания.....	7
3.3. Регулярное обслуживание.....	10
3.4. Обновление программного обеспечения и схемы базы данных.....	11
3.5. Файлы и имена.....	12
3.6. Другие команды.....	13
3.7. Авторизация.....	14
3.8. Настойка авторизации через LDAP.....	15
3.11. Настройка рабочего календаря.....	17
3.12. Настройка почтовых уведомлений.....	20
3.13. Интеграционное тестирование и безопасность.....	22
3.14. Запрос технической поддержки.....	23
3.17. Правила уровня оказания услуг.....	23
4. Архитектура решения.....	24
Техническая поддержка.....	25

1. Что за документ

- 1.1. В этом документе схема развёртывания BIPULSE на сервер Клиента при поставке коробочной версии.

2. Функциональные характеристики

2.1. Возможности планирования

- 2.1.1. Указание оценки длительности задачи в днях, оптимистично и пессимистично.
- 2.1.2. Указание оценки трудозатрат каждого ресурса, в днях, процентах, оптимистично и пессимистично.
- 2.1.3. Указание оценки объёма/сложности задачи: км, sp, куб.м. и другие.
- 2.1.4. Автоматическое планирование расписания проектов и программ проектов с выравниванием ресурсов на уровне проекта, портфеля, программы проектов.
- 2.1.5. Автоматический расчёт критической цепи проекта, программы проектов (Critical chain project management, CCPM).
- 2.1.6. Автоматический расчёт и отображение потраченного рабочего времени на задачу.
- 2.1.7. Прогнозирование сроков завершения проекта: оптимистично, реалистично, пессимистично.
- 2.1.8. Контроль сроков проекта и автоматические рекомендации по корректировке проекта.
- 2.1.9. Расчёт потребления буферов критической цепи, питающей цепи, и буфера проекта в режиме Agile.
- 2.1.10. Расчёт коэффициента точности планирования
- 2.1.11. Эшелонирование проектов
- 2.1.12. Выявление ресурса-ограничения
- 2.1.13. Планирование спринта и анализ буфера расписания спринта в много-проектной среде

2.2. Возможности отображения

- 2.2.1. Срез данных: «План работ» в детализации: программы, проекты, задачи, производственные задачи
- 2.2.2. Срез данных: «Поток» — доски текущих задач и планируемых задач и планируемых пакетов работ для двух-уровневого управления проектом.
- 2.2.3. Срез данных: «Персонал» в детализации: загрузка персонала, участвовавших в задачах, персональные доски задач.
- 2.2.4. Срез данных: Расписание в детализации: программы, проекты, задачи.
- 2.2.5. Срез данных: Ресурсный план в детализации занятости ресурсов: ресурс, занятость в проекте, занятость в программе, занятость в задаче.
- 2.2.6. Срез данных: Продуктивность в детализациях: суммарно, подробно, по типам активностей, по проектам.
- 2.2.7. Отображение моментальной аналитики по проекту и программе проектов на основе исторических данных:
 - 2.2.7.1. Диаграмма сгорания
 - 2.2.7.2. Скорости решения и добавления объема работ
 - 2.2.7.3. Кумулятивная диаграмма потока работ
 - 2.2.7.4. Анализ процесса выполнения задач
 - 2.2.7.5. Диаграмма расхода буфера критической цепи во времени

2.3. Иные возможности:

- 2.3.1. Система контроля доступа основанная на списках доступа.
- 2.3.2. Возможность организации сотрудников по подразделениям с указанием прав доступа.
- 2.3.3. Двухсторонняя синхронизация с внешними системами: Jira, YouTrack, Redmine, Trello, Bitrix24 и другими системы по согласованию с Заказчиком.
- 2.3.4. Поддержка wiki формата MediaWiki, Markdown форматирования текста описания задачи, и устава проекта
- 2.3.5. Добавления комментариев, файлов, меток к задаче.
- 2.3.6. Указание ожидаемых трудозатрат на выполнение работы.
- 2.3.7. Хранение устава программы проектов.
- 2.3.8. Хранение устава проекта.
- 2.3.9. Хранение целей и критериев приёмки проекта.
- 2.3.10. Авторизации по LDAP или MS Active Directory
- 2.3.11. Уведомление по электронной почте об изменениях в задачах.

