ООО «Мелвад»
390026, г. Рязань, ул. Ленинского Комсомола, д. 106, кв. 18
ИНН 6234193052, КПП 623401001
ОГРН 1206200011907
тел. 8 (920) 961-27-77
support@melvad.com

Руководство для проведения проверки эксплуатации Melvad DSP

Оглавление

1. Руководство для работы....4
1.1. Авторизация....4
1.2. Контакты техподдержки....4
2. Общее описание....5
2.1. Дашборд....5
2.2. Пользователи и роли....6
2.2.1. SSP....7
2.3. Договоры и контрагенты....9
2.4. Бренды....10
2.5. Платежи....11
2.6. Автоправила....12
2.7. Создание рекламной кампании....13
2.7.1. Создание рекламной кампании....13
2.7.2. Создание группы объявлений....15
2.7.3. Создание объявления....17
2.8. Отчеты....18
2.8.1. Шаблоны отчетов....19
2.9. Статистика....21
3. Руководство для проведения проверки эксплуатации ПО....25
3.1. Авторизация....25
3.2. Создание основных сущностей для запуска рекламы....25
3.2.1. Создание рекламодателя....25
3.2.2. Заведение договора....26
3.2.3. Баланс рекламодателя....28
3.2.4. Создание бренда....29
3.2.5. Создание рекламной кампании, группы объявления и объявления....30
3.2.6. Создание SSP....34
3.3. Запрос на получение рекламы....36
4. Описание развернутого экземпляра инфраструктуры ПО....39
4.1. Общее описание....39
4.2. Подробное описание контейнеров сервисов, их процессов и компонентов....40
4.2.1. UI....40
4.2.2. API....40
4.2.3. Bidder....41
4.2.4. ResCleaner....42
4.2.5. Currator....42
4.2.6. PostgreSQL....42

4.2.7. KeyDB 43
4.2.8. ClickHouse 43
4.2.9. Kafka 43
4.2.10. nginx и lego 44
4.3. Настройка удаленного доступа к развернутому экземпляру ПО 44
4.4. Расположение файлов приложения 45
4.4.1. Список volumes 45
4.4.2. Смонтированные каталоги хоста 46
Приложение 1. Общая схема контейнеров и движения данных между ними 47

1. Руководство для работы

1.1. Авторизация

Для ознакомления с работой Melvad DSP необходимо авторизоваться в личном кабинете, для этого перейдите по ссылке **https://ui.dsp.melvad.com** и:

- Укажите электронный адрес: **review@dsp.melvad.com**
- Введите пароль: **CkLHXtfa77LCPfQTKC**

Учётная запись имеет роль «Администратор» и доступ ко всем разделам. Сессия действует 30 дней, после чего вход нужно повторить. Кнопка «Выйти» находится в меню пользователя в правом верхнем углу.

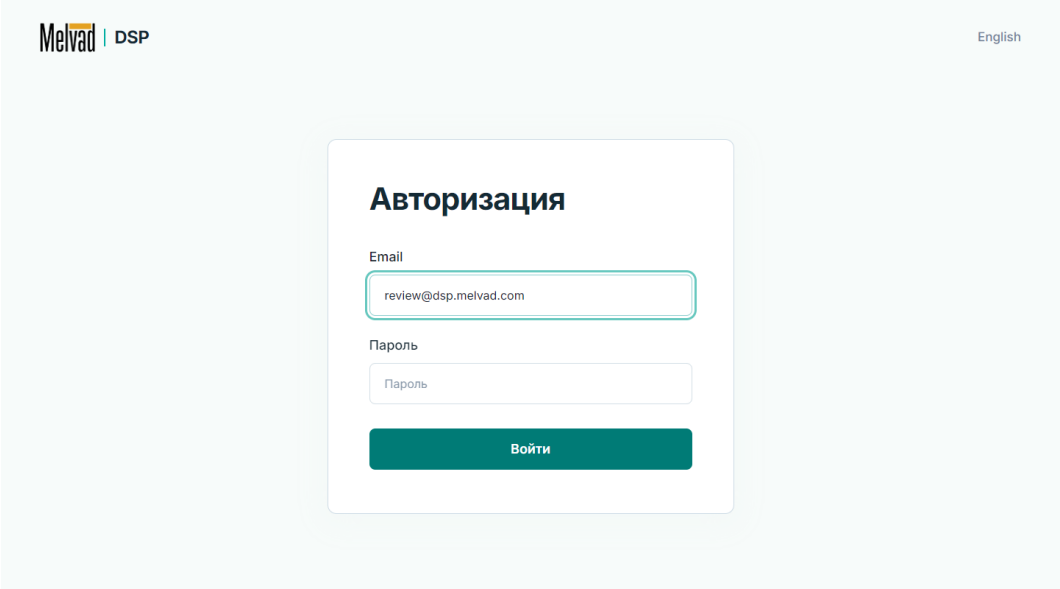


Рисунок 1 — Экран входа

1.2. Контакты техподдержки

Стуколкин Александр

e-mail: a.stukolkin@melvad.com

Телефон: +7(920) 961-27-77

2. Общее описание

Melvad DSP (Demand-Side Platform) — платформа закупки рекламы для рекламодателей. Рекламодатель заводит договор, бренд и рекламную кампанию с группами объявлений и креативами, пополняет баланс, а RTB-биддер DSP в реальном времени отвечает на запросы SSP (рекламных бирж) по протоколу OpenRTB 2.5 и покупает показы, подходящие под таргетинги групп. Статистика показов, кликов и расходов собирается автоматически, сведения о рекламе передаются в ОРД.

Меню личного кабинета: «Дашборд», «Бренды», «Кампании» (группы объявлений и креативы открываются из кампании), «Статистика», «Автоправила», «Платежи», «Договоры», «SSP», «Пользователи». Во всех списках есть поиск (строка «Найти» в шапке), сортировка по колонкам, вкладки по статусу (все, активные, выключенные, удалённые) и постраничный вывод; статус записи меняется переключателем в колонке «Статус», редактирование — значком карандаша в колонке «Действия». Клик по названию записи открывает вложенные записи: кампании — группы, группы — креативы.

2.1. Дашборд

Дашборд включает в себя следующие показатели:

- Группировка по периоду: сегодня, вчера, последние 7 дней, последние 30 дней или произвольный период (до 366 дней);
- Метрики администратора: показы, клики, доход, прибыль, расход (выплата SSP); метрики рекламодателя: показы, клики, CTR, конверсии, расход;
- График показов, кликов и денег за период: до 2 дней — по часам, дольше — по дням.

Эта информация позволяет пользователю оценить общую информацию о ходе рекламных кампаний. Рекламодатель видит только свои кампании, администратор — все.

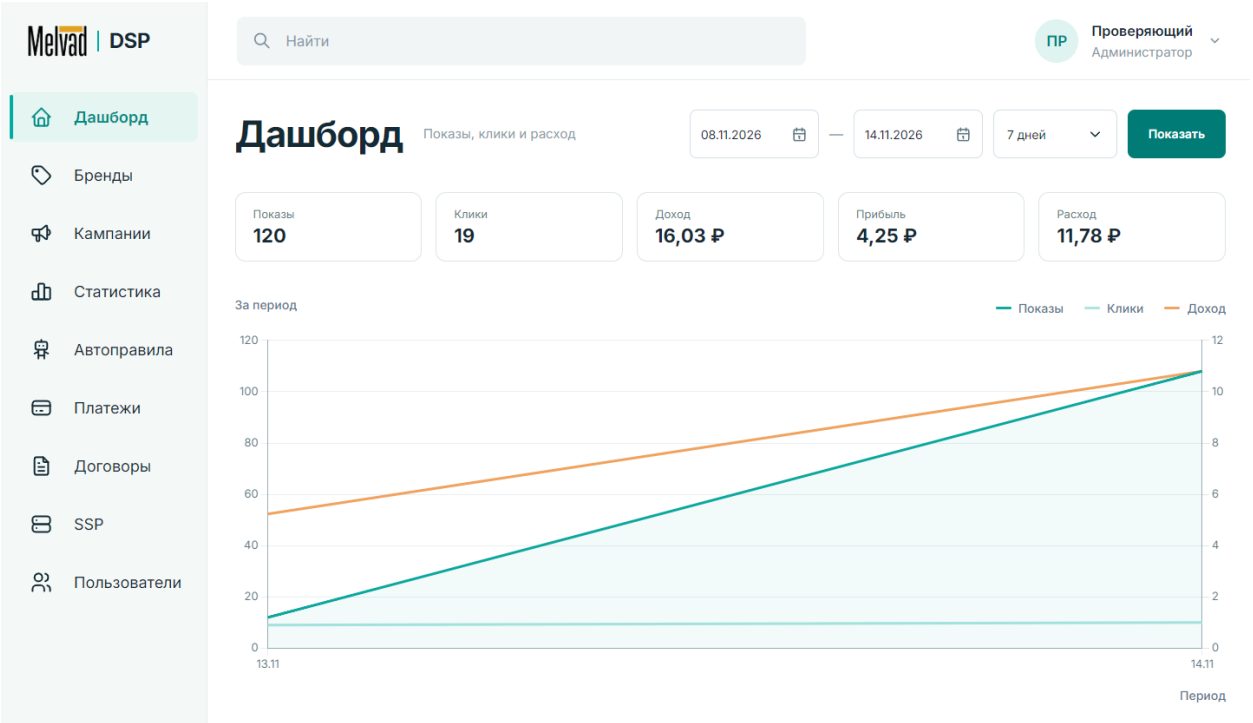


Рисунок 2 — Дашборд

Чтобы изменить период, выберите его в выпадающем списке или укажите даты и нажмите кнопку «Показать». Наведение курсора на график показывает значения за час или день.

2.2. Пользователи и роли

В Melvad DSP есть две роли — администратор и рекламодатель.

Администратор — сотрудник оператора DSP. Он управляет всеми сущностями: пользователями, SSP, договорами, брендами, кампаниями, автоправилами и балансами всех рекламодателей, видит все отчёты, а также доход и прибыль платформы.

Рекламодатель — это пользователь системы, который непосредственно запускает рекламные кампании. Вместе с пользователем с ролью «Рекламодатель» создаётся рекламодатель: валюта, баланс и реквизиты (тип контрагента — физическое лицо, юридическое лицо или ИП, название, ИНН). По сохранённым реквизитам рекламодатель автоматически регистрируется в ОРД; в списке пользователей видны его «ОРД статус» и «ЕРИР статус». Рекламодатель видит только свои договоры, бренды, кампании, автоправила, платежи и статистику; разделы «SSP» и «Пользователи» ему недоступны.

Список пользователей можно посмотреть на вкладке «Пользователи».

