

# UnicoreCMS

Документация



# Table of contents:

- Введение
  - Преимущества
  - Экосистема
  - TechStack
  - Покупая CMS вы получаете
    - Интегрированные компоненты
    - Внешние компоненты
    - API для разработчиков
- Установка
  - Docker-установка
    - Подготовка сервера
    - Базовая настройка
    - Развёртывание контейнеров
    - Синхронизация БД
    - Заполнение БД (не обязательно)
  - Manual-установка
    - Системные требования
    - Сборка UnicoreCMS
    - Базовая настройка
    - Синхронизация БД
    - Заполнение БД (не обязательно)
- Подключение к БД
  - При использования Docker
  - При использования Manual
    - Все настройки подключения
    - Установка драйвера базы данных
- Отправка писем
  - Настройка
    - Enviroment-переменные
    - Использование пресета
    - Ручная настройка подключения
- Recaptcha
  - Настройка
    - Производим настройку .env.
- VKontakte LongPoll
- Переменные Enviroment

- UnicoreConnect
  - Установка и настройка
  - Сборка
  - Команды и права
    - Экономика
    - PlayTime
    - Склад и магазин
    - Администрирование
- UnicoreProvider
  - Установка и настройка
  - Сборка
- Статистика
- Пользователи
- Роли и разрешения
  - Обязательные роли
  - Добавление роли
    - Формирование разрешений
  - Список разрешений
    - Основной сайт
    - Панель управление
- Новости
- Статические страницы
  - Добавление страницы
- Email-сообщения
  - Редактирование сообщений
- Серверы и моды
  - Добавление сервера
  - Добавление мода
- Донат группы и права
  - Создание донат-группы
  - Создание донат-права
  - Создание периода
  - Создание донат-кита
- Магазин
  - Создание товаров
  - Создание китов
  - Создание категорий
- Платежи

- Промо-коды
- Голсование
- Расширение OAuth
- Изменение клиента
- Ивенты
- Разрешения
- Rest-API
- Вебхуки
- WebSocket-API



# Введение

**UnicoreCMS** - Headless CMS для модовых проектов Minecraft. Современные технологии и профессиональный подход, невероятная производительность и скорость.

## Преимущества

- **Современная.** Построен по последним технологиям.
- **Надёжная.** Проверена в производственной среде.
- **Быстрая.** Каждый компонент оптимизирован и обрабатывает сверх-много запросов за сверх-мало времени.
- **Безопасная.** На ряду с актуальными версиями пакетов и современным концептом работы — неуязвима.
- **Не зависима от БД.** Поддерживает MySQL, MariaDB, MongoDB, Postgres, CockroachDB, SQLite, Microsoft SQL Server, Oracle или SAP Hana.
- **Простая установка.** Docker или Manual? В обоих случаях установка довольно проста и описана [тут](#).
- **Легко интегрируема.** Свои решения для простой интеграции с лаунчером и сервером.
- **OpenSource.** Нам нечего скрывать и вы можете это увидеть

## Экосистема

Админ-панель, клиент и сервер разделены на 3 независимых приложения. Система может работать, как одно целое или, как вам угодно.

## TechStack

- Клиент и админ-панель разработаны на Nuxt.JS, PrimeVue и Vuesax.
- Сервер разработан на NestJS, Fastify и TypeORM.

- Uniconnect и UniconnectProvider разработаны на Kotlin

# Покупая CMS вы получаете

## Интегрированные компоненты

- 👤 Система учётных записей пользователей.
- 👤 **Публичные** профили пользователей.
- 📁 Загрузка/Удаление скинов/плащей.
- 👑 Система **ролей и прав** пользователей, принцип работы схож с PEX/LuckPerms. Встроенные группы для default и banned (например можно запретить перевод денег заблокированным на серверах пользователям)
- 🔑 Авторизация на основе JWT- токенов (Access/Refresh).
- 📄 Система сеансов, история авторизаций.
- 📱 **Двухфакторная авторизация** (GoogleAuth).
- 📧 Смена/Восстановление пароля подтверждение Email.
- ⚙️ **Админ-Панель**, охватывающая весь функционал системы (далее ПУ)
- 📊 **Статистические страницы**, автоматически генерируемые **статистические страницы для описания серверов и донат-групп**.
- 🗒️ **Логирование действий** в ЛК, Магазине, ПУ.
- 📊 Сбор внутриигровой статистики PlayTime (мультисерверно).
- 📊 **Real-Time мониторинг** на веб-сокетах.
- 💡 **Новости**. Парсинг новостей с помощью **VK Longpoll**.
- 💎 **E-Commerce**
  - **Полная мультисерверность** двух следующих компонентов, **пересечения между серверами**.
  - Продажа внутриигровых **товаров и китов**, группирование по **категориям**.
  - Продажа внутриигровых **прав/групп**
  - Продажа **веб-прав** (Например возможность загрузки HD-скина и т. д.)
  - Конструктор **вознаграждений при голосовании** в мониторингах (TopCraft, MCTop, Minecraft-Raiting).
  - Конструктор на основе триггеров для топ-листов (Топ "онлайн" и топ голосующих).

- **Внутриигровая экономика** (мультисерверно). Перевод валюты между серверами, игроками, обмен валюты.

## Внешние компоненты

- **UnicoreConnect** - **Sponge/Spigot-плагин** для двухнаправленной интеграции экономики, прав, групп, склада-покупок, банов и сбора статистики между сайтом и серверами. [Подробнее.](#)
- **UnicoreProvider** - серверный модуль **GravitLauncher** для интеграции сервисов авторизации. [Подробнее.](#)

## API для разработчиков

Поверх возможностей NestJS/NuxtJS реализованный свои методы взаимодействия, которые вы можете использовать при расширении функционала системы, как внутри приложения, так и из вне. Например:

- WS/Rest-API может предоставлять аутификацию и права на основе встроенной системы **API-ключей**.
- Внутренняя **система ивентов**.
- Система веб-хуков (Json/Discord)

Более подробно можно ознакомиться в разделе Разработчикам.



# Установка

Установка UnicoreCMS производится на VDS/Dedicated-сервер, либо локальный компьютер. Обратите внимания, что shared-хостинги не подойдут для этой роли.

## Docker-установка

Этот метод прекрасен, ведь его особенность заключается в том, что вам практически ничего не нужно делать самостоятельно.

### ! INFO

**Docker** - программное обеспечение для автоматизации развёртывания и управления приложениями в средах с поддержкой контейнеризации, контейнеризатор приложений. Простыми словами Docker позволяет нам развернуть всё, что так необходимо для работы UnicoreCMS одной командой.

## Подготовка сервера

Установка Docker и Docker Compose.

- Установка Docker: [CentOS](#), [Fedora](#), [RHEL](#), [SLES](#), [Ubuntu](#), [Windows](#), [MacOS](#)
- [Установка Docker Compose](#)

## Базовая настройка

### ! НАСТРОЙКА NGINX-СЕРВЕРА

Только если вы используете встроенный nginx-сервер предоставляемый аргументом `--profile nginx`



```
CLIENT_HOST= # Хост клиента (например: example.com) [string]
ADMIN_HOST= # Хост админ-панели (например: admin.example.com) [string]
SERVER_HOST= # Хост сервера (например: api.example.com) [string]
```

**Обратите внимание, что вам предварительно необходимо подключить домен и поддомены исключительно к вашему VDS/Dedicated-серверу, либо настроить vhost.**

Переменные `CLIENT_PORT`, `ADMIN_PORT`, `SERVER_PORT` передаются в Nginx-сервер автоматически, если вы их измените и перезапустите контейнеры, но в этом нет необходимости, ведь они закрыты и доступны только внутри контейнера.

- Выполните настройку [Подключения к БД](#).
- Выполните настройку [Отправки писем](#).
- Выполните настройку [Recaptcha](#).
- Укажите `BASEURL` и `API_BASEURL` из раздела [Переменные Enviroment](#)

## Развёртывание контейнеров

Пропишите в терминале команду:

**⚠ ВОЗМОЖНО У ВАС УЖЕ ЕСТЬ УСТАНОВЛЕННЫЙ NGINX НА СЕРВЕРЕ?**

В этом случае стоит отключить Nginx-сервер предоставляемый UnicoreCMS, убрав аргумент `--profile nginx` во всех командах, где он используется.

```
docker-compose --profile nginx --profile prod up -d
# Ждем окончания операции и запуска контейнеров
```

## Синхронизация БД

UnicoreCMS не использует подготовленные миграции, а генерирует их "на лету" под **используемую базу данных**, в случае *Docker Установки* - это **MariaDB**.

```
docker exec -t -i unicorecms npm run sync

# Останавливаем контейнеры
```

```
docker-compose stop
```

*# На этом этапе запускаем контейнеры с просмотром вывода консоли, для этого необходимо убрать аргумент "-d".*

```
docker-compose --profile nginx --profile prod up
```

Если всё пройдет успешно, то после запуска UnicoreCMS в консоли отобразятся данные подобного характера:

```
[UsersModule] Administrator account has been created +167ms  
[UsersModule] Username: admin +0ms  
[UsersModule] Password: EcUn3lYEA34CzNZV +0ms  
[UsersModule] Save or remember this information +0ms
```

Это означает, что был создан новый аккаунт - суперпользователя.

Останавливаем контейнеры и перезапускаем с выключенным просмотром вывода консоли

```
docker-compose stop
```

```
docker-compose --profile nginx --profile prod up -d
```

## Заполнение БД (не обязательно)

Данный пункт заполнит сайт демо-контентом, это может помочь при ознакомлении с сайтом и работе с ПУ.

```
docker exec -t -i unicorecms npm run seed
```

# Manual-установка

## Системные требования

- NodeJS  $\geq 14 \leq 16$ , пакетный менеджер Yarn.
- Nginx или Apache веб-сервер в качестве Reverse-proxy.

- База данных: MySQL, MariaDB, MongoDB, Postgres, CockroachDB, SQLite, Microsoft SQL Server, Oracle или SAP Hana.