3. Руководство администратора

3.1. Развёртывание на сервер

- 3.1.1. Развёртывание на сервер Клиента выполняется только в случае поставки «коробочной» версии Продукта.

3.2. Алгоритм развёртывания

- 3.2.1. Предусловия развёртывания приложения:
- 3.2.1.1. У вас установлена и настроена на сервере ОС Ubuntu 20.04 или ОС AstraLinux.
 - 3.2.1.2. На сервере есть 10 Гб свободного места на диске (ПЗУ, НЖМД))
 - 3.2.1.3. На сервере есть 64 Гб ОЗУ.
 - 3.2.1.4. **Работоспособность приложения проверялась с СУБД PostgreSQL версий 9.5, 12, 14. На другие версии гарантийные обязательства не распространяются.**
- 3.2.2. Если сервер «чистый» и СУБД PostgreSQL не установлен, то необходимо выполнить настройку локали:
- 3.2.2.1. Необходимо проверить установленные локали, что среди них есть ru_RU.UTF-8, если нет то выполнить следующие шаги

```
# apt install sudo locales
# locale -a
en_AU.utf8
...
POSIX
```

- 3.2.2.2. Сгенерировать новую локаль **ru_RU.UTF-8**:

```
sudo locale-gen ru_RU.UTF-8
LANG=ru_RU.UTF-8
sudo update-locale "LANG=ru_RU.UTF-8"
dpkg-reconfigure --frontend noninteractive locales
```

- 3.2.2.3. Проверить установленные локали, что среди них есть **ru_RU.UTF-8**:

```
# locale -a
```

```
en_AU.utf8
...
POSIX
ru_RU.utf8
```

3.2.3. Все следующие команды необходимо выполнять от настоящего пользователя **root**.

3.2.4. **Проверьте, что PostgreSQL установлен и работает. Иначе не выполнится начальная настройка пользователя БД и базы данных.**

3.2.5. Выполните развёртывание приложения:

3.2.5.1. Установите пакеты **bipulse-*X.Y.NNN_amd64_ubuntu20.04.deb**

```
apt install /path/to/bipulse-*X.Y.NNN_amd64_ubuntu20.04.deb
```

3.2.5.2. Измените параметры подключения в базе данных сервера на работу с PostgreSQL в файле **/etc/opt/bipulse/bipulse.ini**:

```
database_server = pgsql:localhost
database_user   = bipulse_user
database_pass   = bipulse
database_db     = bipulse
```

3.2.5.3. Выполните загрузку схемы БД:

```
/opt/bipulse/bin/bipulse-cli psqlCreateSchema localhost bipulse bipulse_user bipulse
```

3.2.5.4. Выполните инициализацию таблиц:

```
/opt/bipulse/bin/bipulse-cli initialize bipulse
```

3.2.5.5. Ваш сервер готов к использованию.

3.2.6. Войдите в систему

3.2.6.1. Откройте страницу <http://localhost:20220/>

3.2.6.2. Введите логин: **admin**

3.2.6.3. Введите пароль: **admin**

3.2.6.4. Вы администратор системы с полным доступом.

3.2.7. Перед началом работы получите стандартную или пробную

лицензию:

- 3.2.7.1. Откройте страницу: «Управление лицензией» (Меню пользователя ➤ Администрирование ➤ Управление лицензией) http://localhost:20220/?page=license_status&aspect=administration
- 3.2.7.2. Укажите параметры лицензии.
- 3.2.7.3. Нажмите кнопку **«Запросить лицензионный ключ»**.
- 3.2.7.4. Отправьте запрос лицензионного ключа по электронной почте на адрес `support@bipulse.ru`.
- 3.2.7.5. Дождитесь ответа, выпуск ключа занимает 2-3 дня.
- 3.2.7.6. Полученный лицензионный ключ введите в поле разделе «Установка лицензионного ключа»
- 3.2.7.7. Нажмите кнопку **«Применить лицензионный ключ»**.
- 3.2.8. Настройте отправку почтовых уведомлений
- 3.2.8.1. Откройте страницу «Настройки» (Меню пользователя ➤ Администрирование ➤ Настройки) <http://localhost:20220/?page=settings&aspect=settings>
- 3.2.8.2. В разделе «Настройки почтовых уведомлений» заполните параметры подключения к SMTP серверу и данные отправителя.
- 3.2.8.3. Нажмите кнопку **«Сохранить»**.
- 3.2.9. Поздравляем, теперь вы можете начать работу!
- 3.2.10. При необходимости, настройте Nginx, чтобы он принимал подключения по HTTPS. Конфигурация проксирования уже подготовлена в `/etc/nginx/sites-available/bipulse`