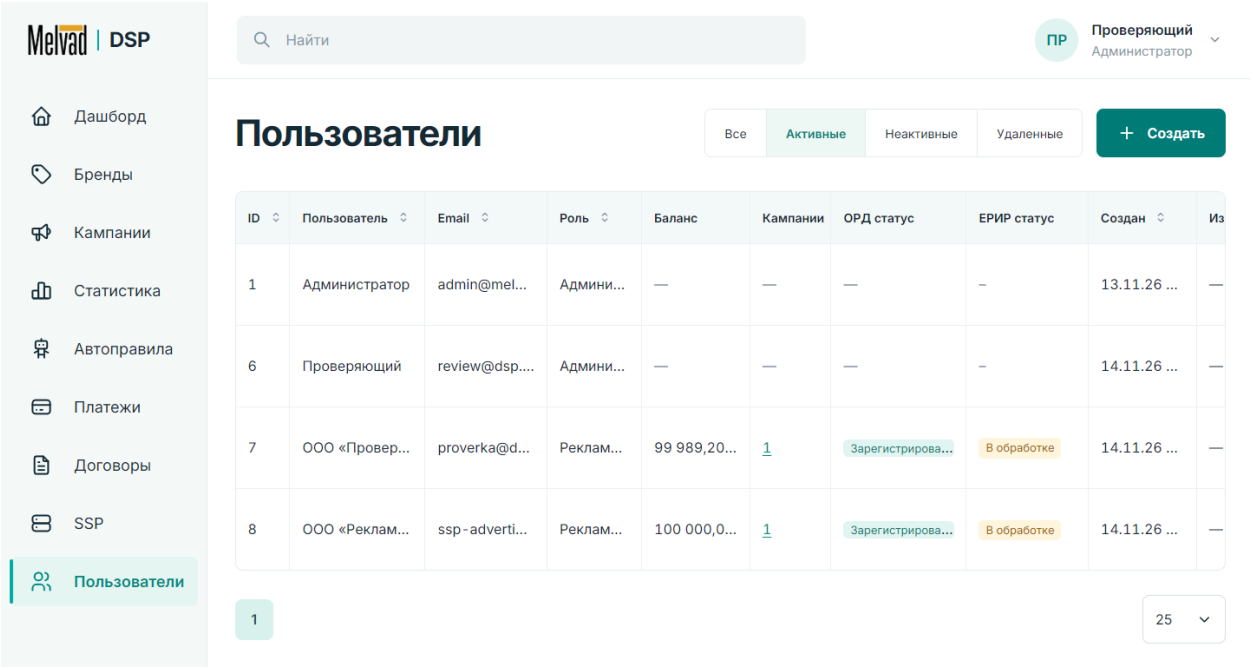


Рисунок 3 — Пользователи

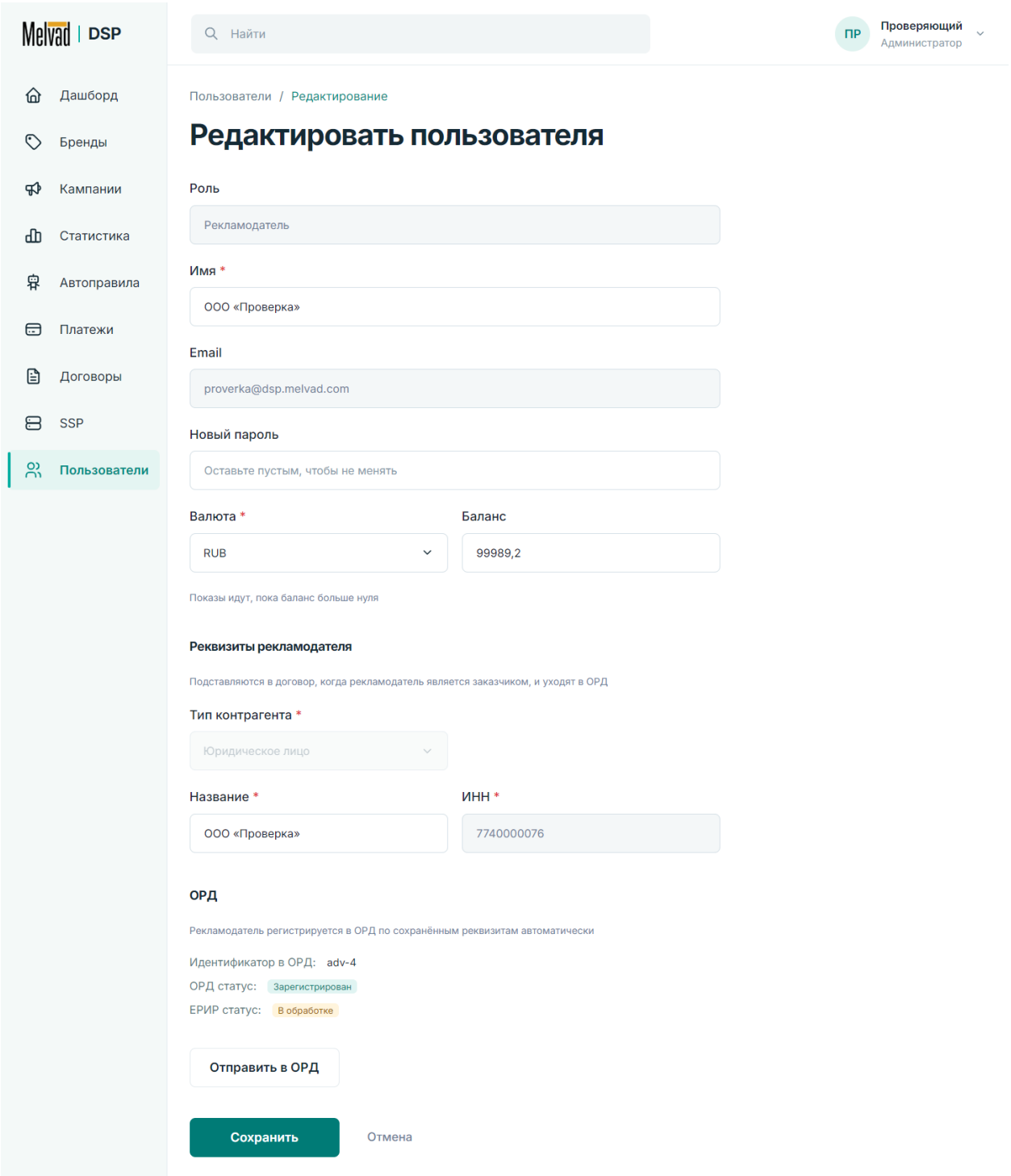


Рисунок 4 — Карточка пользователя с ролью «Рекламодатель»: реквизиты и регистрация в ОРД

2.2.1. SSP

SSP — это партнёры (компании), которые отправляют в Melvad DSP рекламные запросы с целью дальнейшей их продажи по аукционной модели. У каждой SSP есть уникальное имя (латиницей), домен, необязательный макрос для клика, признаки оплаты («Оплата за один показ», «Платёжная ссылка не поддерживается») и реквизиты (тип контрагента, ИНН, наименование). SSP регистрируется в ОРД как рекламная система с площадкой (кнопка «Зарегистрировать в ОРД» в списке).

Для каждой SSP DSP формирует RTB-ссылку вида https://bid.dsp.melvad.com/ssps/1/<имя SSP>, на которую SSP отправляет OpenRTB-запросы методом POST. Ссылка открывается кнопкой «Конечные точки» (значок </>) в списке SSP. В колонке «Расход» показана сумма

выплат SSP за последний год. На стенде подключена, в том числе, SSP Melvad (melvad-ssp) — собственная рекламная биржа Melvad, которая получает рекламу из этой DSP.

Список SSP можно посмотреть на вкладке «SSP».

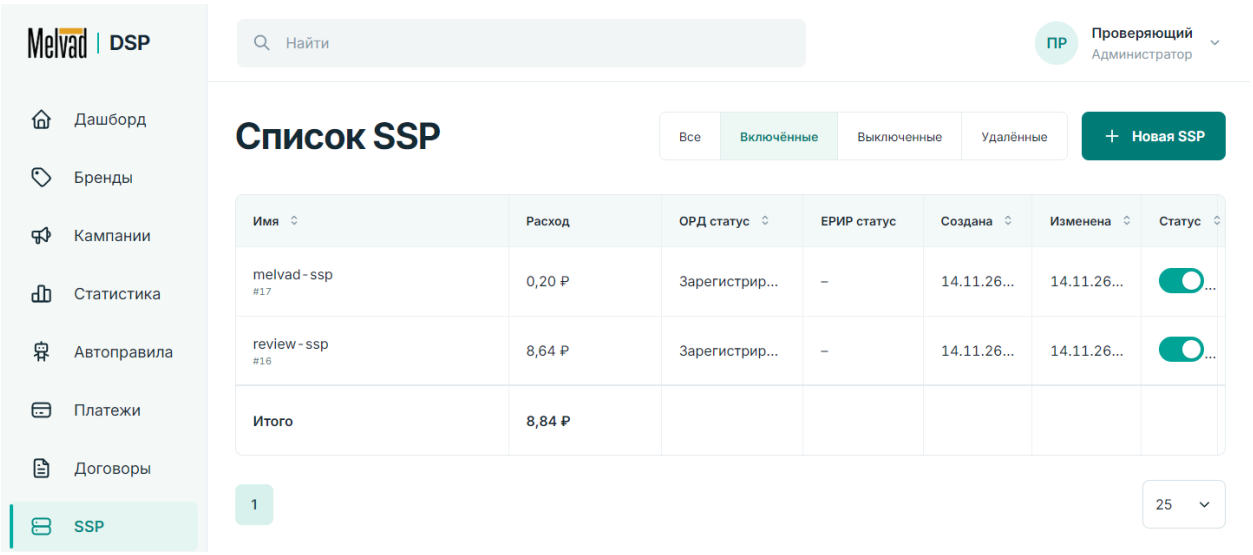


Рисунок 5 — Список SSP

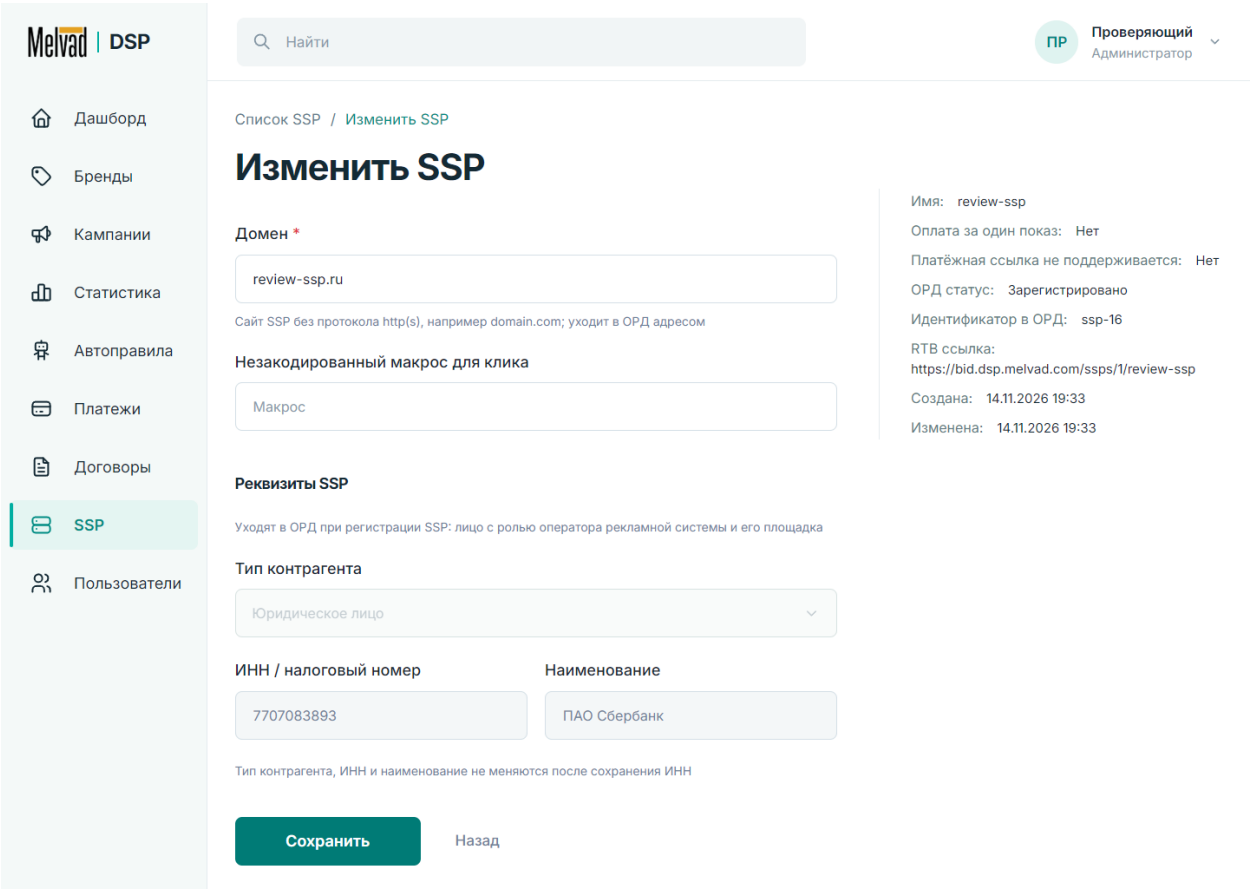


Рисунок 6 — Карточка SSP

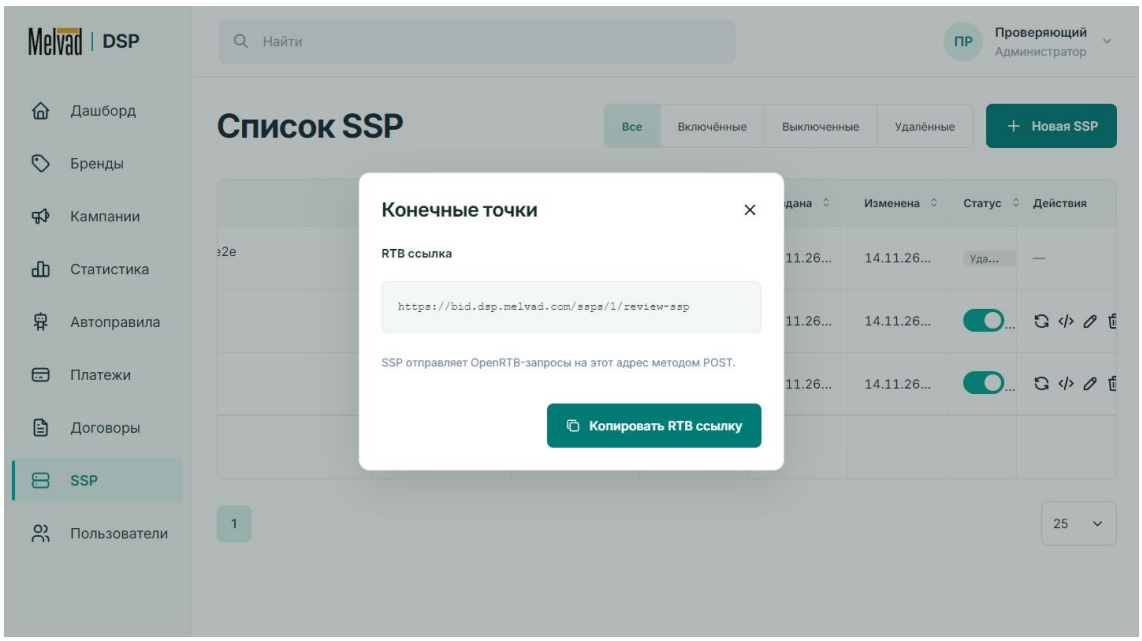


Рисунок 7 — RTB-ссылка SSP («Конечные точки»)

2.3. Договоры и контрагенты

Согласно Федеральному закону №38-ФЗ от 13.03.2006 г. «О рекламе» рекламодатель по требованию рекламораспространителя обязан предоставлять документально подтверждённые сведения о соответствии рекламы требованиям настоящего Федерального закона, а сведения о рекламе в сети «Интернет» (о рекламодателях, договорах, креативах и актах) передаются в Единый реестр интернет-рекламы (ЕРИР) через операторов рекламных данных (ОРД).

Melvad DSP является оператором рекламной системы (ОРС) и самостоятельно предоставляет всю информацию о рекламе оператору рекламных данных — ОРД VK (на проверочном стенде используется тестовая среда ОРД VK). Каждому креативу ОРД присваивает токен erid, который DSP передаёт SSP вместе с рекламой.

Рекламодатель (или администратор за рекламодателя) заводит в Melvad DSP договор, на основании которого осуществляется рекламная деятельность. Отдельного справочника контрагентов нет: стороны договора указываются в самом договоре. Первая сторона — заказчик: флажок «Рекламодатель является заказчиком» подставляет реквизиты рекламодателя из его карточки, иначе вводятся тип контрагента, название и ИНН заказчика (рекламодатель или рекламное агентство). Вторая сторона — исполнитель: «Мелвад» (оператор DSP) или рекламное агентство. Типы договора — договор оказания услуг, посреднический договор и доп. соглашение (к основному договору); подтип задаёт предмет договора (распространение рекламы, организация распространения, представительство, посредничество, иное).