## Сборка UnicoreCMS

Производим установку npm-пакетов

```
npx lerna bootstrap
```

Выполняем сборку UnicoreCMS

```
npm run build
```

## Базовая настройка

### ⚠ НАСТРОЙКА NGINX-СЕРВЕРА

Пример настройки виртуальных хостов для Nginx

```
# conf.d/client.conf
server {
    listen 80 default_server;

    server_name example.com www.example.com;

    location / {
        proxy_pass http://127.0.0.1:3000;
    }
}

# conf.d/admin.conf
server {
    listen 80;

    server_name admin.example.com;

    location / {
        proxy_pass http://127.0.0.1:4000;
    }
}
```

```
}

# conf.d/server.conf
server {
    listen 80;

    server_name api.example.com;

    location / {
        proxy_pass http://127.0.0.1:5000;
    }
}
```

- Выполните настройку [Подключения к БД](#).
- Выполните настройку [Отправки писем](#).
- Выполните настройку [Recaptcha](#).
- Укажите `BASEURL` и `API_BASEURL` из раздела [Переменные Enviroment](#)

## Синхронизация БД

```
npm run sync
```

Производим запуск

```
npm run start
```

Если всё пройдет успешно, то после запуска UnicoreCMS в консоли отобразятся данные подобного характера:

```
[UsersModule] Administrator account has been created +167ms
[UsersModule] Username: admin +0ms
[UsersModule] Password: EcUn3lYEA34CzNZV +0ms
[UsersModule] Save or remember this information +0ms
```

Это означает, что был создан новый аккаунт - суперпользователя.

## Заполнение БД (не обязательно)

Данный пункт заполнит сайт демо-контентом, это может помочь при ознакомлении с сайтом и работе с ПУ.

```
npm run seed
```

[Настройка](#)[Подключение к БД](#)

Содержание этой страницы



# Подключение к БД

Совокупность данных, хранимых UnicoreCMS, манипулирование которыми выполняют в соответствии с правилами средств моделирования базой данной.

## При использования Docker

Редактирование следующих параметров производится в файле **.env**.

```
MYSQL_DATABASE= # Генерируемое имя БД Docker (например: unicore) [string]
MYSQL_USER= # Генерируемое имя пользователя БД Docker (например: unicore) [string]
MYSQL_PASSWORD= # Генерируемый пароль пользователя БД Docker [string]
MYSQL_ROOT_PASSWORD= # Генерируемый пароль root пользователя БД Docker [string]

DATABASE_HOST=mariadb
DATABASE_USER= # Должен совпадать с MYSQL_USER
DATABASE_PASSWORD= # Должен совпадать с MYSQL_PASSWORD
DATABASE_NAME= # Должен совпадать с MYSQL_DATABASE
```

### Enviroment-переменные UnicoreCMS

#### CAUTION

Удалите переменные DATABASE\_TYPE, DATABASE\_PORT из .env при их наличии

## При использования Manual

### Все настройки подключения

Редактирование следующих параметров производится в файле **.env**.

```
DATABASE_TYPE= # Драйвер подключения к БД [string, по-умолчанию: mysql]
DATABASE_HOST= # Адресс БД [string, по-умолчанию: 127.0.0.1]
DATABASE_PORT= # Порт БД [0 - 65536, по-умолчанию: 3306]
DATABASE_USER= # Имя пользователя БД [string, обязательно в зависимости от драйвера]
DATABASE_PASSWORD= # Пароль пользователя БД [string, обязательно в зависимости от драйвера]
DATABASE_NAME= # Имя БД или путь к файлу БД [string, обязательно]
```

## Enviroment-переменные UnicoreCMS

# Установка драйвера базы данных

Если используемая база-данных не является MySQL/MariaDB, то вам необходимо вручную установить один из следующих драйверов прописав команду в каталоге приложения.

- PostgreSQL или CockroachDB

```
npm install pg -w uncore-server --save
```

- SQLite

```
npm install sqlite3 -w uncore-server --save
```

- Microsoft SQL Server

```
npm install mssql -w uncore-server --save
```

- Oracle

```
npm install oracledb -w uncore-server --save
```

- SAP Hana

```
npm install @sap/hana-client hdb-pool -w uncore-server --save
```

- MongoDB (экспериментальный)

```
npm install mongodb@^3.6.0 -w unicon-server --save
```



[Настройка](#)[Отправка писем](#)

Содержание этой страницы



# Отправка писем

Для отправки писем UnicoreCMS использует пакет NodeMailer.

## Виды сообщений:

- Подтверждение почты
- Вход с нового устройства
- Восстановление пароля

## Настройка

### Enviroment-переменные

Редактирование следующих параметров производится в файле **.env**.

```
MAIL_FROM= # Имя ящика с которого будут отправляться письма (например: no-reply@example.com) [string]
SMTP_SERVICE= # Имя сервиса nodemailer [string]
SMTP_HOST= # Хост SMTP-сервера [string]
SMTP_PORT= # Порт SMTP-сервера [0 - 65536]
SMTP_IGNORE_TLS= # Отключает поддержку STARTTLS, если значение true [boolean]
SMTP_SECURE= # Логическое значение, если следует использовать SMTP Secure [boolean]
SMTP_USER= # Пользователь для аутентификации на сервере SMTP [string]
SMTP_PASSWORD= # Пароль для пользователя SMTP
```

Enviroment-переменные UnicoreCMS

### Использование пресета

Вы можете использовать сервис NodeMailer, для автоматической настройки параметров подключения к серверу

```
MAIL_FROM= # Имя ящика с которого будут отправляться письма (например: no-reply@example.com) [string]
SMTP_SERVICE= # Имя сервиса nodemailer [string]
SMTP_USER= # Пользователь для аутентификации на сервере SMTP [string]
SMTP_PASSWORD= # Пароль для пользователя SMTP
```

## Поддерживаемые сервисы/пресеты

126, 163, 1und1, AOL, DebugMail, DynectEmail, FastMail, GandiMail, Gmail, Godaddy, GodaddyAsia, GodaddyEurope, hot.ee, Hotmail, iCloud, mail.ee, Mail.ru, Maildev, Mailgun, Mailjet, Mailosaur, Mandrill, Naver, OpenMailBox, Outlook365, Postmark, QQ, QQex, SendCloud, SendGrid, SendinBlue, SendPulse, SES, SES-US-EAST-1, SES-US-WEST-2, SES-EU-WEST-1, Sparkpost, Yahoo, Yandex, Zoho, qiye.aliyun

## Ручная настройка подключения

```
MAIL_FROM= # Имя ящика с которого будут отправляться письма (например: no-reply@example.com) [string]
SMTP_HOST= # Хост SMTP-сервера [string]
SMTP_PORT= # Порт SMTP-сервера [0 - 65536]
SMTP_IGNORE_TLS= # Отключает поддержку STARTTLS, если значение true [boolean]
SMTP_SECURE= # Логическое значение, если следует использовать SMTP Secure [boolean]
SMTP_USER= # Пользователь для аутентификации на сервере SMTP [string]
SMTP_PASSWORD= # Пароль для пользователя SMTP
```

[Настройка](#)[Recaptcha](#)

Содержание этой страницы



# Recaptcha

Для защиты некоторых роутов UnicoreCMS помимо rate-limiter использует фоновую проверку Recapth V3, а именно:

- Вход
- Регистрация
- Восстановление пароля
- Подтверждение Email
- Активация Gift-кода

## Настройка

Переходим в [админ-консоль](#) и создаём новый сайт, указав необходимую версию Recaptcha.

Ярлык



unicorecms

10/50

Тип reCAPTCHA



reCAPTCHA v3

Выполнять проверку запросов с учетом оценок



reCAPTCHA v2

Проверка запросов с помощью заданий

Домены



unicore.uno



Добавьте домен, например example.com

## Производим настройку .env.

```
RECAPTCHA_PUBLIC= # Публичный ключ Recaptcha v3 [string, обязательно]  
RECAPTCHA_SECRET= # Секретный ключ Recaptcha v3 [string, обязательно]
```

Enviroment-переменные UnicoreCMS



Настройка



VKontakte LongPoll

# VKontakte LongPoll



Настройка



Переменные Enviroment

# Переменные Enviroment

На этой странице представлена перечень всех переменных, которые есть в UnicoreCMS и где-либо используются.

## ! INFO

Файл .env — это перечень переменных окружения UnicoreCMS и UnicoreCMS Docker с присвоенными им определенными значениями.

```
SITENAME= # Имя проекта [string, по-умолчанию: UnicoreCMS]
BASEURL= # Базовый URL-адресс клиента (например: https://example.com) [string, обязательно]
API_BASEURL= # Базовый URL-адресс сервера (например: https://api.example.com) [string, обязательно]

# Настройки подключения к БД
DATABASE_TYPE= # Драйвер подключения к БД [string, по-умолчанию: mysql]
DATABASE_HOST= # Адресс БД [string, по-умолчанию: 127.0.0.1]
DATABASE_PORT= # Порт БД [0 - 65536, по-умолчанию: 3306]
DATABASE_USER= # Имя пользователя БД [string, обязательно в зависимости от драйвера]
DATABASE_PASSWORD= # Пароль пользователя БД [string, обязательно в зависимости от драйвера]
DATABASE_NAME= # Имя БД или путь к файлу БД [string, обязательно]

# Подпись JWT-токенов
JWT_KEY= # Секретный ключ для подписи токенов [string, обязательно]
JWT_EXPIRES= # Время истечения AccessToken (например: 5m, 1h, 30d и т.д.) [string, по-умолчанию: 5m]
JWT_EXPIRES= # Время истечения RefreshToken (например: 5m, 1h, 30d и т.д.) [string, по-умолчанию: 30d]

# Настройки Recaptcha v3
RECAPTCHA_PUBLIC= # Публичный ключ Recaptcha v3 [string, обязательно]
RECAPTCHA_SECRET= # Секретный ключ Recaptcha v3 [string, обязательно]

# VK Longpoll
VK_LONGPOLL= # Использовать VK Longpoll для парсинга новостей из группы [boolean]
```