3.3. Регулярное обслуживание

- 3.3.1. Действий по регулярному обслуживанию не требуется. Все настройки регулярного обслуживания выполнены пакетом установки.
- 3.3.2. При подключении интеграций настройте регулярное выполнение:

```
/opt/bipulse/bin/bipulse-cli serviceProcess bipulse
```

3.4. Обновление программного обеспечения и схемы базы данных

- 3.4.1. Обновление версии выполняется по процедуре установки по процедуре 3.2.5.1.
- 3.4.2. В большей части случаев обновления схемы базы данных не требуется, однако при изменении второго номера версии (например 7.14 на 7.15) схема базы данных может обновиться. В этом случае необходимо выполнить обновление схемы базы данных вручную.
- 3.4.3. BIPULSE оснащён встроенными средствами контроля версии схемы базы данных. Но запуск утилиты обновления **НЕ ВЫПОЛНЯЕТСЯ** автоматически при установке.
- 3.4.4. Для выполнения обновления схемы базы данных выполните команду:

```
/opt/bipulse/bin/bipulse-cli updateDatabase bipulse
```

- 3.4.5. Проверьте, что команда завершилась успешно. В некоторых случаях может потребоваться ручной ввод команд для СУБД PostgreSQL (таких как CREATE EXTENSION) так как требуются повышенные привилегии администратора СУБД. Если ошибка произошла, то выполните команды администратора вручную и повторите команду обновления схемы базы данных.

3.5. Файлы и имена

- 3.5.1. Установочный каталог:
/opt/bipulse
- 3.5.2. Журналы сервера:
/var/log/bipulse
- 3.5.3. Рабочие файлы сервера, загружаемые пользовательские файлы:
/var/lib/bipulse
- 3.5.4. Файл мастер-пароля
/var/lib/bipulse/.htpasswd
- 3.5.5. Конфигурация путей в параметрах работы:
/etc/opt/bipulse/bipulse.ini
- 3.5.6. Утилита запуска сервиса:
/opt/bipulse/bin/start-server.sh
systemctl start bipulse
- 3.5.7. Интерфейс обслуживания из командной строки
/opt/bipulse/bin/bipulse-cli
- 3.5.8. Имя сервиса для systemd:
bipulse
- 3.5.9. **Сервис обслуживания почтовых уведомлений**
 - 3.5.9.1. Имя сервиса для systemd:
bipulse-mail
 - 3.5.9.2. Утилита запуска сервиса:
/opt/bipulse/bin/start-mailSender.sh
systemctl start bipulse-mail

3.6. Другие команды

- 3.6.1. Создание пользователя в БД для работы BIPULSE (если не создан при установке пакета):

```
sudo -u postgres /opt/bipulse/bin/psqlCreateUser
```

- 3.6.2. Создание БД в СУБД для работы BIPULSE (если не создана при установке пакета)

```
sudo -u postgres /opt/bipulse/bin/psqlCreateDatabase
```

- 3.6.3. Установка возможностей пользователя `bipulse` для нормальной работы BIPULSE на AstraLinux

```
sudo usercaps -m PARSEC_CAP_IPC_OWNER bipulse
```

3.7. Авторизация

- 3.7.1. Обычный вход в систему выполняется с логином (имя учётной записи) и паролем.
- 3.7.2. Если пароль в учётной записи не задан, то такая учётная запись блокируется.
- 3.7.3. BIPULSE поддерживает режим входа Администратора (первый пользователь с id=1) через мастер-пароль. Такой пароль задаётся в файле `/var/lib/bipulse/.htpasswd` в стандартном формате Apache2 с режимом хеширования sha1. Таким образом, у разных администраторов может быть свой пароль заданный через файл. Сохранение конфиденциальности и недоступности файла должно обеспечиваться системным администратором сервера.