Кнопка «Отправить в ОРД» в списке договоров регистрирует в ОРД обе стороны и договор; колонки «ОРД статус» и «ЕРИР статус» показывают результат. После регистрации в договоре меняется только название, остальные сведения показаны справа в карточке. Акты вручную не создаются.

Список договоров можно посмотреть на вкладке «Договоры».

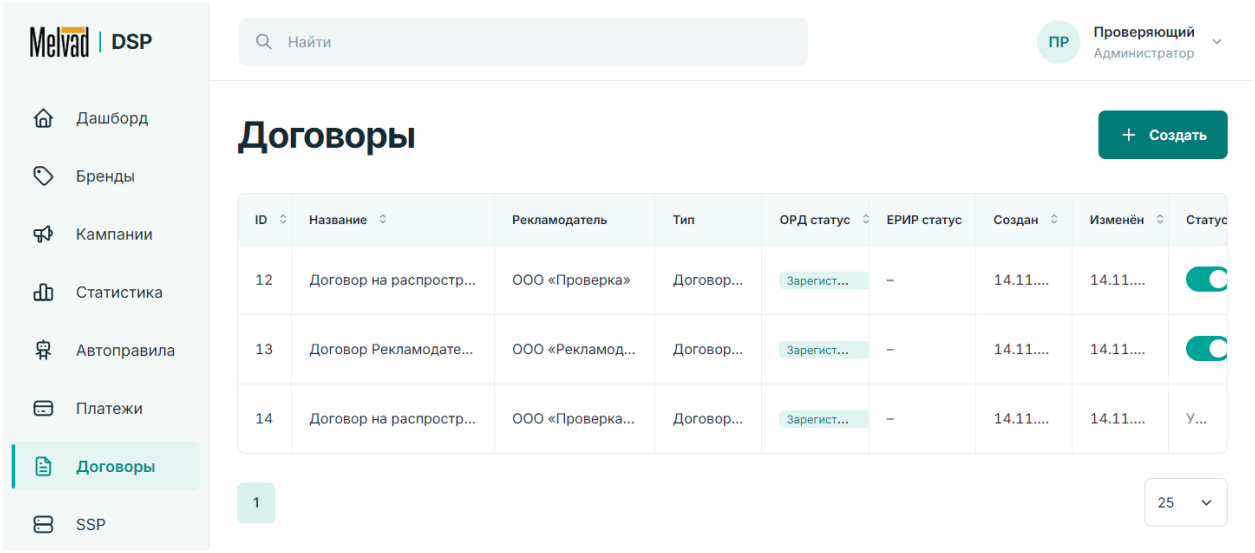


Рисунок 8 — Договоры

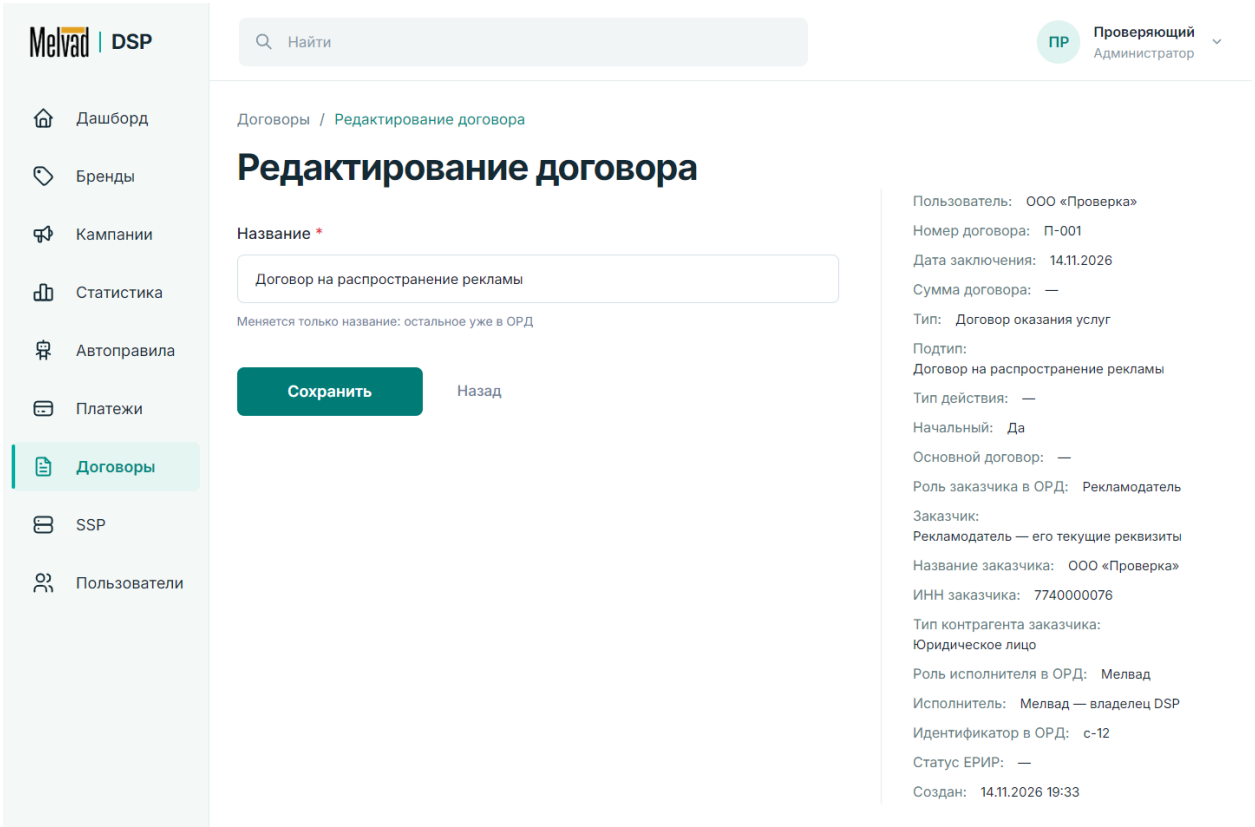


Рисунок 9 — Карточка зарегистрированного договора

2.4. Бренды

Бренд объединяет рекламные кампании одного рекламодателя по объекту рекламирования. У бренда указывается рекламодатель, название и категория ККТУ — код классификатора товаров и услуг ОРД (выбирается только категория третьего уровня из дерева с поиском). ККТУ бренда передаётся в ОРД вместе с креативами его кампаний. Каждая кампания обязательно привязана к бренду своего рекламодателя.

Список брендов можно посмотреть на вкладке «Бренды».

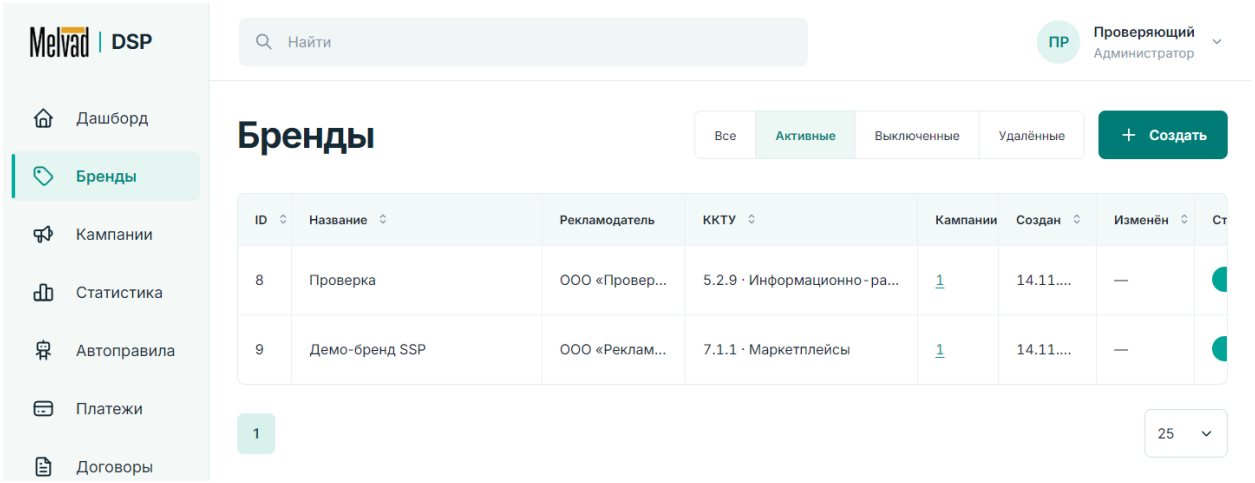


Рисунок 10 — Бренды

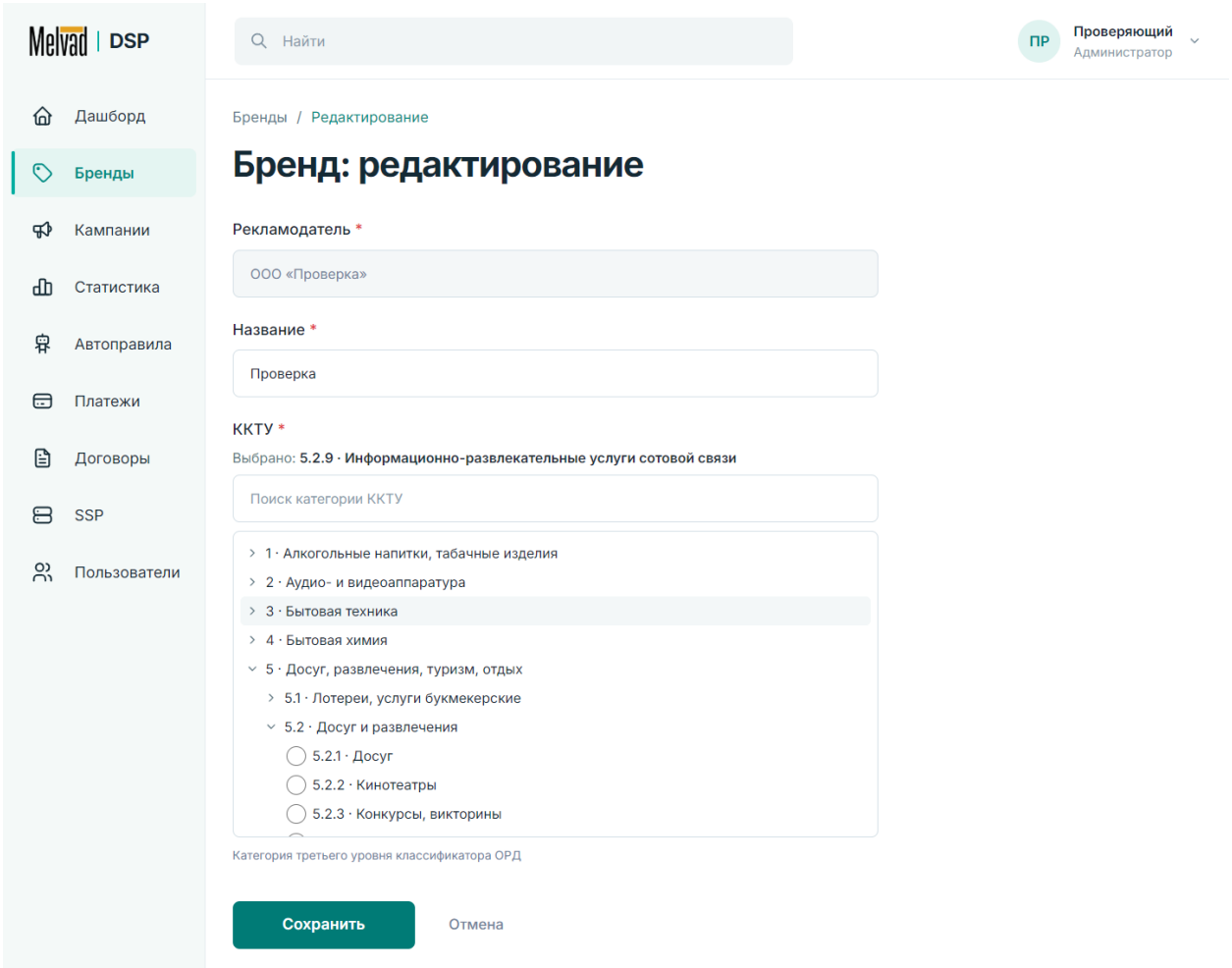


Рисунок 11 — Бренд с категорией ККТУ

2.5. Платежи

Реклама рекламодателя участвует в аукционах, пока его баланс больше нуля. Баланс пополняет или уменьшает администратор кнопкой «Изменить баланс» (пополнение только ручное, оплаты картой нет). Раздел «Платежи» показывает за выбранный период историю операций: дату, рекламодателя, поступления, списания и описание; у администратора есть фильтры по рекламодателю и операции, у рекламодателя — движения его баланса и списания по кампаниям по дням. Внизу таблицы — итоги за период.

Список операций можно посмотреть на вкладке «Платежи».