`VK_APIKEY=` # API-ключ VK Longpoll [string]

`# SMTP`

`MAIL_FROM=` # Имя ящика с которого будут отправляться письма (например: no-reply@example.com) [string]

`SMTP_SERVICE=` # Имя сервиса nodemailer [string]

`SMTP_HOST=` # Хост SMTP-сервера [string]

`SMTP_PORT=` # Порт SMTP-сервера [0 - 65536]

`SMTP_IGNORE_TLS=` # Отключает поддержку STARTTLS, если значение true [boolean]

`SMTP_SECURE=` # Логическое значение, если следует использовать SMTP Secure [boolean]

`SMTP_USER=` # Пользователь для аутентификации на сервере SMTP [string]

`SMTP_PASSWORD=` # Пароль для пользователя SMTP

`# Порты`

`CLIENT_PORT=` # Порт клиента [0 - 65536, по-умолчанию: 3000]

`ADMIN_PORT=` # Порт админ-панели [0 - 65536, по-умолчанию: 4000]

`SERVER_PORT=` # Порт сервера [0 - 65536, по-умолчанию: 5000]

`# Следующие переменные обязательны только при использовании Docker`

`MYSQL_DATABASE=` # Генерируемое имя БД Docker (например: uncore) [string]

`MYSQL_USER=` # Генерируемое имя пользователя БД Docker (например: uncore) [string]

`MYSQL_PASSWORD=` # Генерируемый пароль пользователя БД Docker [string]

`MYSQL_ROOT_PASSWORD=` # Генерируемый пароль root пользователя БД Docker [string]

`CLIENT_HOST=` # Хост клиента (например: example.com) [string]

`ADMIN_HOST=` # Хост админ-панели (например: admin.example.com) [string]

`SERVER_HOST=` # Хост сервера (например: api.example.com) [string]

[Настройка](#)[UnicoreConnect](#)

Содержание этой страницы



# UnicoreConnect

Плагин **Spigot/Sponge** для двухнаправленной интеграции экономики, прав, групп, склада-покупок, банов и сбора статистики между сайтом и серверами.

## Поддерживаемые ядра

- Spigot/Bukkit 1.7.10 - 1.18.1, также их форки по типу PaperSpigot.
- Гибритные ядра: Thermos, Mohist, Magma и т. д.
- Sponge Vanilla и Sponge Forge.

### ❗ КСТАТИ

Модуль склада имеет полную поддержку модовых предметов, включая NBT-теги, а также выдаёт вещи правильным размером стака!

## Опциональные зависимости

- **LuckPerms** - включит модуль выдачи донат-прав/групп (Spigot/Sponge)
- **BanManager** - включит модуль двухнаправленной интеграции банов с сайтом (Spigot/Sponge)
- **Essentials/EssentialsX/CMI/Nucleus** - проверка на AFK модулем PlayTime (Spigot/Sponge)
- **Vault** - включит модуль внутриигровой экономики (Spigot)

# Установка и настройка

1. Создайте API-ключ с правом `unicore.kernel.connect`.
2. Поместите Jar-файл в папку плагинов.
3. Произведите настройку файла конфигурации UnicoreConnect.

`server`: id сервера  
`api`:



**url**: Адрес UnicoreCMS-сервера  
**key**: API-ключ

# Сборка

UnicoreConnect использует Gradle для обработки зависимостей и сборки.

## Зависимости

- Java 8 JDK или более поздней версии
- Git

## Компиляция

```
git clone https://github.com/UnicoreProject/UnicoreConnect.git
cd UnicoreConnect/
./gradlew build
```

Собранный Jar-файл будет лежать в папке build/libs

# Команды и права

## Экономика

Команды	Пермишен	Описание
/money, /bal, /balance	unicoreconnect.command.money	Баланс внутриигровой валюты
/moneytop, /baltop, /balancetop	unicoreconnect.command.moneytop	Топ 10 богачей сервера
/pay [player] [amount]	unicoreconnect.command.pay	Перевести монеты игроку

# PlayTime

Команды	Пермишен	Описание
/playtime, /pt	unicoreconnect.command.playtime	Время проведённое на сервере
/playtimetop, /pttop	unicoreconnect.command.playtimetop	Топ 10 по времени онлайн сервера

## Склад и магазин

Команды	Пермишен	Описание
/cart list	unicoreconnect.command.showcase.list	Список товаров на складе
/cart give [id]	unicoreconnect.command.showcase.give	Выдать товар со склада
/cart all	unicoreconnect.command.showcase.all	Выдать все товары со склада
/cart add [price] [name]	unicoreconnect.command.showcase.create	Добавить предмет, находящийся в руке в магазин

## Администрирование

Команды	Пермишен	Описание
/uc reload	unicoreconnect.admin	Перезагрузить плагин
/uc sync	unicoreconnect.admin	Переподключится к Unicoreserver

[Настройка](#)[UnicoreProvider](#)

Содержание этой страницы



# UnicoreProvider

Модуль для интеграции сервисов авторизации [UnicoreCMS](#) и [GravitLauncher](#)

## Установка и настройка

1. Создайте API-ключ с правом `unicore.kernel.provider`.
2. Поместите Jar-файл в **modules/**.
3. Произведите настройку **LaunchServer.json**.

```
// ...  
"core": {  
  "type": "unicore",  
  "apiUrl": "Адрес UnicoreCMS-сервера",  
  "apiKey": "API-ключ"  
},  
// ...
```

## Сборка

UnicoreProvider использует Gradle для обработки зависимостей и сборки.

### Зависимости

- Java 17 JDK или более поздней версии
- Git

### Компиляция

```
git clone https://github.com/UnicoreProject/UnicoreProvider.git  
cd UnicoreProvider/
```

```
./gradlew build
```

Собранный Jar-файл будет лежать в папке build/libs



Работа с ПУ



Статистика

# Статистика



Работа с ПУ



Пользователи

# Пользователи

[Работа с ПУ](#)[Роли и разрешения](#)

Содержание этой страницы



# Роли и разрешения

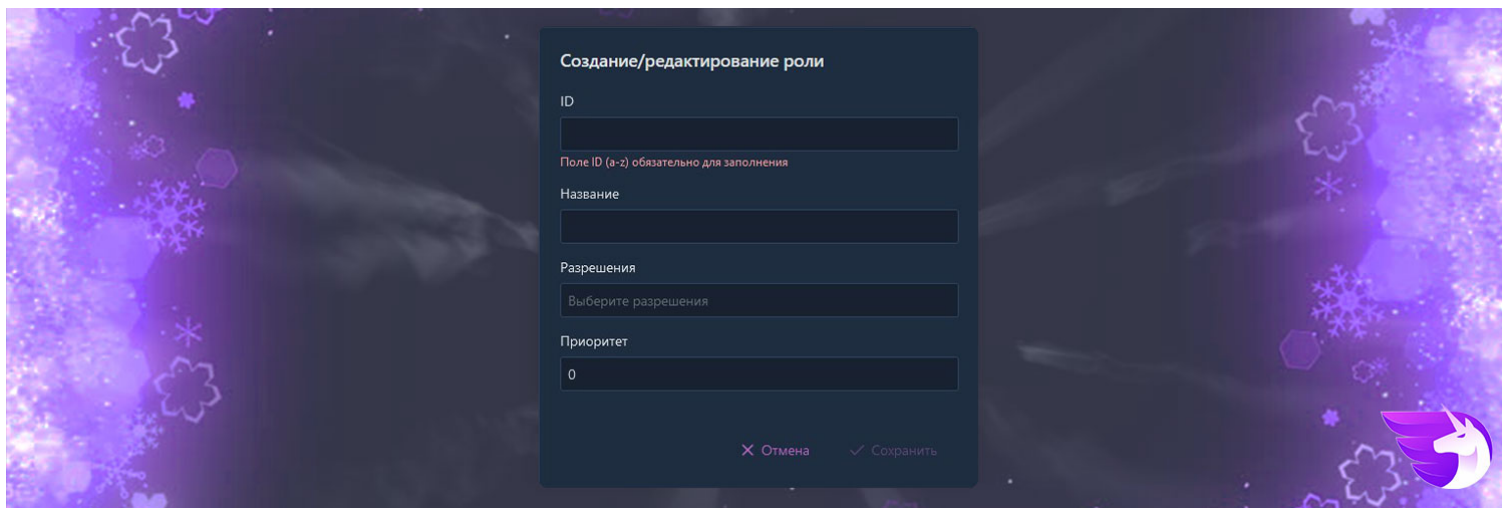


## Обязательные роли

**Обязательные роли** - системные роли, доступные только для редактирования.

- **default** - автоматически присваивается всем пользователям.
- **banned** - автоматически присваивается заблокированным пользователям.

## Добавление роли



## Параметры:

- **ID** - идентификатор роли (неизменяем в будущем).
- **Название** - отображаемое имя роли.
- **Разрешения** - перечень прав роли.

## Формирование разрешений

При указании разрешений вы можете использовать glob-синтаксис, например:

- `user.cabinet.*` - будут выданы разрешения `user.cabinet.skin`, `user.cabinet.skin.hd`, `user.cabinet.cloak`, `user.cabinet.cloak.hd`.
- `user.cabinet.(skin|cloak)` - будут выданы разрешения `user.cabinet.skin`, `user.cabinet.cloak`.
- Исключать права более приоритетной ролью, добавив `!` перед разрешением.