3.8. Настойка авторизации через LDAP

- 3.8.1. Для настройки авторизации через LDAP необходимо:
- 3.8.1.1. Установить в систему пакет **ldap-utils (sudo apt install ldap-utils)**
 - 3.8.1.2. На странице «Интеграции» (Меню пользователя ➤ Администрирование ➤ Интеграции) нажать кнопку «Подключить интеграцию»
 - 3.8.1.3. Заполните поля настройки интеграции:
 - 1. «Тип трекера» (Тип системы) : LDAP
 - 2. Синхронизация: импорт
 - 3. Репликация: запрещена
 - 4. Адрес LDAP сервера в формате: ldap://server.com:port
 - 5. В параметрах поиска авторизации укажите строку поиска пользователя по логину. USERNAME — это маска подстановки логина пользователя.
 - 6. Укажите правила импорта полей для импорта учётной записи пользователя.
 - 7. Заполните остальные поля.
 - 3.8.1.4. Нажмите кнопку «Сохранить». Теперь мы можем выбрать эту систему для авторизации.
 - 3.8.2. Для выбора системы для авторизации откройте страницу Настройки (Меню пользователя ➤ Настройки)
 - 3.8.3. В разделе «Общие настройки ➤ Правила авторизации» выберите «Способ авторизации» по имени системы из раздела подключения интеграции.
 - 3.8.4. Для автоматического импорта учётной карточки сотрудника выберите режим «Автоматически регистрировать при успешной

проверке пароля». Иначе BIPULSE будет проверять пароль, но не пустит в систему так как будет отсутствовать учётная карточка сотрудника с указанным логином.

3.8.5. Нажмите кнопку «Сохранить» для сохранения настроек.

3.8.6. ВАЖНО!!! Откройте браузер в анонимном режиме, и проверьте вход с учётной записью администратора. Она теперь также проверяется внешней системой (LDAP).

3.9. В случае неуспешной авторизации и утрате административного доступа используйте «режим входа Администратора».

3.10. Примечания:

3.10.1. Учётные карточки пользователей создаются в режиме использования «Ресурс», то есть без доступа к возможностям системы.

3.11. Настройка рабочего календаря

- 3.11.1. Рабочий календарь импортируется из сервиса <https://production-calendar.ru> , для использования нужно получить API токен на сервисе.
- 3.11.2. Все файлы настройки рабочего календаря размещаются в пути: `/var/lib/bipulse/productionCalendar/`
- 3.11.3. Токен для получения рабочего календаря размещается в файле **production-calendar.token**
- 3.11.4. Рабочий календарь может быть загружен и размещён вручную в файлах вида: **calendar.ru.<год>.json** , где:
«год» номер года.
«ru» код страны.
- 3.11.5. При загрузке файла вручную, проверьте что права на чтение доступны на пользователя **bipulse.bipulse**. После загрузки календаря необходимо перезапустить сервис (**systemctl stop bipulse; systemctl start bipulse**).
- 3.11.6. Формат файла соответствует сервису **production-calendar.ru**.
- 3.11.7. Структура файла календаря (авторство сайта <https://production-calendar.ru/manual>):

```
{
  "status": "ok"
  "country_code": "ru",
  "country_text": "Российская Федерация",
  "region_id": 26,
  "region_text": "Ставропольский край",
  "dt_start": "01.01.2025",
  "dt_end": "15.01.2025",
  "work_week_type": "5-и дневная рабочая неделя",
  "period": "Произвольный период",
  "statistic":
    {
      "calendar_days": 15,
      "calendar_days_without_holidays": 7,
      "work_days": 5,
      "weekends": 2,
      "holidays": 8,
      "shortened_working_days": 0,
      "working_hours": 40
    }
}
```

```
    }
  },
  "days": [
    {
      "date": "01.01.2025",
      "type_id": 3,
      "type_text": "Государственный праздник",
      "note": "Новогодние каникулы",
      "week_day": "вс",
      "working_hours": 0
    },
    ...
    {
      "date": "08.01.2025",
      "type_id": 2,
      "type_text": "Выходной день",
      "week_day": "вс",
      "working_hours": 0
    },
    ...
    {
      "date": "15.01.2025",
      "type_id": 2,
      "type_text": "Выходной день",
      "week_day": "вс",
      "working_hours": 0
    }
  ]
}
```

Основные свойства главного объекта (country_code, country_text, region_id, region_text, dt_start, dt_end, work_week_type) интуитивно понятны и не требуют отдельного описания.