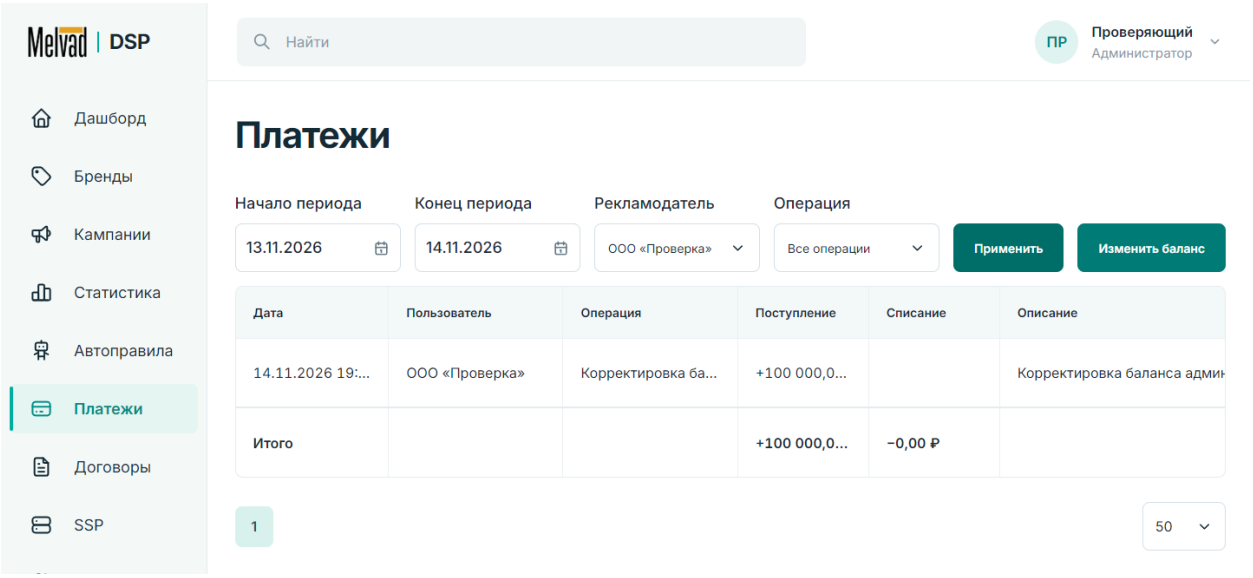


Рисунок 12 — Платежи

2.6. Автоправила

Автоматические правила позволяют настроить условия по выбранным метрикам с целью оптимизации закупки рекламных запросов на основе статистических данных. Правило применяется к выбранным кампаниям, группам объявлений или объявлениям рекламодателя и выполняет одно из действий: выключить, повысить лимит по показам, повысить бюджет, повысить дневной бюджет, изменить ставку или добавить источник (сайт или мобильное приложение) в чёрный список. Условия задаются группами «Метрика / Условие / Значение», соединёнными через «И» или «ИЛИ» (метрики: показы, клики, конверсии, расход, CTR, VTR, средняя стоимость за тысячу показов, за клик, за конверсию, события видео). Частота проверки — от каждых 30 минут до каждых 24 часов, период данных — от последних 24 часов до последних 60 дней. В карточке правила видна история срабатываний и заблокированные площадки.

Список всех автоправил можно посмотреть на вкладке «Автоправила».

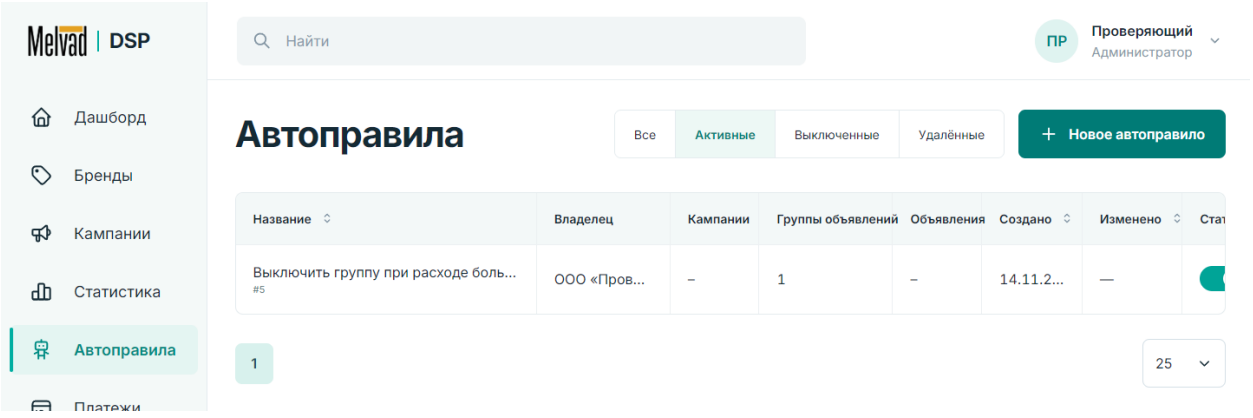


Рисунок 13 — Автоправила

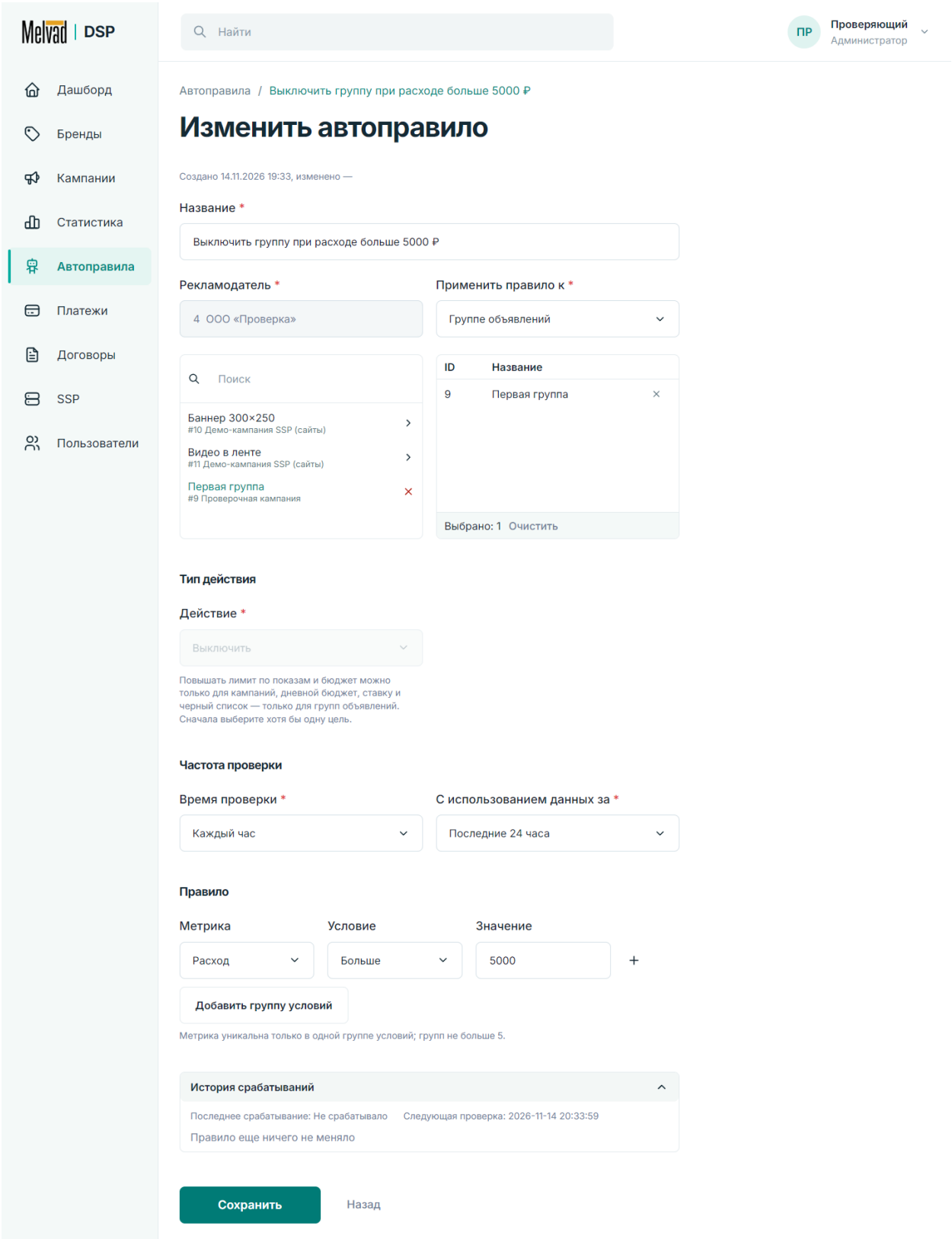


Рисунок 14 — Карточка автоправила

2.7. Создание рекламной кампании

2.7.1. Создание рекламной кампании

Создание рекламной кампании включает в себя выбор рекламодателя, договора, на основании которого проводится рекламная кампания, бренда, типа кампании (сайт или

приложение), конечного URL (адрес, куда ведёт реклама) и периода (дата начала и необязательная дата окончания).

Все рекламные кампании принадлежат рекламодателю. По договору кампании её креативы регистрируются в ОРД. Кнопка «Сохранить и продолжить» сохраняет кампанию и сразу открывает создание группы объявлений.

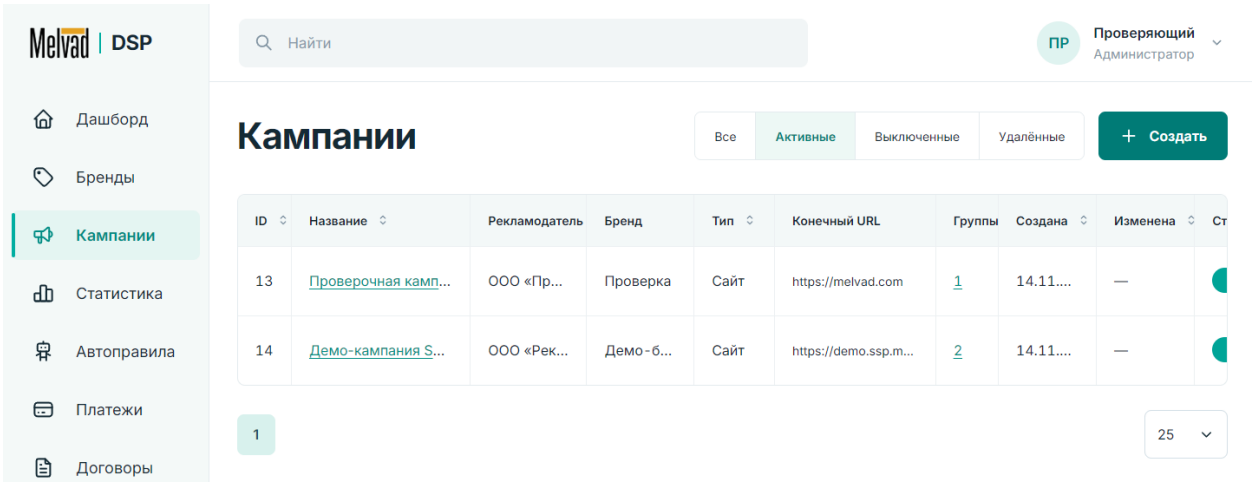


Рисунок 15 — Кампании

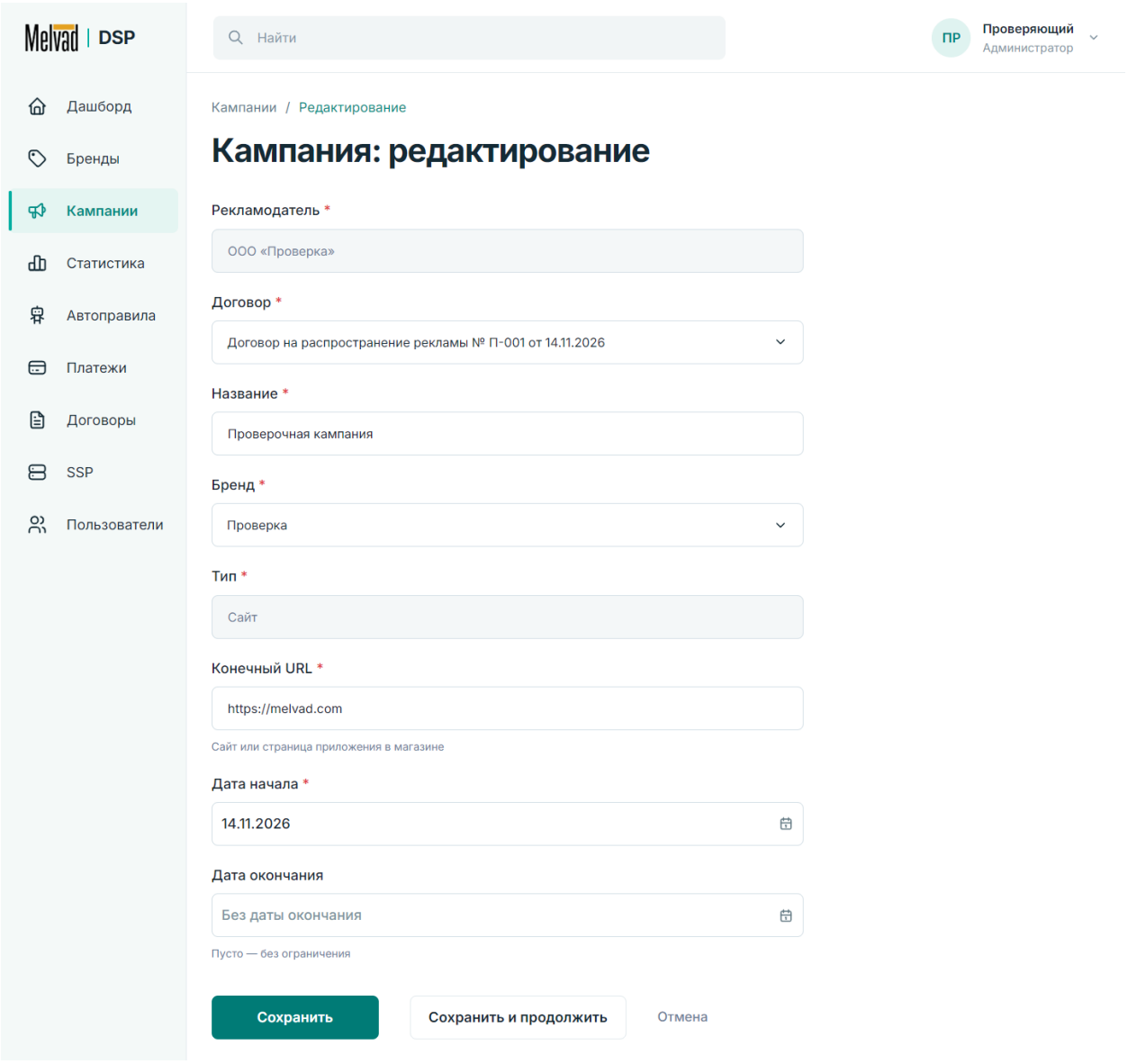


Рисунок 16 — Карточка кампании

2.7.2. Создание группы объявлений

Группа объявлений является промежуточным звеном, в котором задаются основные настройки, такие как цена закупки и параметры целевой аудитории.