## Список разрешений

### Основной сайт

Пермишен	Описание
<code>user.cabinet.skin</code>	Возможность загружать/удалять скин



Пермишен	Описание
<code>user.cabinet.skin.hd</code>	Возможность загружать HD скин
<code>user.cabinet.cloak</code>	Возможность загружать/удалять плащ
<code>user.cabinet.cloak.hd</code>	Возможность загружать HD плащ

## Панель управление

Пермишен	Описание
<code>admin.dashboard</code>	Просмотр статистики <b>(обязательный пермишен для доступа в ПУ)</b>
<code>admin.users.read</code>	Просмотр пользователей
<code>admin.users.create</code>	Создание пользователей
<code>admin.users.update</code>	Редактирование пользователей
<code>admin.users.delete</code>	Удаление пользователей
<code>admin.users.delete.many</code>	Массовое удаление пользователей
<code>admin.servers.create</code>	Создание серверов
<code>admin.servers.update</code>	Редактирование серверов
<code>admin.servers.delete</code>	Удаление серверов
<code>admin.pages.read</code>	Просмотр статических страниц
<code>admin.pages.create</code>	Создание статических страниц

Пермишен	Описание
<code>admin.pages.update</code>	Редактирование статических страниц
<code>admin.pages.delete</code>	Удаление статических страниц
<code>admin.email.read</code>	Просмотр Email-сообщений
<code>admin.email.test</code>	Тестирование SMTP-сервера
<code>admin.email.update</code>	Редактирование Email-сообщений
<code>admin.webhooks.read</code>	Просмотр вебхуков
<code>admin.webhooks.create</code>	Создание вебхуков
<code>admin.webhooks.update</code>	Редактирование вебхуков
<code>admin.webhooks.delete</code>	Удаление вебхуков
<code>admin.webhooks.delete.many</code>	Массовое удаление вебхуков
<code>editor.mods.create</code>	Создание модов
<code>editor.mods.update</code>	Редактирование модов
<code>editor.mods.delete</code>	Удаление модов
<code>editor.mods.delete.many</code>	Массовое удаление модов
<code>editor.news.read</code>	Просмотр новостей
<code>editor.news.create</code>	Создание новостей
<code>editor.news.update</code>	Редактирование новостей

Пермишен	Описание
<code>editor.news.delete</code>	Удаление новостей
<code>editor.news.delete.many</code>	Массовое удаление новостей
<code>editor.donate.read</code>	Просмотр донат-групп, донат-прав, донат-китов
<code>editor.donate.groups.create</code>	Создание донат-групп
<code>editor.donate.groups.update</code>	Редактирование донат-групп
<code>editor.donate.groups.delete</code>	Удаление донат-групп
<code>editor.donate.groups.delete.many</code>	Массовое удаление донат-групп
<code>editor.donate.permissions.create</code>	Создание донат-прав
<code>editor.donate.permissions.update</code>	Редактирование донат-прав
<code>editor.donate.permissions.delete</code>	Удаление донат-прав
<code>editor.donate.permissions.delete.many</code>	Массовое удаление донат-прав
<code>editor.donate.kits.create</code>	Создание донат-китов
<code>editor.donate.kits.update</code>	Редактирование донат-китов
<code>editor.donate.kits.delete</code>	Удаление донат-китов
<code>editor.donate.kits.delete.many</code>	Массовое удаление донат-китов
<code>editor.donate.periods.create</code>	Создание периодов покупки доната
<code>editor.donate.periods.update</code>	Редактирование периодов покупки доната

Пермишен	Описание
<code>editor.donate.periods.delete</code>	Удаление периодов покупки доната
<code>editor.store.read</code>	Просмотр категорий, товаров, китов магазина
<code>editor.store.category.create</code>	Создание категорий магазина
<code>editor.store.category.update</code>	Редактирование категорий магазина
<code>editor.store.category.delete</code>	Удаление категорий магазина
<code>editor.store.category.delete.many</code>	Массовое удаление категорий магазина
<code>editor.store.products.create</code>	Создание товаров магазина
<code>editor.store.products.update</code>	Редактирование товаров магазина
<code>editor.store.products.update.many</code>	Массовое редактирование товаров магазина
<code>editor.store.products.delete</code>	Удаление товаров магазина
<code>editor.store.products.delete.many</code>	Массовое удаление товаров магазина
<code>editor.store.products.import</code>	Импортирование товаров в магазин
<code>editor.store.products.export</code>	Экспортирование товаров из магазина
<code>editor.store.kits.create</code>	Создание китов магазина
<code>editor.store.kits.update</code>	Редактирование китов магазина
<code>editor.store.kits.delete</code>	Удаление китов магазина
<code>editor.store.kits.delete.many</code>	Массовое удаление китов магазина



Работа с ПУ



Новости

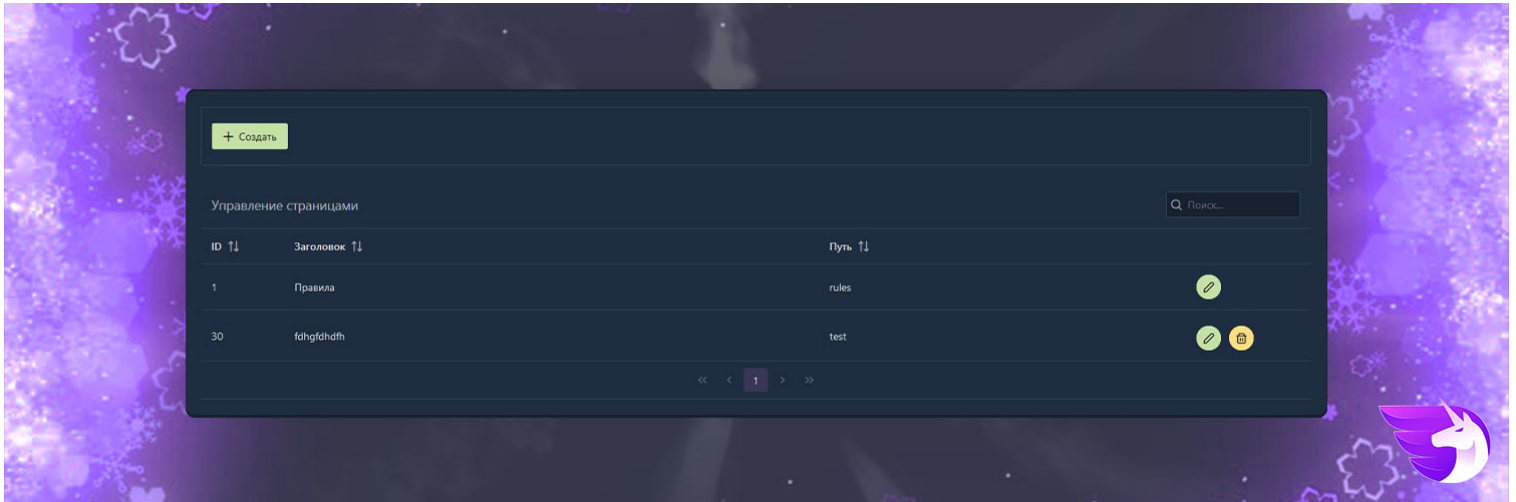
# Новости

[Работа с ПУ](#)[Статические страницы](#)

Содержание этой страницы



# Статические страницы



## Добавление страницы

### Параметры:

- **Путь** - уникальный путь к странице от корня сайта (например test - yourdomain.ru/test).
- **Заголовок** - заголовок страницы.
- **Содержимое** - весь контент выводимый на странице.
- **Meta-description** - описание используемое для оптимизации в поисковых системах (не обязательно).

### Визуальный HTML редактор

Subheading Sans Serif B I U A A [List Icons] [Link Icon] [Code Icon] Ix

# Header

## Suheading

Content Content Content Content  
Content Content Content Content



- fdgdfgdfg
- fdgdfgdfg
- fdgdfgdfgdfg

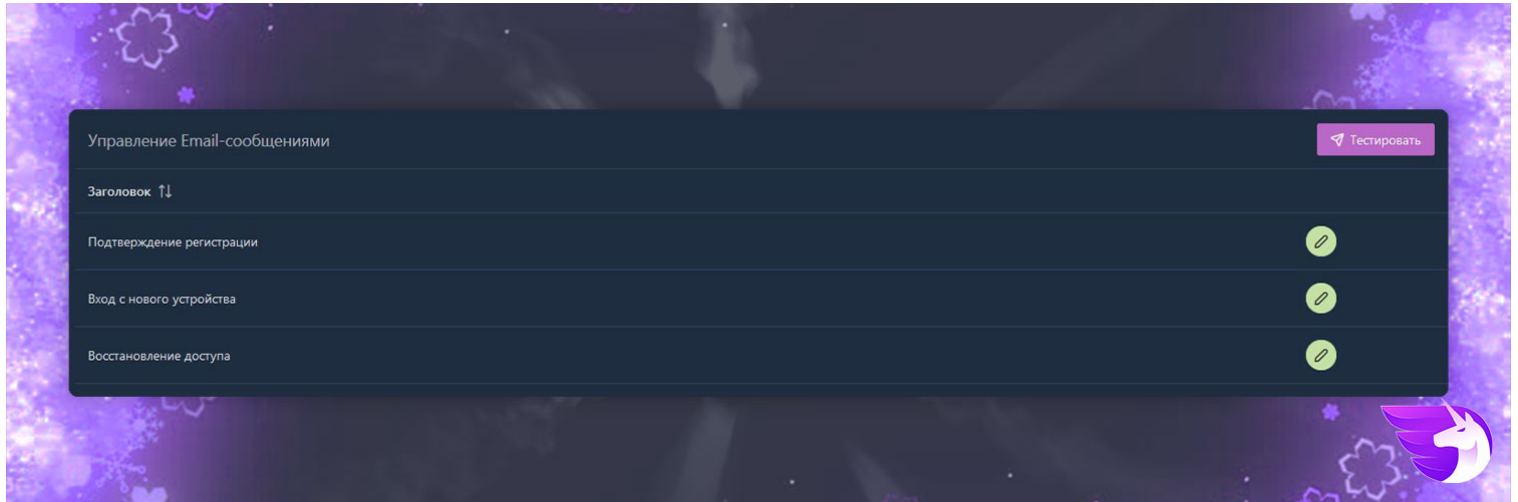


[Работа с ПУ](#)[Email-сообщения](#)

Содержание этой страницы



# Email-сообщения



## Редактирование сообщений

### Параметры:

- **Заголовок** - Заголовок страницы.
- **Содержимое** - Контент сообщения.