Свойство days представляет из себя массив объектов которые описывают календарные сутки. Каждый элемент массива содержит объект со следующими свойствами:

- type_id и type_text определяют тип суток:

id	Описание
1	Рабочий день
2	Выходной день
3	Государственный праздник
4	Региональный праздник
5	Предпраздничный сокращенный рабочий день
6	Дополнительный / перенесенный выходной день

- `week_day` – сокращенное наименование дня недели;
- `working_hours` – количество рабочих часов в данных сутках для 40-й часовой рабочей недели;
- `is_project` – наличие данного свойства равным 1 (true) означает что указанные сутки в производственном календаре находится в проект закона и еще не утверждена правительством;
- `is_wschr` - наличие свойства равным 1, означает что день является так называемый выходным днем с сохранением заработной платы.

Свойство `statistic` отображает ряд статистических данных для задаваемого периода:

- `calendar_days` – количество календарных дней в периоде;
- `calendar_days_without_holidays` - количество календарных дней в периоде без учета праздничных дней (полезный показатель для расчета продолжительности отпуска работника);
- `work_days` – количество рабочих дней в периоде;
- `weekends` – количество выходных дней в периоде (без учета праздничных);
- `holidays` – количество праздничных дней в периоде;
- `shortened_working_days` – количество сокращенных рабочих дней в периоде;
- `working_hours` – количество рабочего времени за период.

3.12. Настройка почтовых уведомлений

- 3.12.1. BIPULSE поддерживает почтовые уведомления о действиях с задачами: описание задачи обновлено, отчёт задачи обновлён, задача запланирована, запущена, завершена, комментарий добавлен.
- 3.12.2. BIPULSE не поддерживает отправку писем через sendmail сервера, необходимо, чтобы был выдан конкретный пользователь на почтовом сервере и указаны реквизиты подключения.
 - 3.12.2.1. Реквизиты подключения указываются в разделе «Настройки почтовых уведомлений» в Настройках (Администрирование ➤ Настройки ➤ Настройки почтовых уведомлений)
- 3.12.3. Заполнение полей «Настройки SMTP»:
 - 3.12.3.1. Адрес сервера — адрес SMTP сервера.
 - 3.12.3.2. Порт сервера - SMTP порт, обычно 25, если SSL то 465.
 - 3.12.3.3. Шифрование — режим шифрования, none — нет, ssl — SSL, tls — TLS.
 - 3.12.3.4. Логин авторизации — логин авторизации на сервере SMTP.
 - 3.12.3.5. Пароль авторизации — пароль авторизации на сервере SMTP.
- 3.12.4. Заполнение полей «Параметры отправителя»:
 - 3.12.4.1. Имя отправителя - Имя отправителя уведомлений (например «Пульс»)
 - 3.12.4.2. Адрес электронной почты отправителя - Электронный адрес отправителя уведомлений (например «pulse@company.com»)
- 3.12.5. Отправка почтовых уведомлений выполняется при условиях:
 - 3.12.5.1. В Настройках (страница Настройки) указаны реквизиты подключения к SMTP серверу для пользователя - отправителя уведомлений.
 - 3.12.5.2. Модули расширений отправки уведомлений разрешены.
 - 3.12.5.3. Пакет bipulse-mail установлен корректно и сервис обслуживания очереди отправки работает.

- 3.12.5.4. В карточке сотрудника указан адрес электронной почты. (Если адрес не указан, то уведомления приходить не будут)
- 3.12.6. Почтовые уведомления отправляются в случае если:
 - 3.12.6.1. Задача поставлена в план работ (запланирована) сотруднику.
 - 3.12.6.2. Задача начата.
 - 3.12.6.3. Задача завершена.
 - 3.12.6.4. Комментарий добавлен.
 - 3.12.6.5. Задача обновлена (отчет, описание).
- 3.12.7. Почтовые уведомления **НЕ отправляются** в случае если:
 - 3.12.7.1. Уведомление предназначено автору изменений. Он уже знает, что сделал.
- 3.12.8. Кто получает уведомления:
 - 3.12.8.1. Второй участник задачи,
 - 3.12.8.2. Все кто указал "Следить" в карточке задачи,
 - 3.12.8.3. Все участвовал в переписке в комментариях к задаче.
 - 3.12.8.4. Те кто указан через в комментариях в через текст вида: @login.
- 3.12.9. Как проверить, что уведомления отправляются:
 - 3.12.9.1. Создать задачу на исполнителя у которого указана почта.
 - 3.12.9.2. Запланировать задачу.
 - 3.12.9.3. Проверить что исполнитель получил уведомление.