Одна рекламная кампания может содержать несколько групп объявлений с целью проведения различных гипотез по поиску лучшей целевой аудитории.

Для создания группы объявлений необходимо задать название, ставку (CPM — цена за тысячу показов) и дату начала (не раньше начала кампании). Частота показов опциональна и служит для ограничения показов одному пользователю за период (день, неделя, месяц или всё время). Бюджет в день также опционален и служит для ограничения денежных трат за день; флажок «Равномерное распределение» распределяет дневной бюджет по часам расписания.

Также можно указать следующие таргетинги (позволяют настроить более релевантную аудиторию; пустой список — любые значения, выбранные значения можно включить или исключить):

- Таргетинг по географии (страна, регион, город)
- Таргетинг по типу устройства

- Таргетинг по операционной системе
- Таргетинг по браузеру
- Таргетинг по типу соединения
- Таргетинг по сотовому оператору
- Таргетинг по языку
- Таргетинг по категории площадок
- Таргетинг по позиции на странице
- Таргетинг по среде показа
- Таргетинг по SSP (бирже)
- Таргетинг по часам и дням (расписание показов)

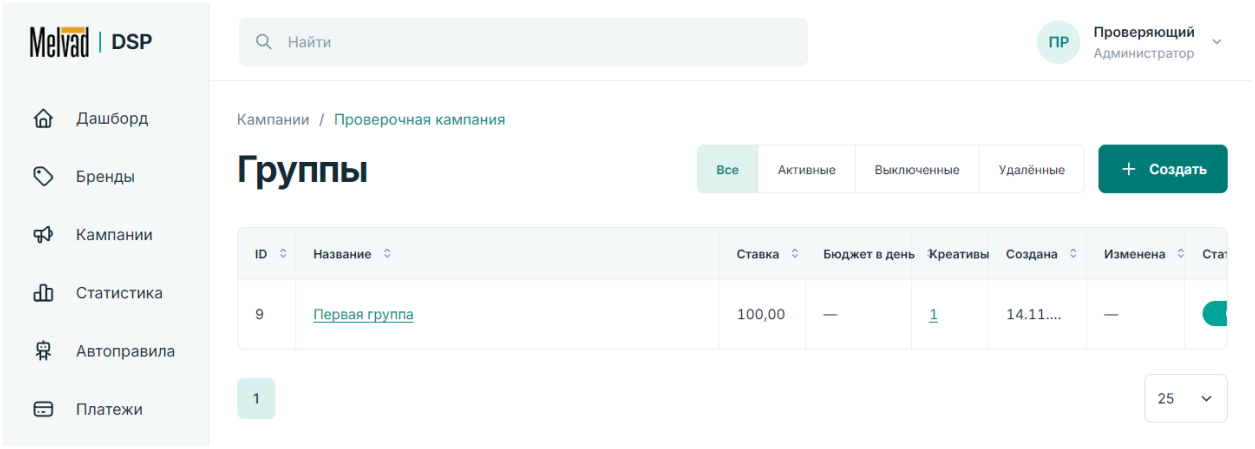


Рисунок 17 — Группы объявлений кампании

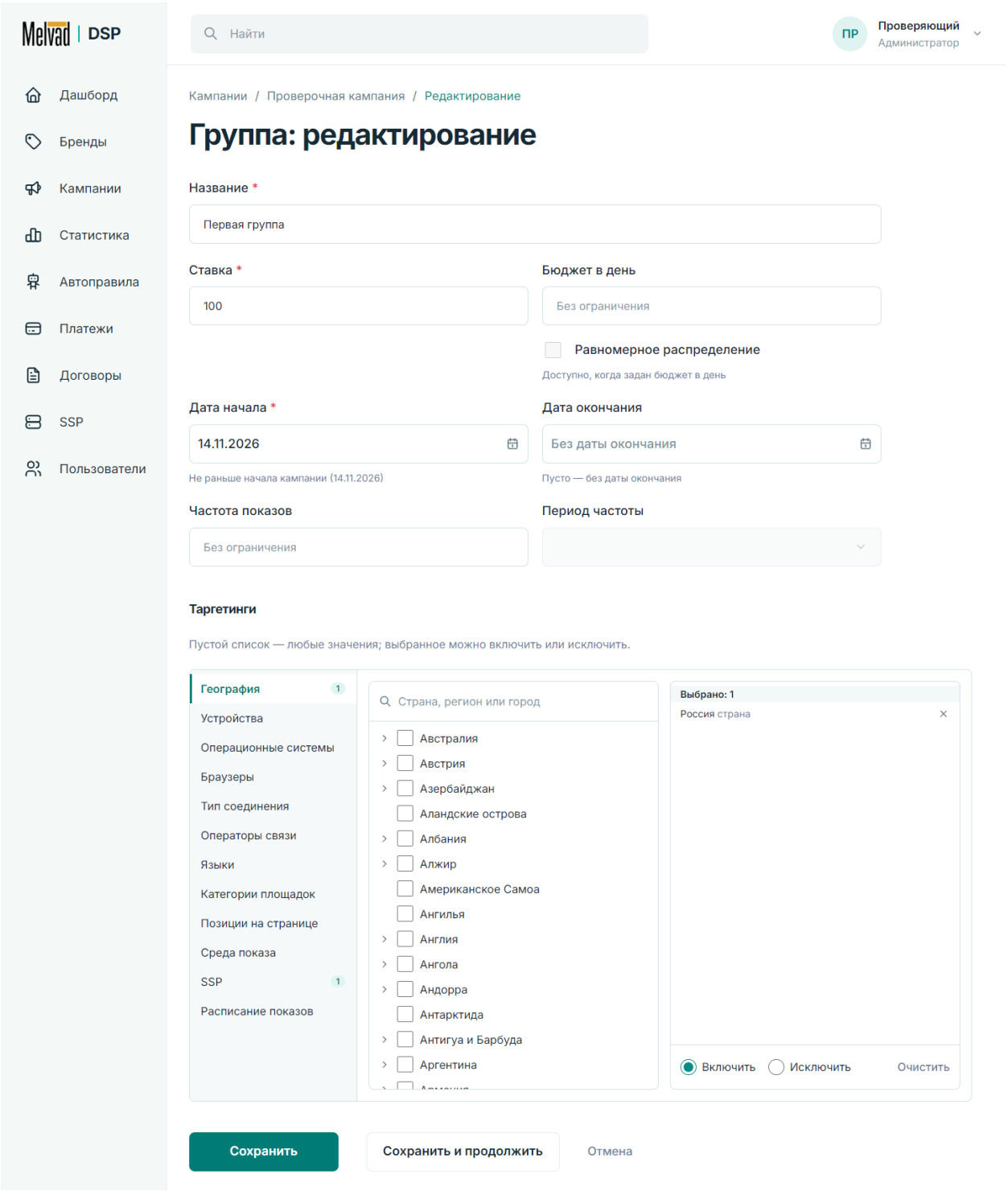


Рисунок 18 — Карточка группы объявлений

2.7.3. Создание объявления

Объявление (креатив) может быть четырёх типов:

1 – Баннер-изображение (картинка PNG, JPEG или GIF до 2 Мбайт, на которой размещён рекламный текст)

2 – HTML-баннер (код баннера с макросом ссылки клика)

3 – Нативное (отдельные компоненты — картинка, заголовок, описание, которые выполнены в стиле сайта или приложения)

4 – Видео (видеоролик MP4 или WebM до 30 Мбайт, который показывается в видеоплеере)

При создании объявления нужно указывать кликовую ссылку, куда при клике на рекламу перейдёт пользователь (под полем показан конечный URL кампании). Справа отображается превью объявления.

Токен erid вручную не вводится: после сохранения креатив автоматически отправляется в ОРД VK вместе с договором кампании, ККТУ бренда и ставкой группы, и ОРД присваивает ему erid. Статус регистрации, erid и кнопка «Отправить в ОРД» показаны в блоке «ОРД» карточки креатива, в списке — колонки «ОРД статус» и «ЕРИР статус». Биддер показывает только креативы, зарегистрированные в ОРД.

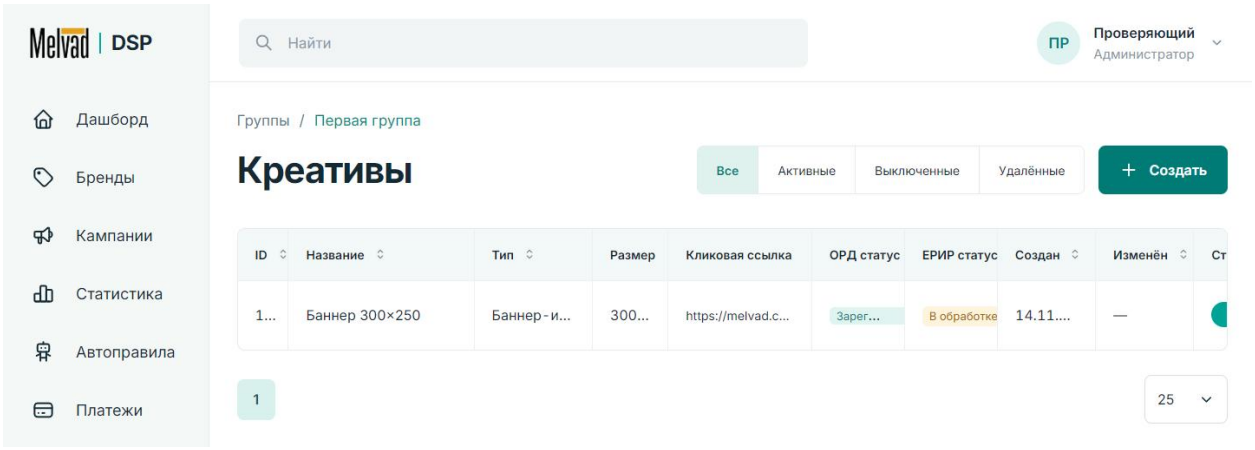


Рисунок 19 — Креативы группы объявлений

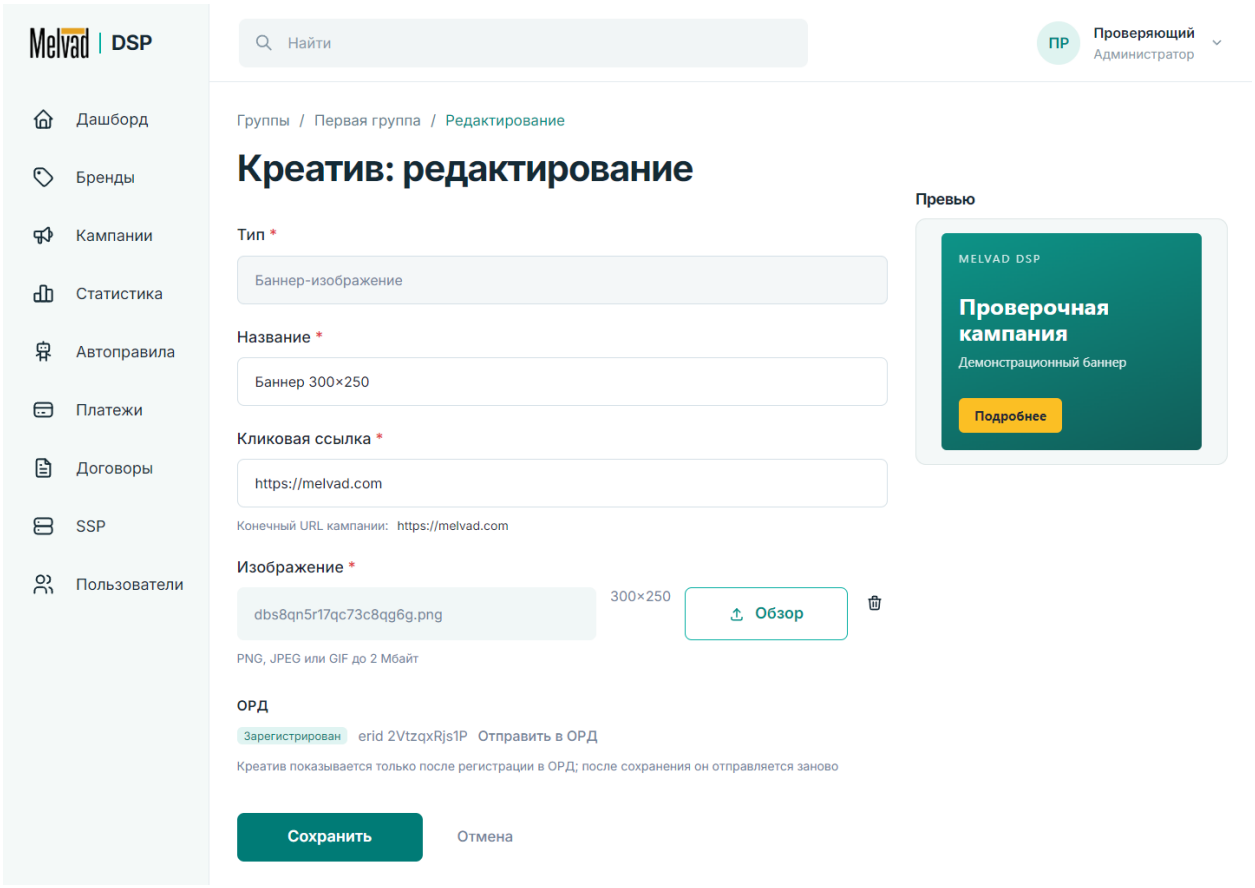


Рисунок 20 — Карточка креатива с erid

2.8. Отчеты

Раздел «Статистика» содержит список шаблонов отчётов пользователя: название, владелец (у администратора), валюта, даты создания и изменения. Звёздочка добавляет шаблон в

избранное, кнопка «Сформировать» открывает отчёт, кнопки «CSV» и «XLSX» скачивают файл отчёта по шаблону.