### Переменные:

#### ! INFO

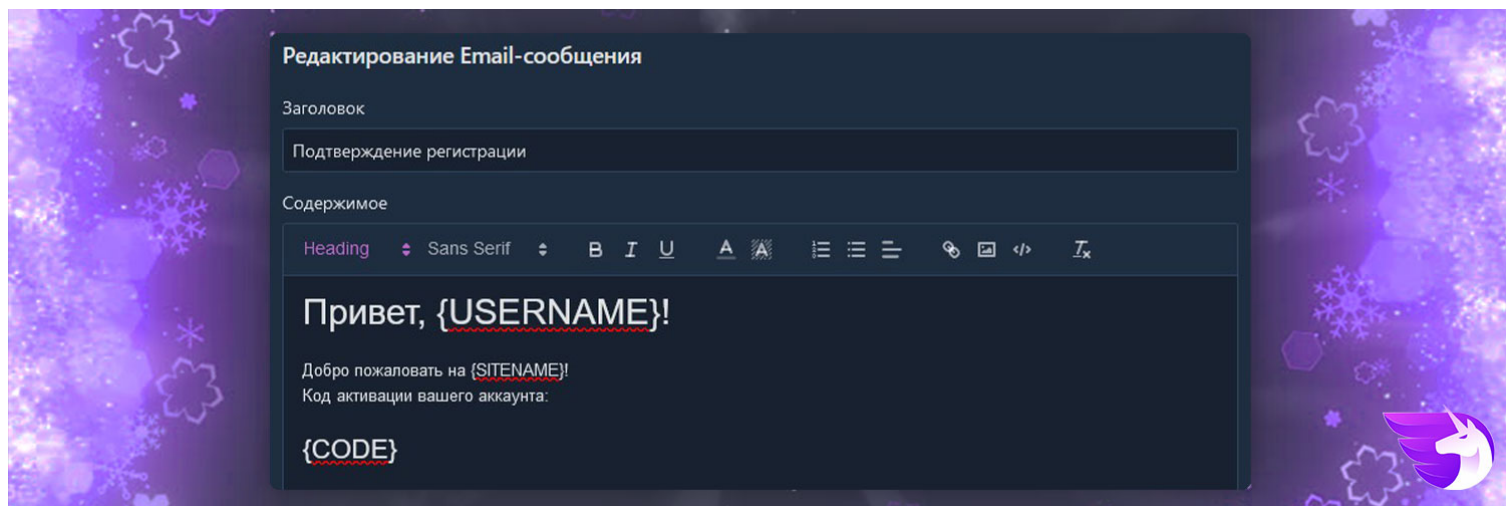
Переменные указываются в фигурных скобках, пример: `{USERNAME}`

- **Подтверждение регистрации:**
  - `USERNAME` - имя пользователя.
  - `SITENAME` - имя сайта (env SITENAME)
  - `CODE` - Код активации
- **Вход с нового устройства:**
  - `USERNAME` - имя пользователя.



- `SITENAME` - имя сайта (env SITENAME)
- `IP` - ip-адресс с которого был совершён вход
- **Восстановление доступа:**
  - `USERNAME` - имя пользователя.
  - `SITENAME` - имя сайта (env SITENAME)
  - `LINK` - ссылка для смены пароля
  - `IP` - ip-адресс с которого был совершён запрос

## Визуальный HTML редактор



Точно такой же, как и в статических страницах

[Работа с ПУ](#)[Серверы и моды](#)

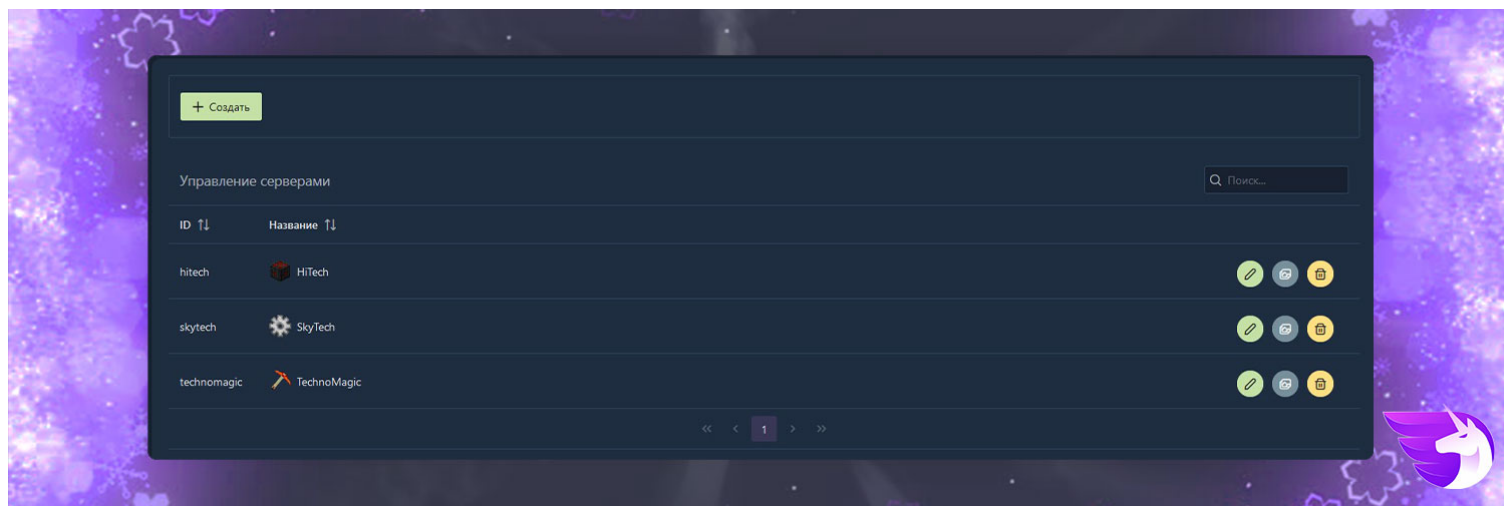
Содержание этой страницы



# Серверы и моды

## Добавление сервера

**Серверы** - это ваши игровые серверы, UnicoreCMS умеет самостоятельно формировать страницы с их описанием, включая таблицу характеристик и блоки модов, а также добавляет их в индексацию поисковыми системами.



### Параметры:

- **ID** - уникальный идентификатор сервера (неизменяем в будущем).
- **Название** - отображаемое имя сервера.
- **Версия** - игровая версия Minecraft (фактически любая строка).
- **Слоган** - что-то вроде очень краткого описания...
- **Моды** - перечень **модов** на сервере (не обязательно).
- **Описание** - полное описание сервера, поддерживает: **Bold**, **Strike**, *Italic* (не обязательно).
- **Meta-description** - описание используемое для оптимизации в поисковых системах (не обязательно).

Создание/редактирование сервера

ID

hitech

Название

HiTech

Версия

1.12.2

Слоган

Мир в котором можно всё

Моды

Applied Energistics 2

Ender IO

Mekanism

Industrial Foregoing

Выберите моды

Описание

B I U


praesentium ut fugiat

Описание (meta-description)

Редактирование иконки и картинки

Редактирование медиа


Иконка сервера



Загрузить

Удалить

Изображение сервера











Загрузить

Удалить

Построение таблицы характеристик

Характеристики (построение таблицы) +

Заголовок	Описание		
Режим сражений	PvP		
Размер основного мира	15000 блоков		
Размер доп. миров	5000 блоков		
Вайп доп. миров каждые	14 дней		

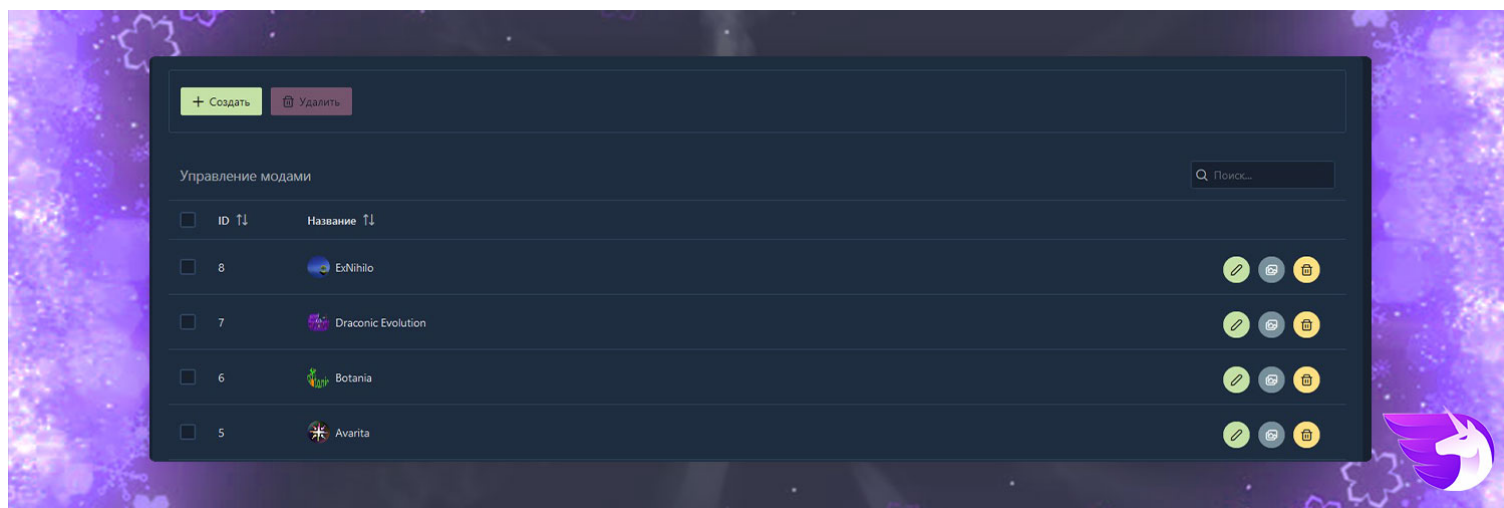
Мониторинг сервера

Вам необходимо включить Query-протокол на вашем сервере в файле **server.properties**

```
enable-query=true  
query.port=25565
```