3.13. Интеграционное тестирование и безопасность

- 3.13.1. Приложение **оснащено** встроенным модулем поддержки интеграционного тестирования. Его можно использовать для автоматизированных проверок разработанных модулей расширений.
- 3.13.2. Модуль поддержки интеграционного тестирования должен **быть запрещён** на основном сервере.
- 3.13.3. **Избегайте** использования доменных имён для сервера **dev*.domain.tld, testing*.domain.tld**. Эти имена зарезервированы для интеграционного тестирования.
- 3.13.4. Доступ к консоли интеграционного тестирования активируется после создания файла `/var/lib/bipulse/testing-enable` . **НИ В КОЕМ СЛУЧАЕ НЕ СОЗДАВАЙТЕ И НЕ ОСТАВЛЯЙТЕ ЭТОТ ФАЙЛ НА ОСНОВНОМ СЕРВЕРЕ.**
- 3.13.5. В конфигурации Nginx используйте только именованные сервера.

3.14. Запрос технической поддержки

- 3.15. Написать письмо на электронную почту **support@bipulse.ru**, в котором:
- 3.15.1. Описать суть проблемы.
 - 3.15.2. Описать в чем проявляется проблема.
 - 3.15.3. Приложить снимки экрана.
 - 3.15.4. Приложить снимки экрана консоли Network отладчика броузера
 - 3.15.5. Приложить лог-файлы (журналы) сервера.
- 3.16. При необходимости, предоставить удалённый доступ к серверу по SSH.

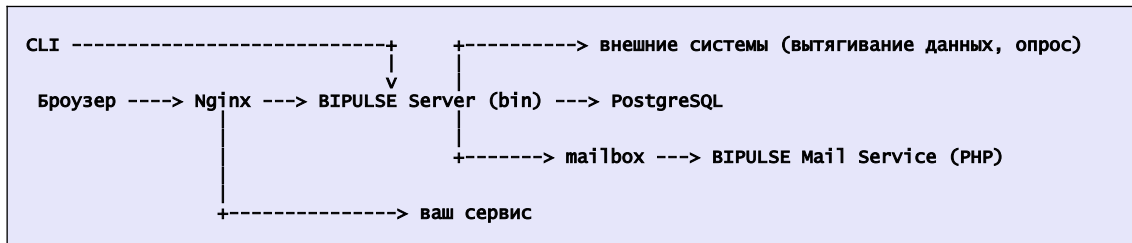
3.17. Правила уровня оказания услуг

- 3.1. Ответ Клиенту по запросу предоставляется в течении **трёх рабочих дней**. Срок реализации запроса в соответствии с соглашением об уровне оказания услуг являющимся приложением к договору.
- 3.2. Срок исполнения гарантийных обязательств по исправлению дефектов — **20 рабочих дней** по стандартному договору.

4. Архитектура решения

Архитектура решения и возможности адаптации.

Архитектура решения простая:



Технологический стек в составе поставки:

- PostgreSQL - СУБД.
- PHP для утилит (BIPULSE Mail Service)
- Java (OpenJDK) - для чтения MSProject и Oracle Primavera форматов.
- JavaScript/React/JSX - фронтенд (клиентская часть).
- бинарный код - BIPULSE Server, поставляется в закрытом формате.

Возможности разработки расширений:

- Своих модули расширений фронтенда (клиентской части) пишутся на языке JavaScript (React/JSX). Модуль может обработать точки расширений или перехватить системные шаблоны. Для обработки серверных событий необходимо обработать события от сервера (Webhook).

Частые вопросы:

Как сделать доступ по HTTPS?

Ответ: настройте Nginx, конфигурация поставляется в комплекте. Настройку выполняет ваш системный администратор.

Как настроить резервирование?

Ответ: Резервирование выполняется средствами СУБД. Настройте зеркалирование СУБД.

Где хранятся файлы?

Ответ: В СУБД.

Я могу подключить свою отчетность?

Ответ: да, через прямой доступ в СУБД или через REST API.

Как получить описание всех вызовов REST API?

Ответ: Предоставляется по запросу. Или исследуйте общение броузера с сервером, структура запросов простая в JSON формате.

Техническая поддержка

Техническая поддержка

support@bipulse.ru

+7 (812) 408-19-78