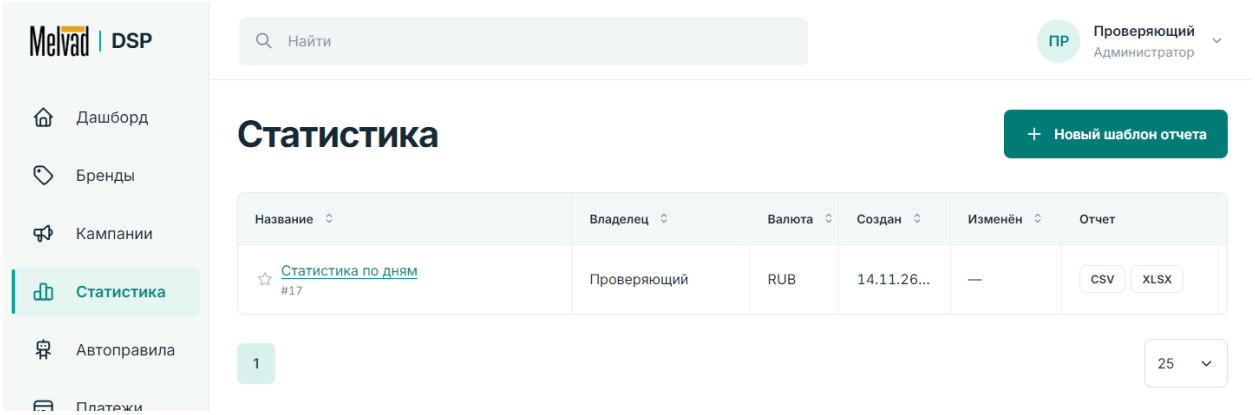


Рисунок 21 — Список шаблонов отчётов

2.8.1. Шаблоны отчетов

Шаблоны отчётов позволяют единовременно настроить нужные параметры (название, валюту, период, часовой пояс, группировки, метрики, порядок колонок, сортировку и фильтры) для получения статистики без постоянного выбора одних и тех же значений. На вкладке «Колонки» доступные группировки и метрики перечислены слева, выбранные — справа (порядок меняется перетаскиванием, у каждой колонки задаётся сортировка). На вкладке «Фильтры» задаются фильтры по группировкам и условия по метрикам (=, ≠, >, ≥, <, ≤). Периоды: сегодня, вчера, позавчера, последние 7 дней, этот месяц, прошлый месяц, этот год, прошлый год, всё время, произвольный период.

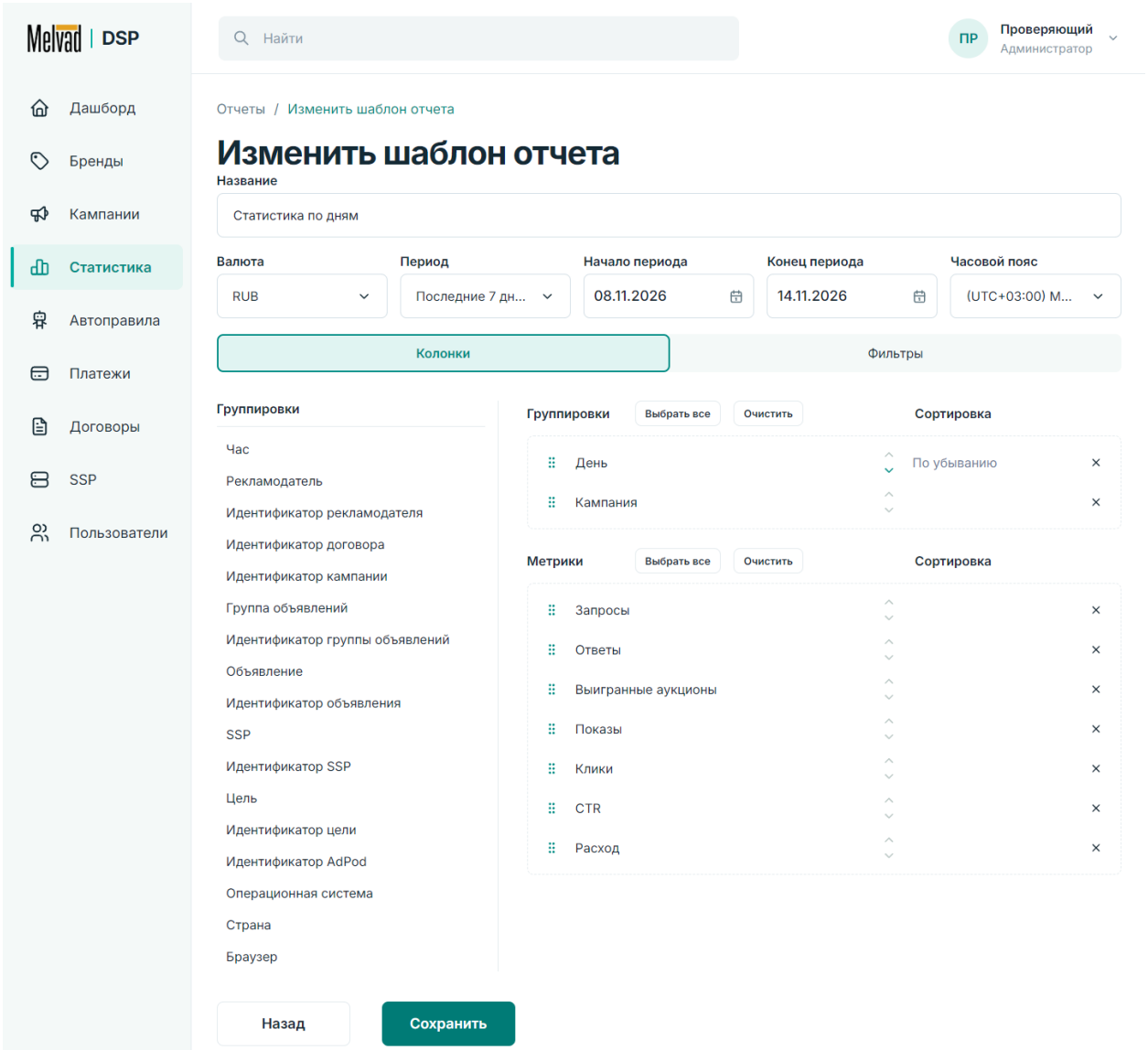


Рисунок 22 — Шаблон отчёта: вкладка «Колонки»

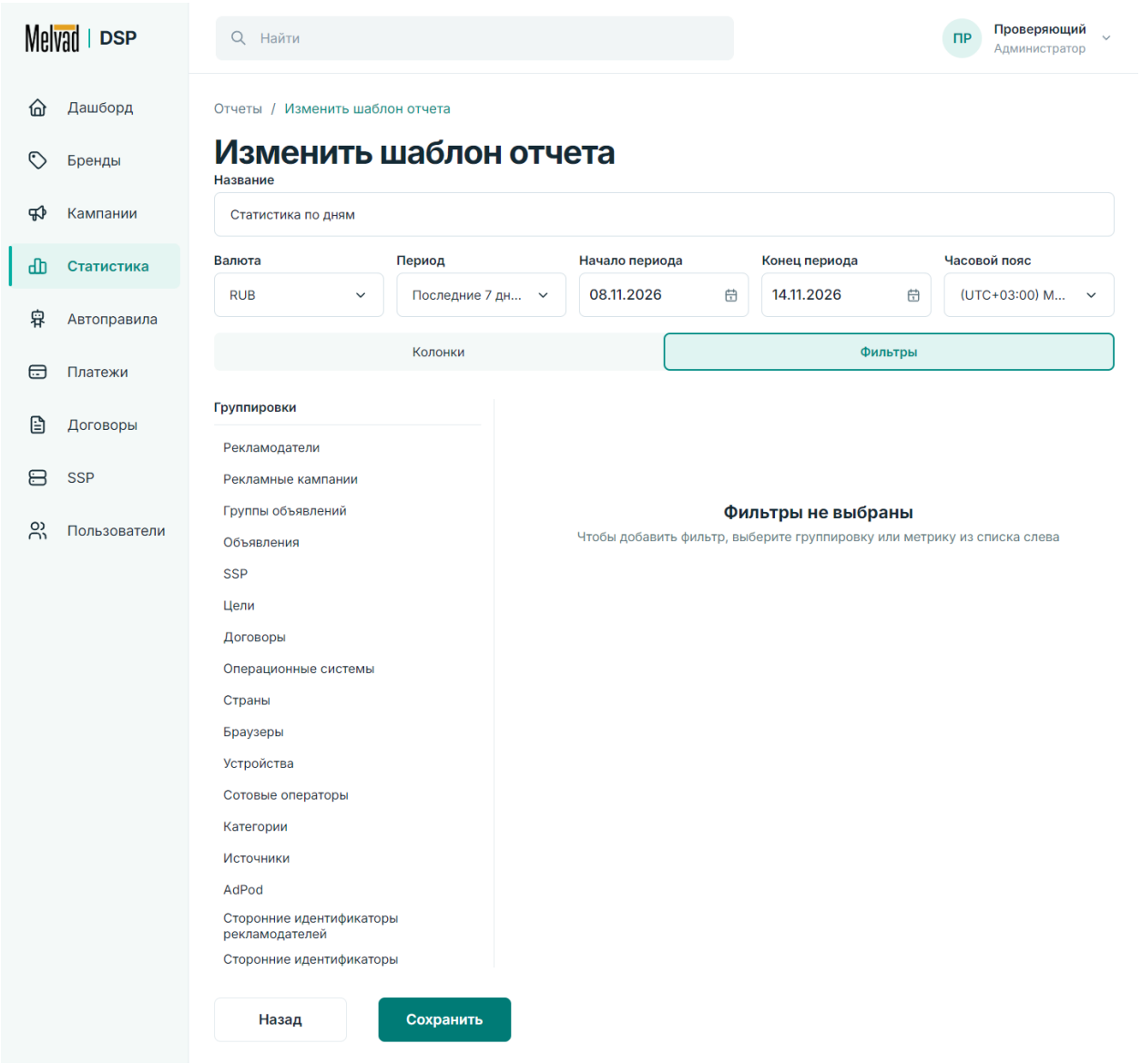


Рисунок 23 — Шаблон отчёта: вкладка «Фильтры»

2.9. Статистика

Отчёт по шаблону показывает график выбранной метрики, таблицу с итогами и выгрузку в CSV; период, валюту и часовой пояс можно поменять перед нажатием «Применить». Статистика содержит следующие группировки, метрики и фильтры (метрики «Выплата SSP» и «Прибыль» и группировка по рекламодателю доступны только администратору).

Группировки:

- Час
- День
- Рекламодатель
- Идентификатор рекламодателя
- Идентификатор договора
- Кампания
- Идентификатор кампании
- Группа объявлений
- Идентификатор группы объявлений

- Объявление
- Идентификатор объявления
- SSP
- Идентификатор SSP
- Цель
- Идентификатор цели
- Идентификатор AdPod
- Операционная система
- Страна
- Браузер
- Производитель
- Тип устройства
- Сотовый оператор
- Категория
- Источник
- Сторонний идентификатор рекламодателя
- Сторонний идентификатор объявления
- Erid

Метрики:

- Запросы
- Ответы
- Выигранные аукционы
- Нет контента
- Таймауты
- Заглушка
- Показы
- Клики
- Уникальные клики
- Конверсии
- Расход
- Выплата SSP
- Прибыль
- Ср. стоимость за тысячу показов
- Ср. стоимость за клик
- Ср. стоимость за конверсию
- CTR
- FillRate
- BidRate
- WinRate
- VTR
- CR
- Start
- FirstQuartile

- Midpoint
- ThirdQuartile
- Complete
- Skip
- Close
- CreativeView
- Pause
- Resume
- Mute
- Unmute
- Errors

Фильтры:

- Рекламодатели
- Рекламные кампании
- Группы объявлений
- Объявления
- SSP
- Цели
- Договоры
- Операционные системы
- Страны
- Браузеры
- Устройства
- Сотовые операторы
- Категории
- Источники
- AdPod
- Сторонние идентификаторы рекламодателей
- Сторонние идентификаторы объявлений
- Erid