## Добавление мода

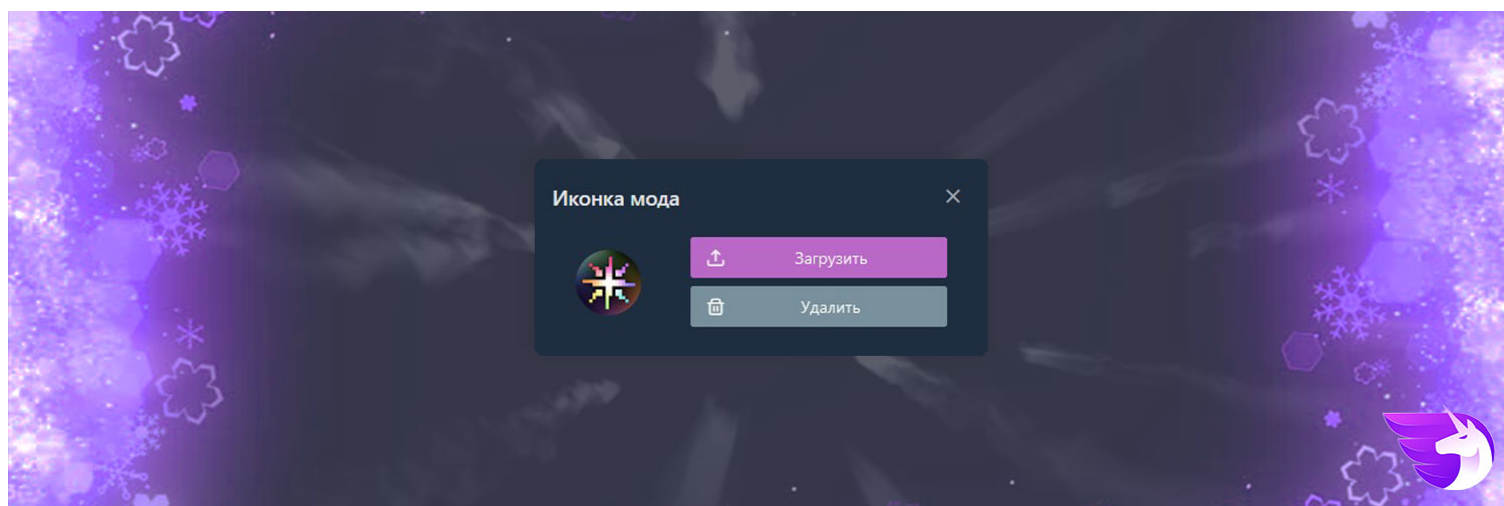
**Моды** - формируют блоки с перечнем модов на странице описания сервера.



### Параметры:

- **Название** - отображаемое имя мода.
- **Описание** - описание мода, поддерживает: **Bold**, *Strike*, *Italic* (не обязательно).

### Редактирование иконки



[Работа с ПУ](#)[Донат группы и права](#)

Содержание этой страницы



# Донат группы и права

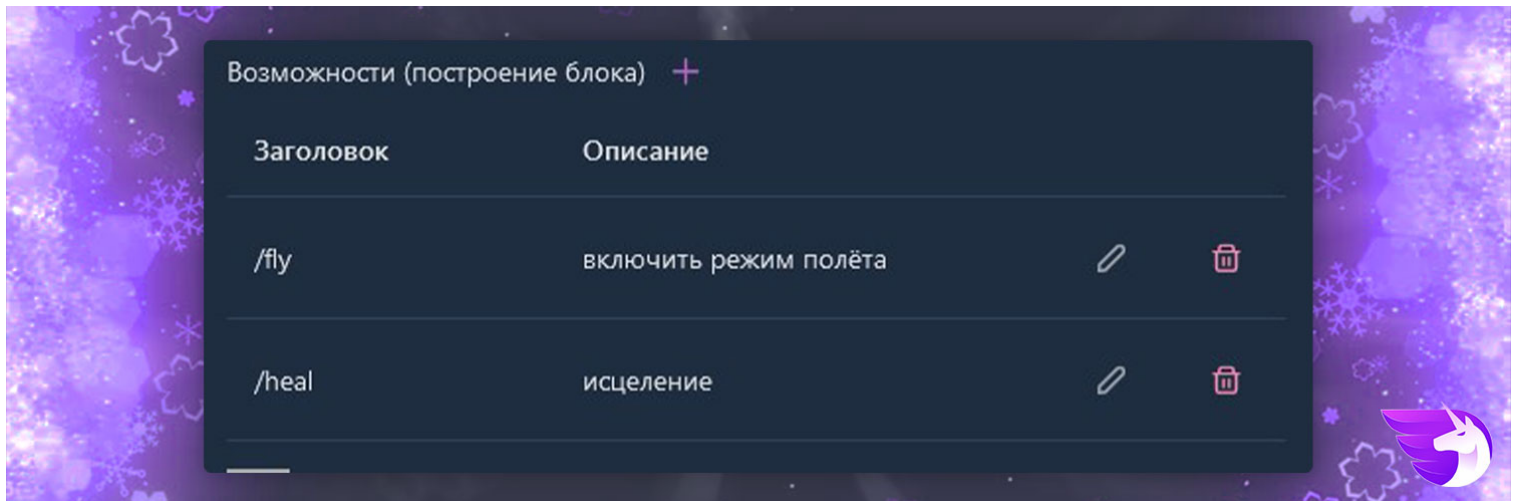
## Создание донат-группы

**Донат-группы** - продаваемые группы вашего сервера (maybe: Vip, Premium и т.д.).

### Параметры:

- **Название** - отображаемое имя донат-группы.
- **ID в игре** - ID в плагине LuckPerms.
- **Серверы** - перечень [серверов](#) на которых можно купить данный донат.
- **Доступные периоды** - перечень [периодов](#) доступных для покупки.
- **Киты** - перечень [донат-китов](#).
- **Инжект веб-прав** - наложение [веб-прав](#) (например: `user.cabinet.skin.hd` - возможность загружать HD скин).
- **Описание** - полное описание, поддерживает: **Bold**, **Strike**, *Italic* (не обязательно).
- **Возможности** - построение блока возможностей на странице описания донат-группы.
- **Цена, скидка, иконка**

### Построение блока возможностей





# Создание донат-права

**Донат-права** - продаваемые отдельные права вашего сервера, либо права сайта, либо права на доступ к донат-китам.

## Параметры:

- **Название** - отображаемое имя донат-права.
- **Доступные периоды** - перечень **периодов** доступных для покупки.
- **Описание** - полное описание, поддерживает: **Bold**, **Strike**, *Italic* (не обязательно).
- **Цена, скидка**

## Типы донат прав:

- **Игровые пермишены** - внутриигровые права (LuckPerms). Добавляет обязательные параметры:
  - **Серверы** - перечень **серверов** на которых можно купить данное донат-право.
  - **Права** - перечень прав LuckPerms
- **Веб-пермишены** - **веб-права** UnicoreCMS
  - **Права** - перечень прав **прав** UnicoreCMS
- **Киты** - тоже, что и **"Игровые пермишены"**, но добавляет параметр:
  - **Связанные киты** - Киты выводимые при покупке.

# Создание периода

## Параметры

**Периоды** - временные периоды, на которые можно купить определённую донат-группу/право.

- **Название** - отображаемое имя периода
- **Время в секундах** - время на которое будет выдаваться донат-группа/право с момента покупки, 0 - безлимитно
- **Множитель цены**

# Создание донат-кита

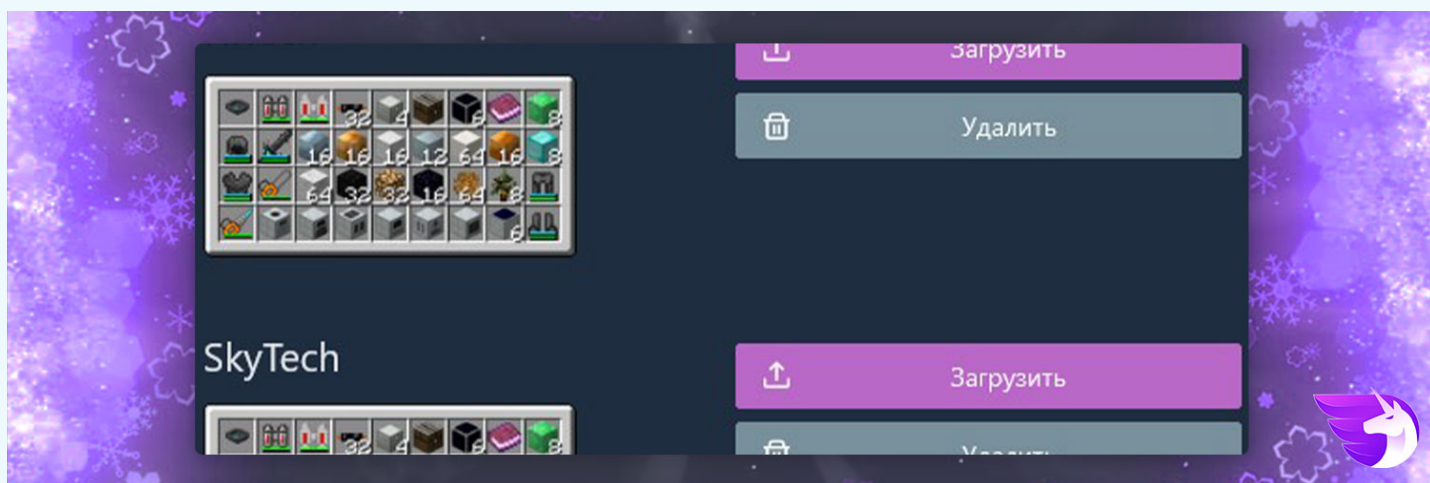
**Донат-киты** - служат для построения перечени китов на странице с описанием донат-групп или донат-прав, включающих этот кит.