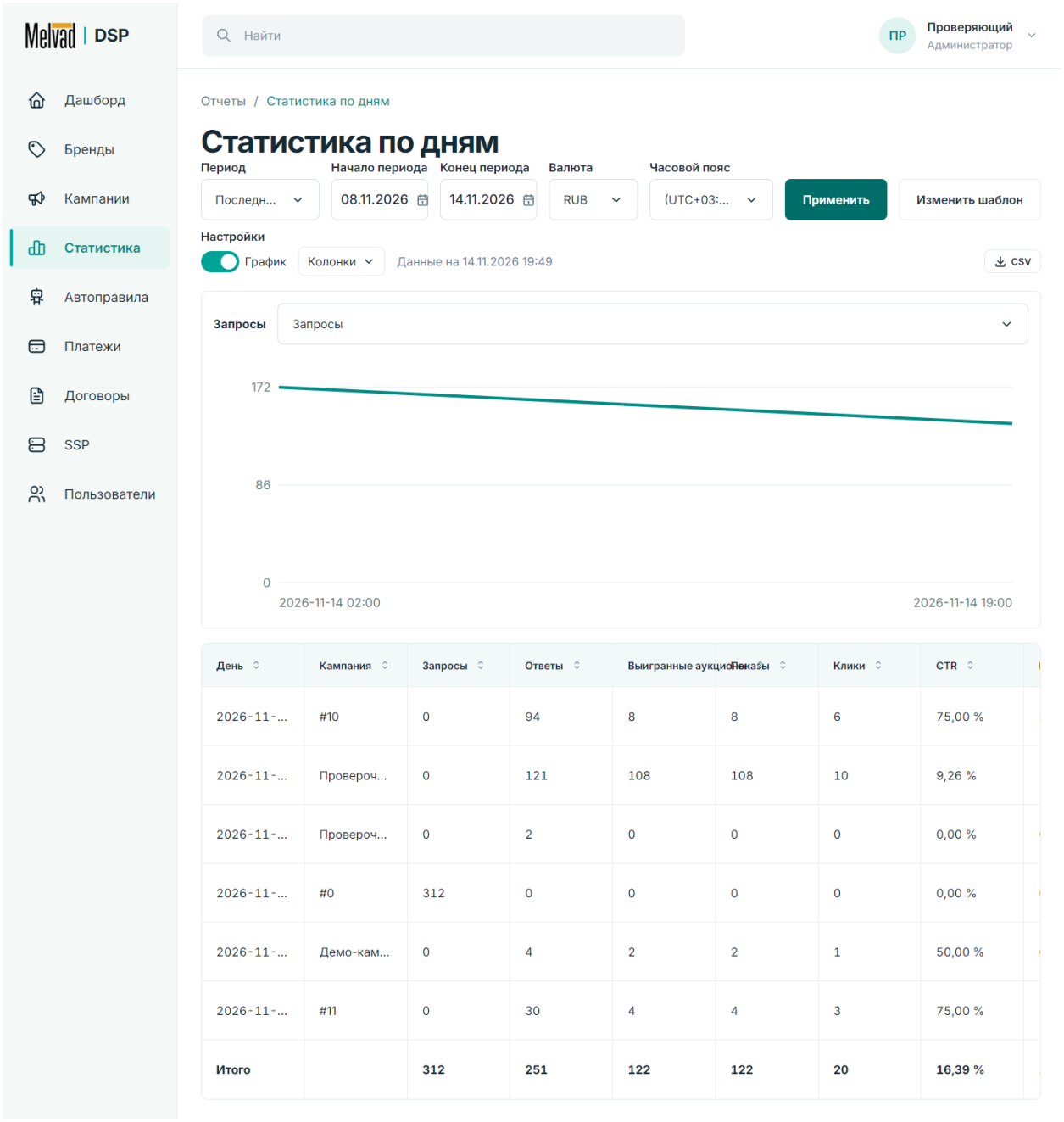


Рисунок 24 — Отчёт по шаблону

3. Руководство для проведения проверки эксплуатации ПО

Ниже — последовательность действий, которая проверяет весь путь рекламы: создание рекламодателя, договора и бренда, регистрацию в ОРД, пополнение баланса, создание кампании, группы объявлений и объявления, создание SSP и получение рекламы по OpenRTB-запросу от имени этой SSP.

На стенде уже есть пример таких сущностей, созданный для проверяющих: рекламодатель «ООО «Проверка»», договор «Договор на распространение рекламы» № П-001, бренд «Проверка», кампания «Проверочная кампания» с группой «Первая группа» и баннером 300×250, SSP «review-ssp», автоправило и шаблон отчёта «Статистика по дням», а также тестовый трафик для дашборда и статистики. Для проверки создайте собственные сущности со значениями ниже (названия отличаются от примера). Имя SSP и email пользователя уникальны: если значение из примера уже занято (ошибка «already exists»), укажите любое другое.

3.1. Авторизация

Перейдите по ссылке **https://ui.dsp.melvad.com**

Для авторизации вам понадобится:

- Пользователь: **review@dsp.melvad.com**
- Пароль: **CkLHXtfa77LCPfQTKC**

3.2. Создание основных сущностей для запуска рекламы

3.2.1. Создание рекламодателя

Для создания рекламодателя нужно перейти в меню «Пользователи» и нажать кнопку «Создать», затем выбрать роль и заполнить поля:

- Роль – Рекламодатель
- Имя – ООО «Проверка 2»
- Email – ivanov@example.com (любой адрес, письмо не отправляется)
- Пароль – Check-2026! (не короче 8 символов)
- Валюта – RUB
- Баланс – оставить пустым (баланс пополняется в п. 3.2.3)
- Тип контрагента – Юридическое лицо
- Название – ООО «Проверка 2»
- ИНН – 7736207543

Далее нажать кнопку «Сохранить». Рекламодатель сам регистрируется в ОРД по реквизитам: через несколько секунд в списке пользователей в колонке «ОРД статус» появится «Зарегистрирован».

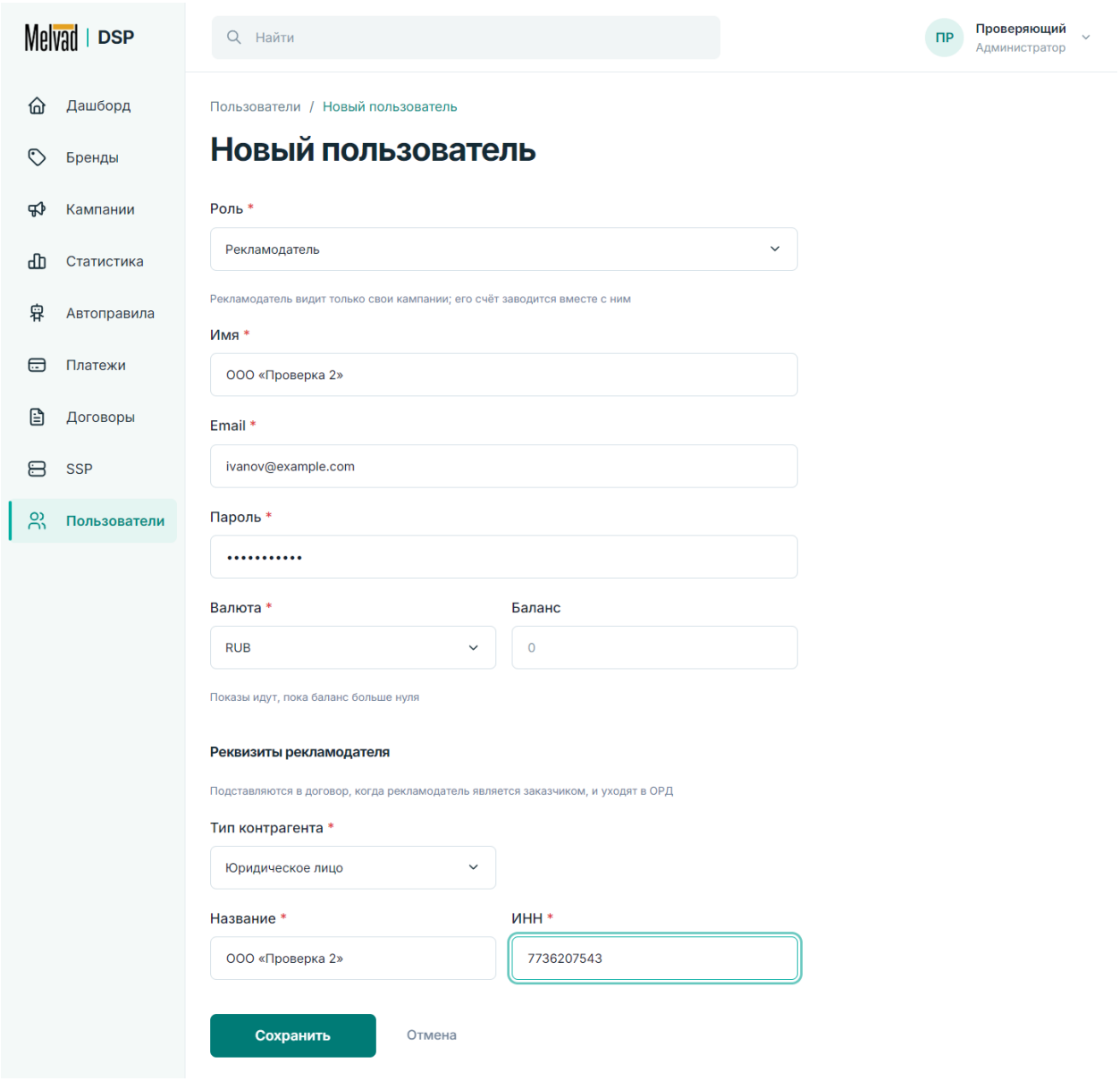


Рисунок 25 — Создание рекламодателя

3.2.2. Заведение договора

Заводим договор. Для этого переходим в меню «Договоры» и жмём кнопку «Создать»:

- Название – Договор на распространение рекламы 2
- Номер договора – П-002
- Дата заключения – сегодняшнее число
- Рекламодатель – ООО «Проверка 2» (выбираем из списка)
- Тип – Договор оказания услуг, ставим галочку «Начальный»
- Подтип – Договор на распространение рекламы
- Первая сторона договора – Заказчик: ставим галочку «Рекламодатель является заказчиком» (реквизиты заказчика подставятся из карточки рекламодателя)
- Вторая сторона договора – Исполнитель: Мелвад (выбрано автоматически)
- Сумма договора – можно не заполнять

Далее нажать кнопку «Сохранить». В списке договоров у созданного договора нажать кнопку «Отправить в ОРД» в колонке «Действия» — «ОРД статус» сменится на «Зарегистрирован». Без регистрации договора креативы кампании не получат erid.