## Параметры:

- **Название** - отображаемое имя донат-кита.
- **Описание** - полное описание донат-кита (например время cooldown), поддерживает: **Bold**, *Strike*, *Italic* (не обязательно).

### ❗ ИЗОБРАЖЕНИЕ КИТА

Вы должны загрузить изображение кита под каждый сервер (где используется донат-группа/ донат-право с ним) отдельно.



[Работа с ПУ](#)[Магазин](#)

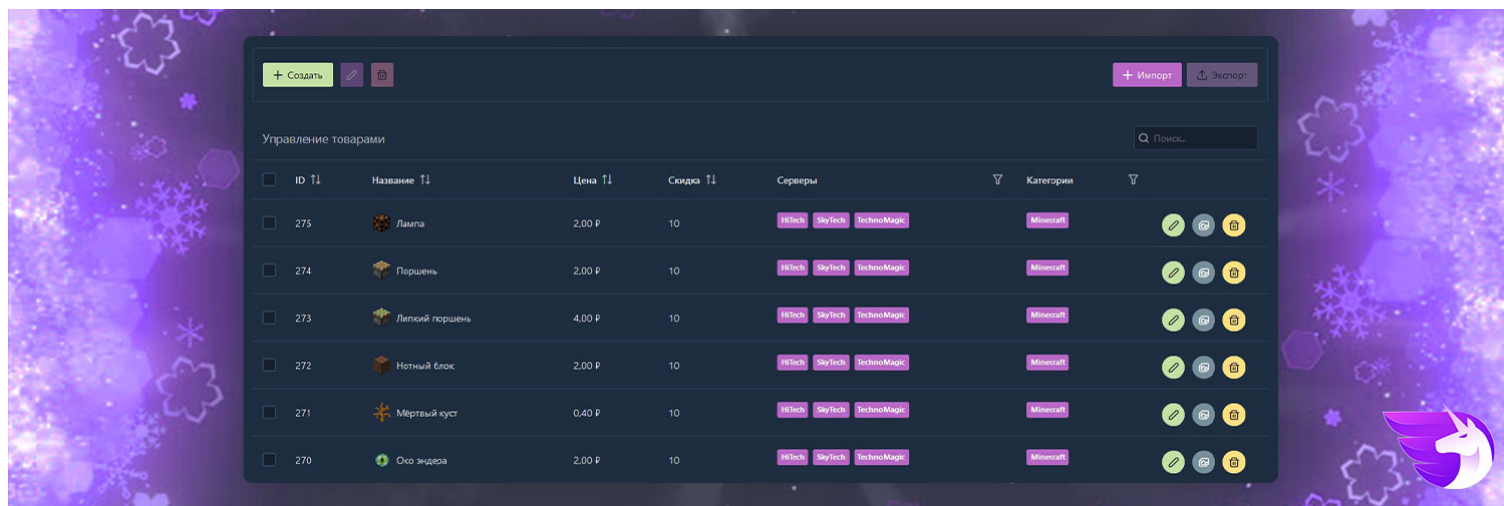
Содержание этой страницы



# Магазин

**Магазин** - интегрированный модуль, который позволяет продавать внутриигровые предметы и наборы.

## Создание товаров



### Параметры:

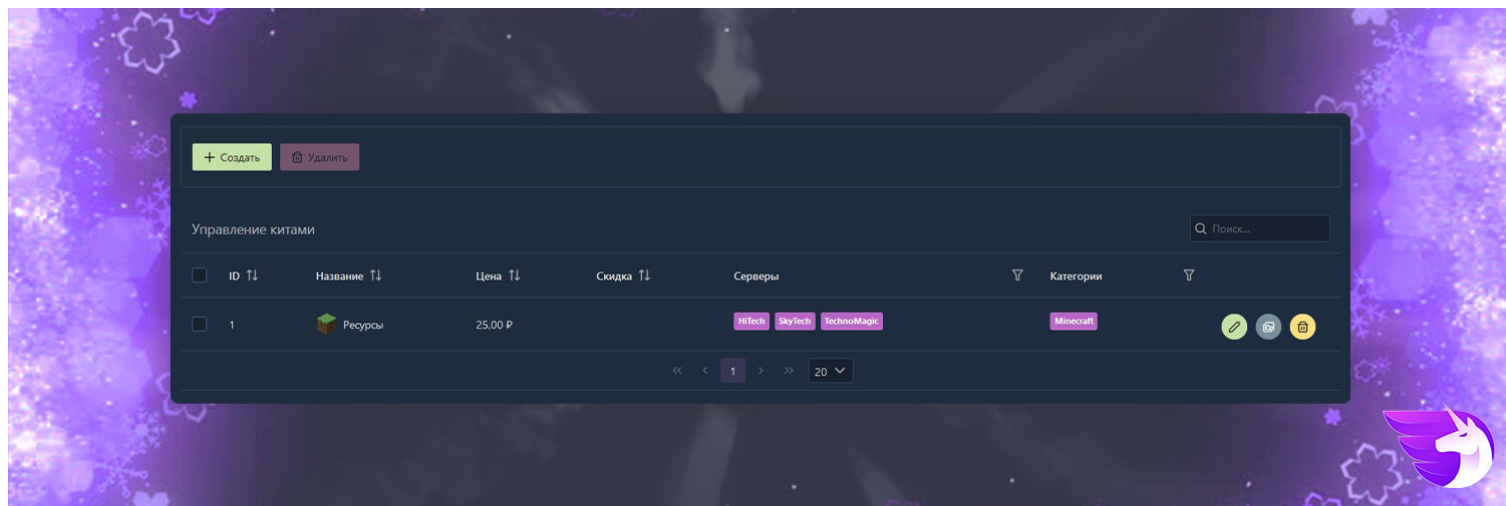
- **Название** - отображаемое имя.
- **ID предмета** - внутриигровое ID предмета (в подобном формате: `minecraft:redstone_lamp`), вы можете увидеть их включив расширенные подсказки в игре **F3 + H**. Также, если у предмета есть meta, её следует указывать в "*ID предмета*" через @ (например: `minecraft:dye@13`).
- **NBT-теги** - NBT-теги в подобном формате: `{EntityTag:{id:"minecraft:villager"}}`.
- **Описание** - поддерживает: **Bold**, *Strike*, *Italic* (не обязательно).
- **Серверы** - перечень серверов на которых можно купить данный товар.
- **Категории, цена, скидка, иконка.**

### Заполнение магазина через UnicoreConnect



Возьмите предмет в руку и пропишите `/cart add [price] [name]`. В результате будет создан новый товар, ему присвоится ему сервер, указанный в конфиге плагина, скопируются id в правильном формате, meta и nbt-теги.

## Создание китов



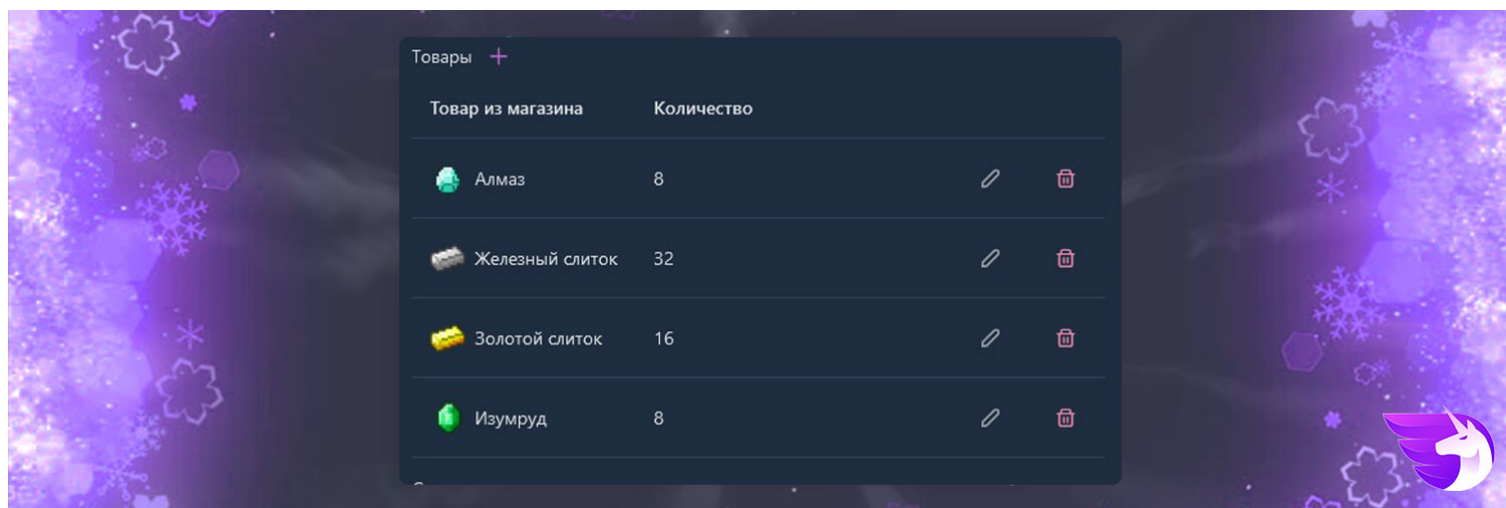
### Параметры:

- **Название** - отображаемое имя.
- **Описание** - поддерживает: **Bold**, ~~Strike~~, *Italic* (не обязательно).
- **Серверы** - перечень **серверов** на которых можно купить данный товар.
- **Категории**, **цена**, **скидка**, **иконка**.

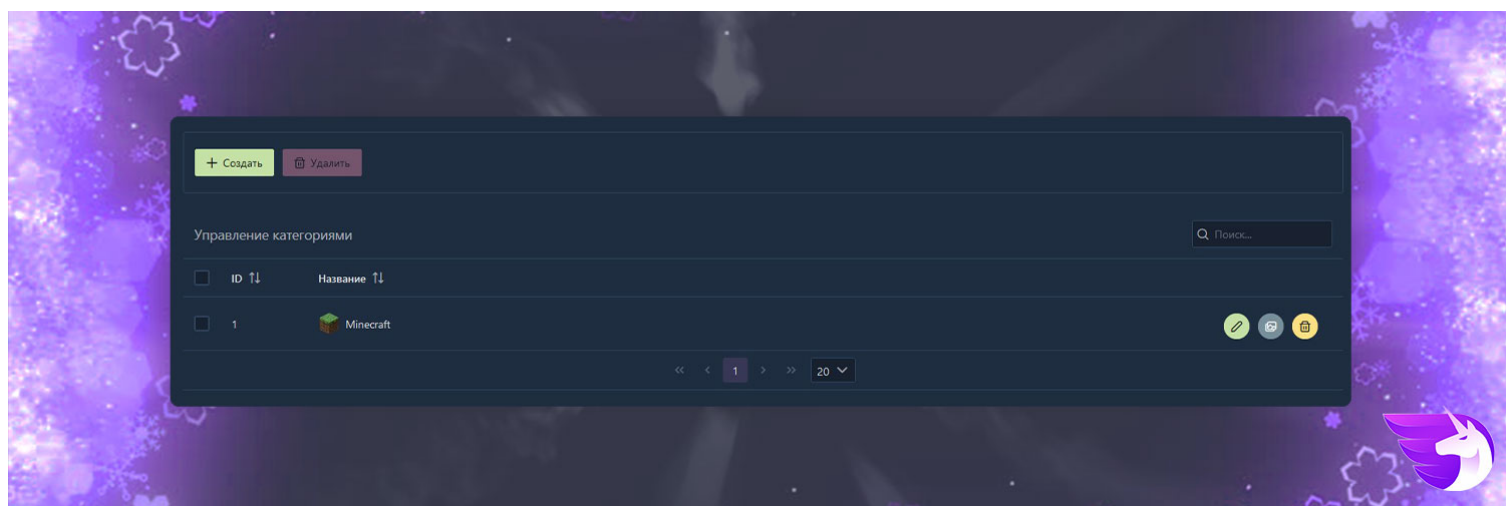
### Заполнение кита товарами:

#### ! INFO

Если вы хотите, чтоб предмет не выводился в магазине, а был доступен только в ките, то просто не заполняйте поле "**Серверы**" у товара.



## Создание категорий



### Параметры:

- **Название** - отображаемое имя.
- **Описание** - поддерживает: **Bold**, ~~Strike~~, *Italic* (не обязательно).
- **Иконка**



Работа с ПУ



Платежи

# Платежи

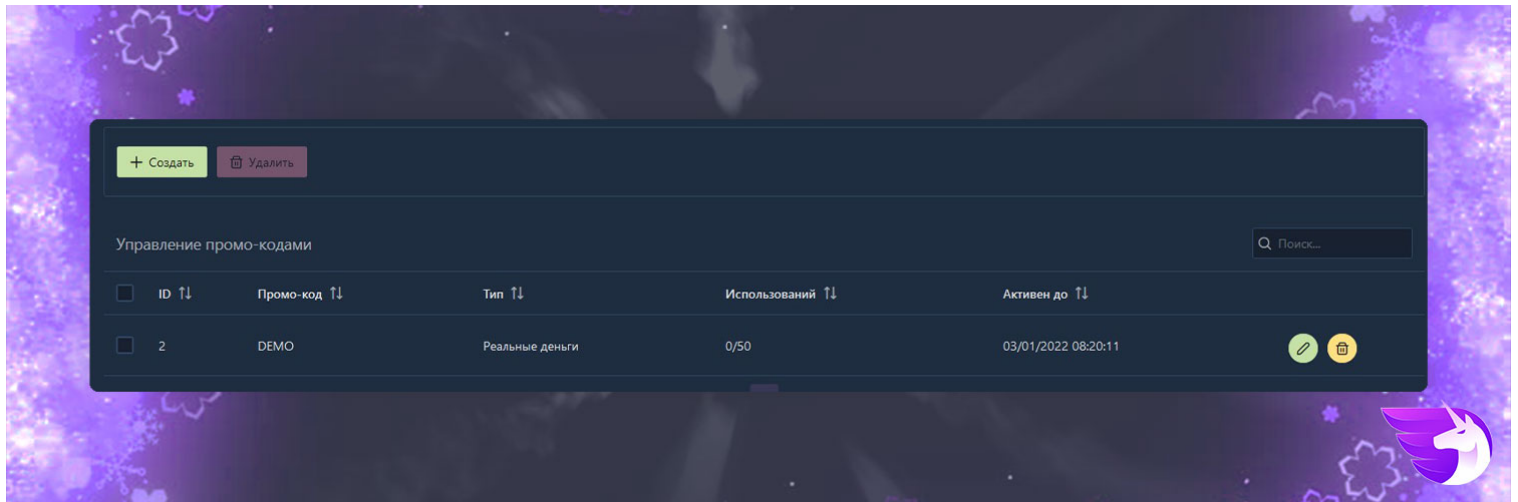
[Работа с ПУ](#)[Промо-коды](#)

Содержание этой страницы



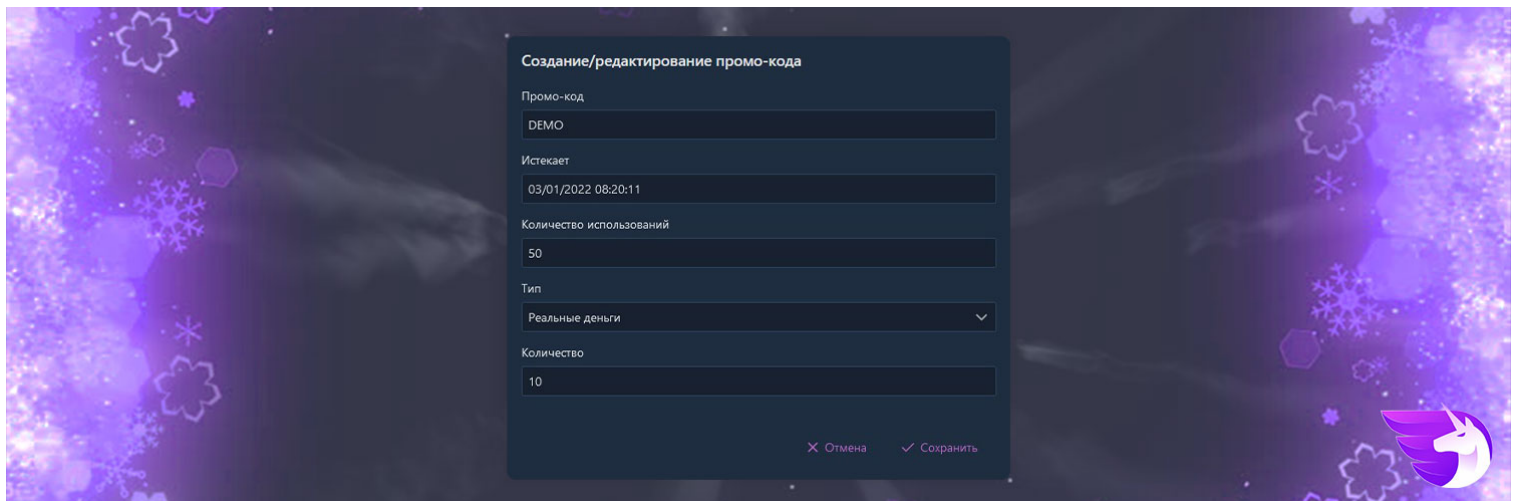
# Промо-коды

**Промо-коды** - купоны при активации которых игроку будет выдаваться подарок.



## Параметры:

- **Промо-код** - секретное слово купона.
- **Истекает** - дата и время истечение купона (не обязательно).
- **Количество использований** - максимальное количество использований купона.
- **Тип** - тип выдаваемого вознаграждения.



## Типы промо-кодов:

- **Реальные деньги** - реальный баланс сайта (рубли). Добавляет обязательные параметры:
  - **Количество** - размер вознаграждения.
- **Игровые деньги** - внутриигровой баланс UnicoreConnect Economy (монетки). Добавляет обязательные параметры:
  - **Количество** - размер вознаграждения.
  - **Сервер** - сервер на котором будет выдаваться вознаграждение.
- **Донат-группа** - добавляет обязательные параметры:
  - **Период** - время на которое будет выдаваться группа.
  - **Сервер** - сервер на котором будет выдаваться группа.
  - **Группа**
- **Донат-право** - добавляет обязательные параметры:
  - **Период** - время на которое будет выдаваться право.
  - **Сервер** - сервер на котором будет выдаваться право.
  - **Донат-право**
- **Товар** - товар из магазина. Добавляет обязательные параметры:
  - **Количество** - количество выдаваемого предмета.
  - **Сервер** - сервер на котором будет выдаваться товар.
  - **Товар**
- **Кит** - кит из магазина. Добавляет обязательные параметры:
  - **Сервер** - сервер на котором будет выдаваться кит.
  - **Кит**



Работа с ПУ



Голсование

# Голсование



Разработчикам



Расширение OAuth

# Расширение OAuth

Soon...



Разработчикам



Изменение клиента

# Изменение клиента

Soon...





Разработчикам



Ивенты

# Ивенты

Soon...



Разработчикам



Разрешения

# Разрешения



Разработчикам



Rest-API

# Rest-API

Soon...



Разработчикам



Вебхуки

# Вебхуки



Разработчикам



WebSocket-API

# WebSocket-API

Soon...