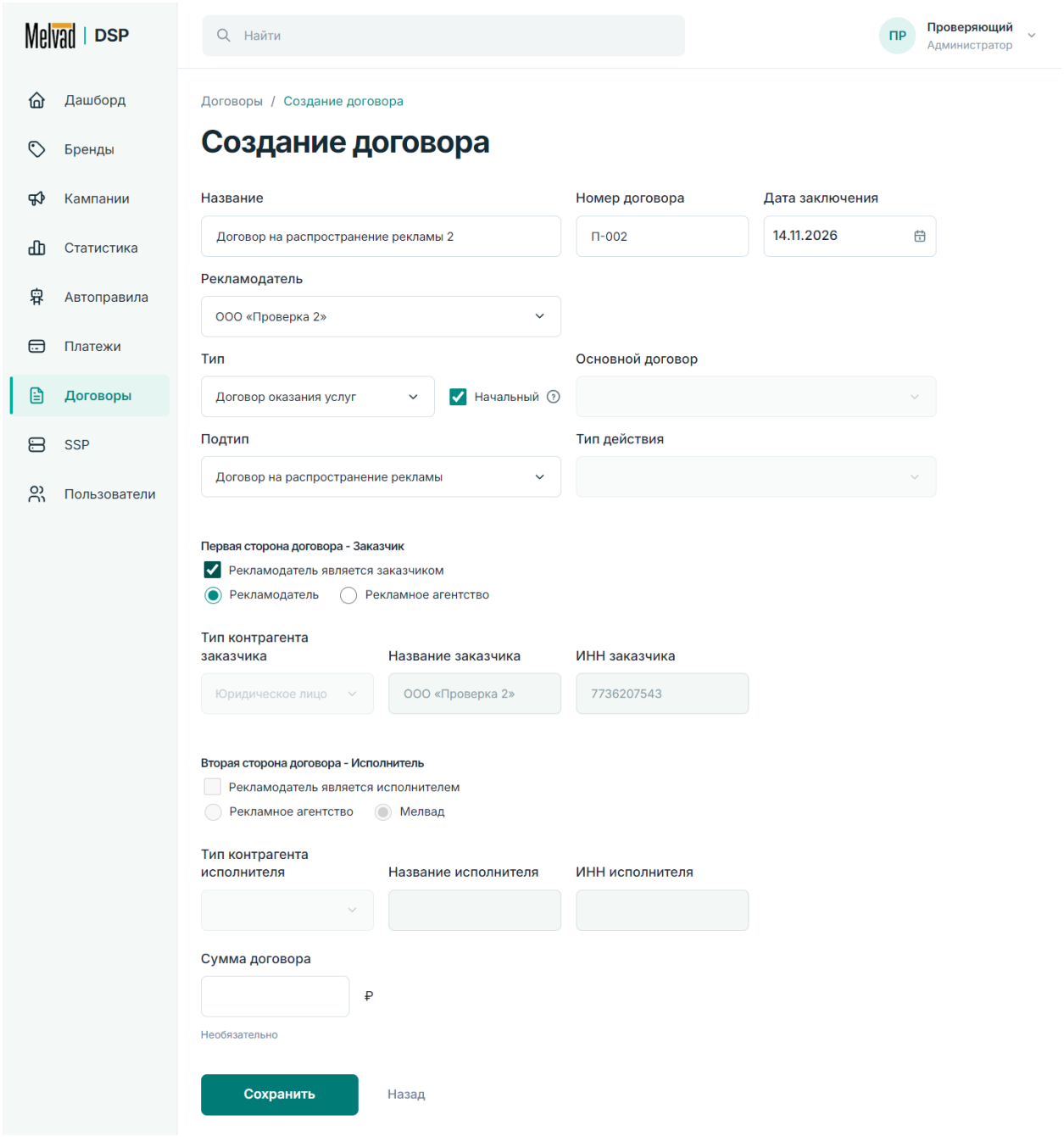


Рисунок 26 — Создание договора

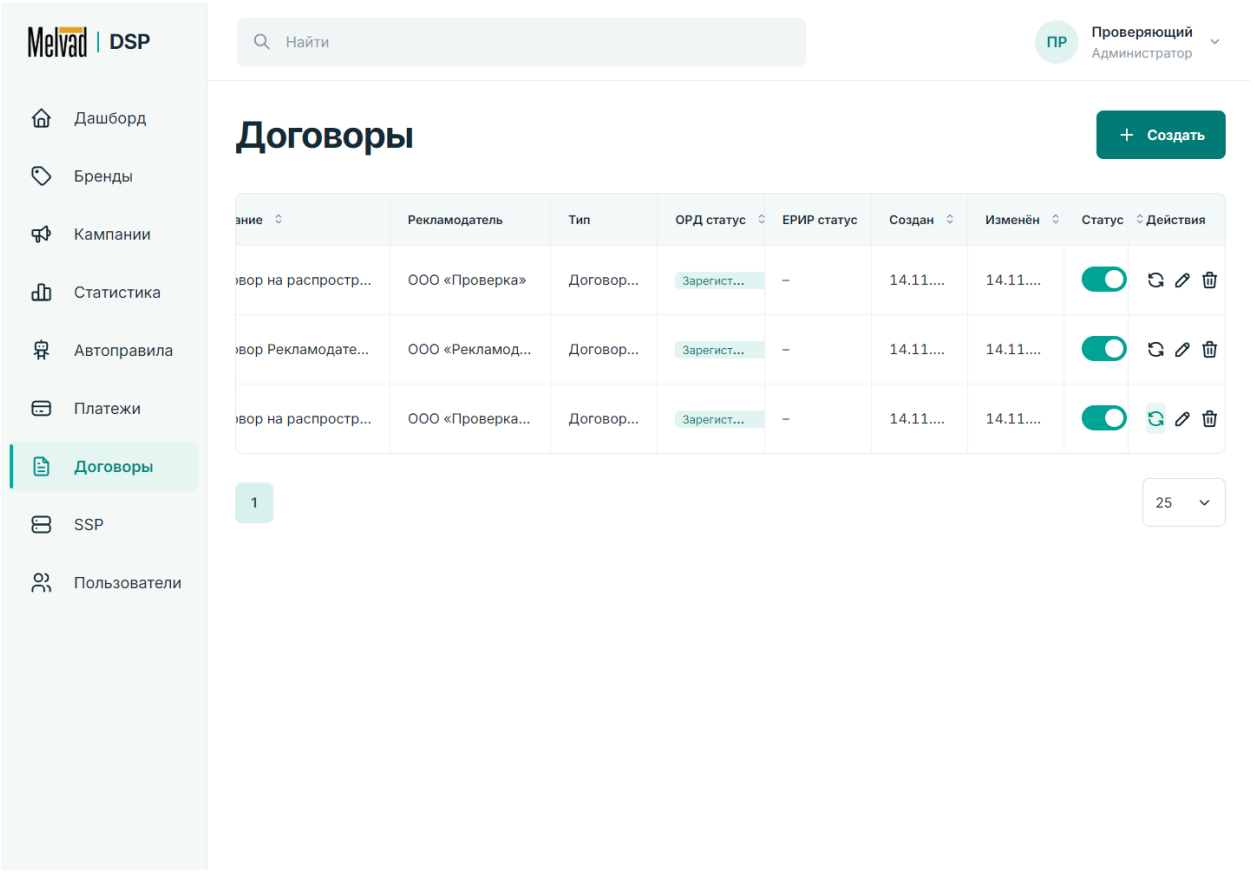


Рисунок 27 — Договор зарегистрирован в ОРД

3.2.3. Баланс рекламодателя

Чтобы рекламная кампания созданного рекламодателя попала в ротацию, ему необходимо пополнить баланс. Для этого нужно перейти в меню «Платежи», нажать кнопку «Изменить баланс», выбрать рекламодателя «ООО «Проверка 2»», операцию «Пополнить», ввести значение 1000 в поле «Сумма» и нажать «Подтвердить». Операция появится в списке «Платежи».

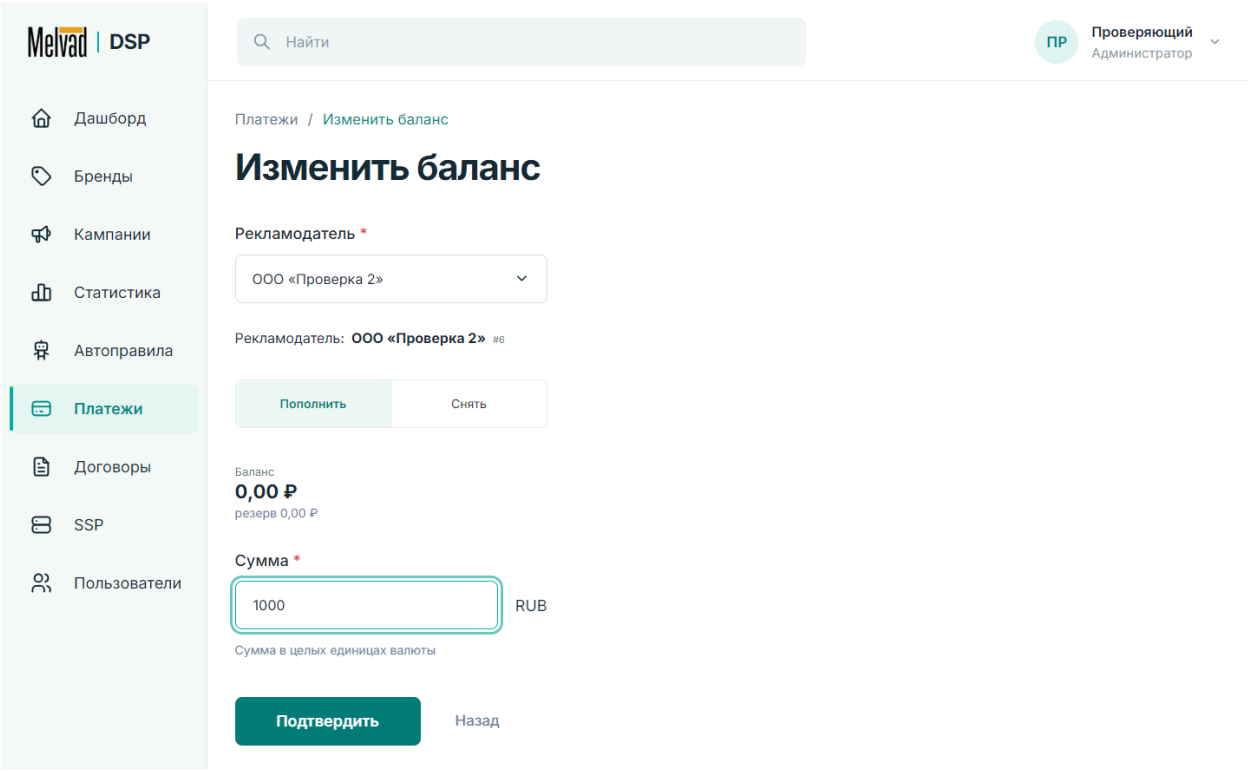


Рисунок 28 — Пополнение баланса рекламодателя

3.2.4. Создание бренда

Для создания бренда необходимо перейти в меню «Бренды», нажать кнопку «Создать» и заполнить поля:

- Рекламодатель – ООО «Проверка 2»
- Название – Проверка 2
- ККТУ – ввести в поиск 5.2.9 и выбрать «5.2.9 · Информационно-развлекательные услуги сотовой связи»

Далее нажать кнопку «Сохранить».

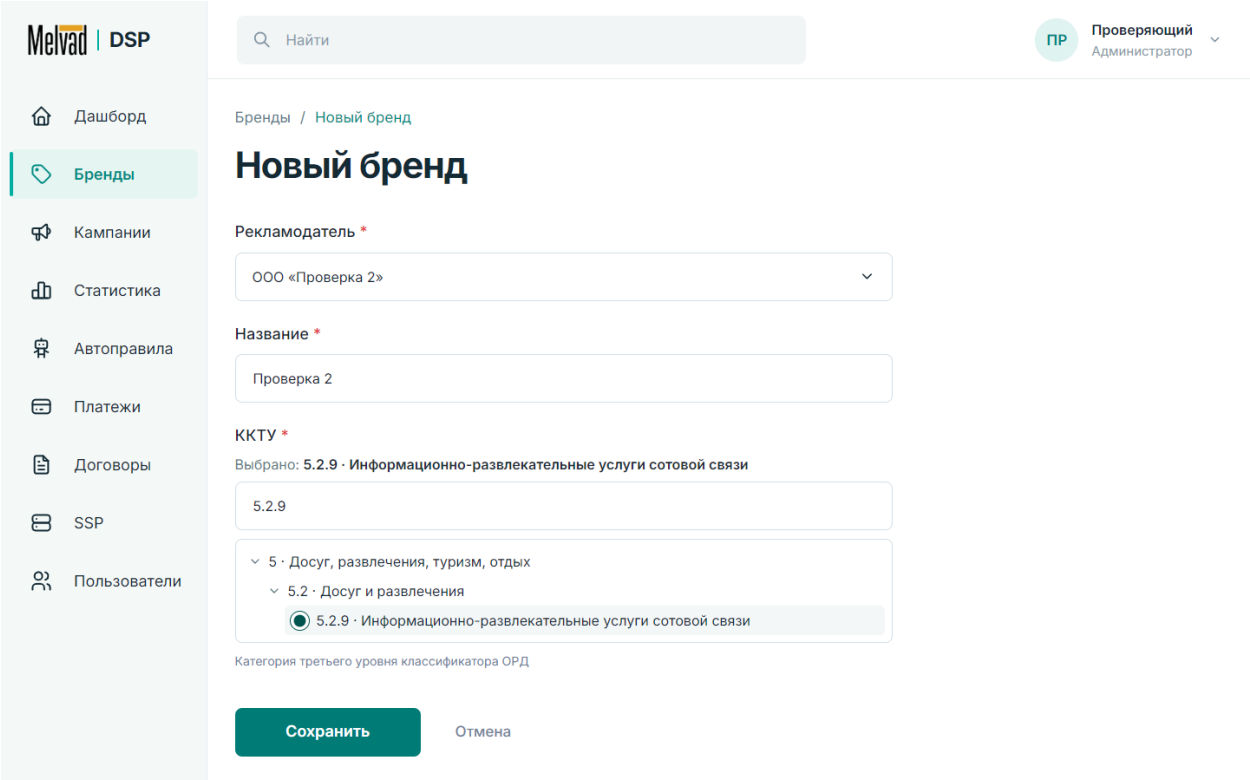


Рисунок 29 — Создание бренда

3.2.5. Создание рекламной кампании, группы объявления и объявления

Для создания рекламной кампании необходимо перейти в меню «Кампании», нажать кнопку «Создать» и заполнить следующие поля:

- Рекламодатель – выбрать из списка ранее созданного рекламодателя «ООО «Проверка 2»»
- Договор – выбрать из списка ранее созданный договор «Договор на распространение рекламы 2 № П-002»
- Название – Проверочная кампания 2
- Бренд – Проверка 2
- Тип – Сайт
- Конечный URL – https://melvad.com
- Дата начала – сегодняшнее число
- Дата окончания – оставить пустой (без даты окончания)

Далее нажать кнопку «Сохранить и продолжить».

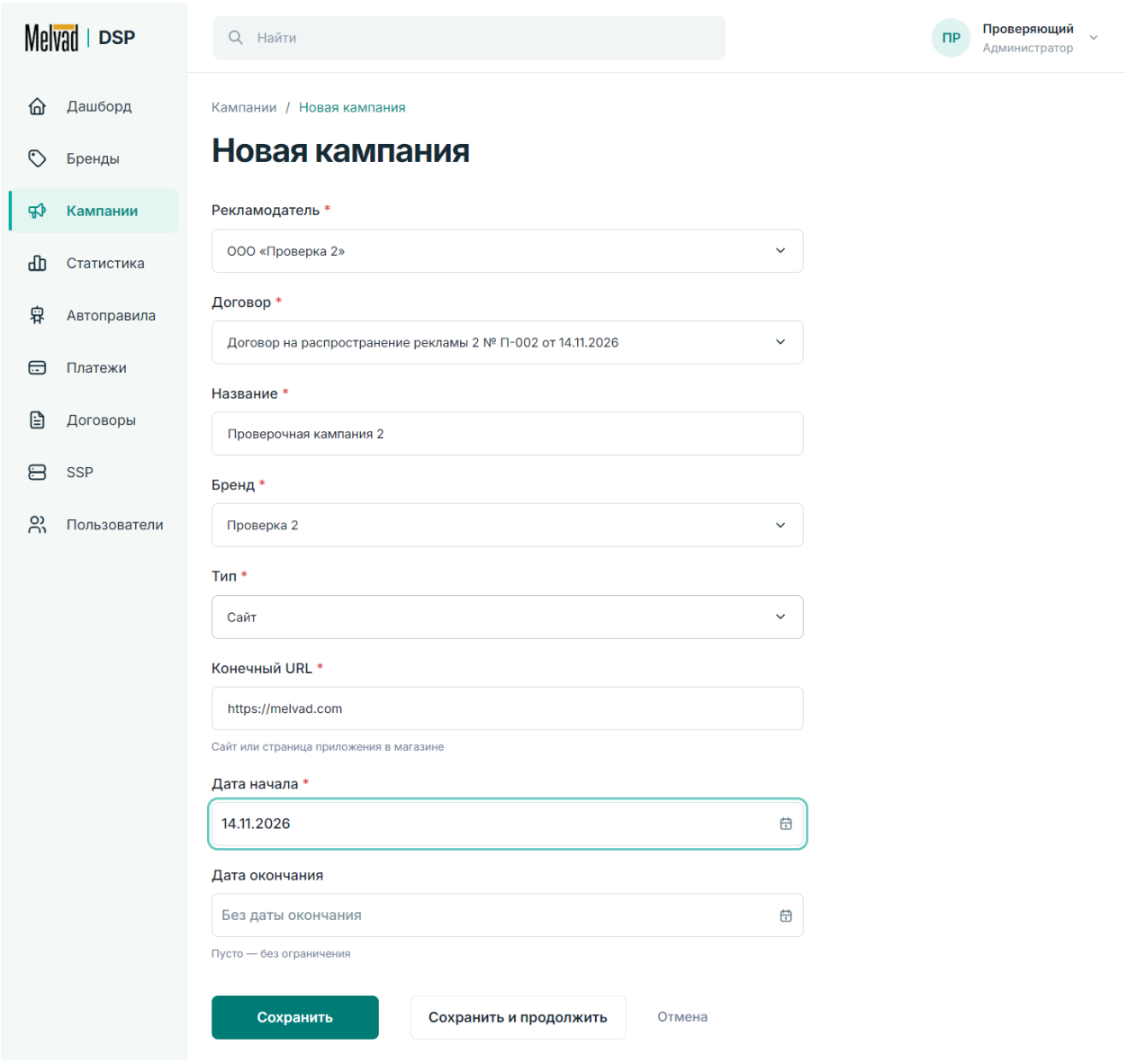


Рисунок 30 — Создание рекламной кампании

Для создания группы объявления необходимо заполнить следующие поля:

- Название – Первая группа
- Ставка – 100
- Дата начала – подставлена дата начала кампании, оставить
- Таргетинги → География – ввести в поиск «Россия» и отметить строку «Россия» (страна)

Остальные поля оставить пустыми. Далее нажать кнопку «Сохранить и продолжить».

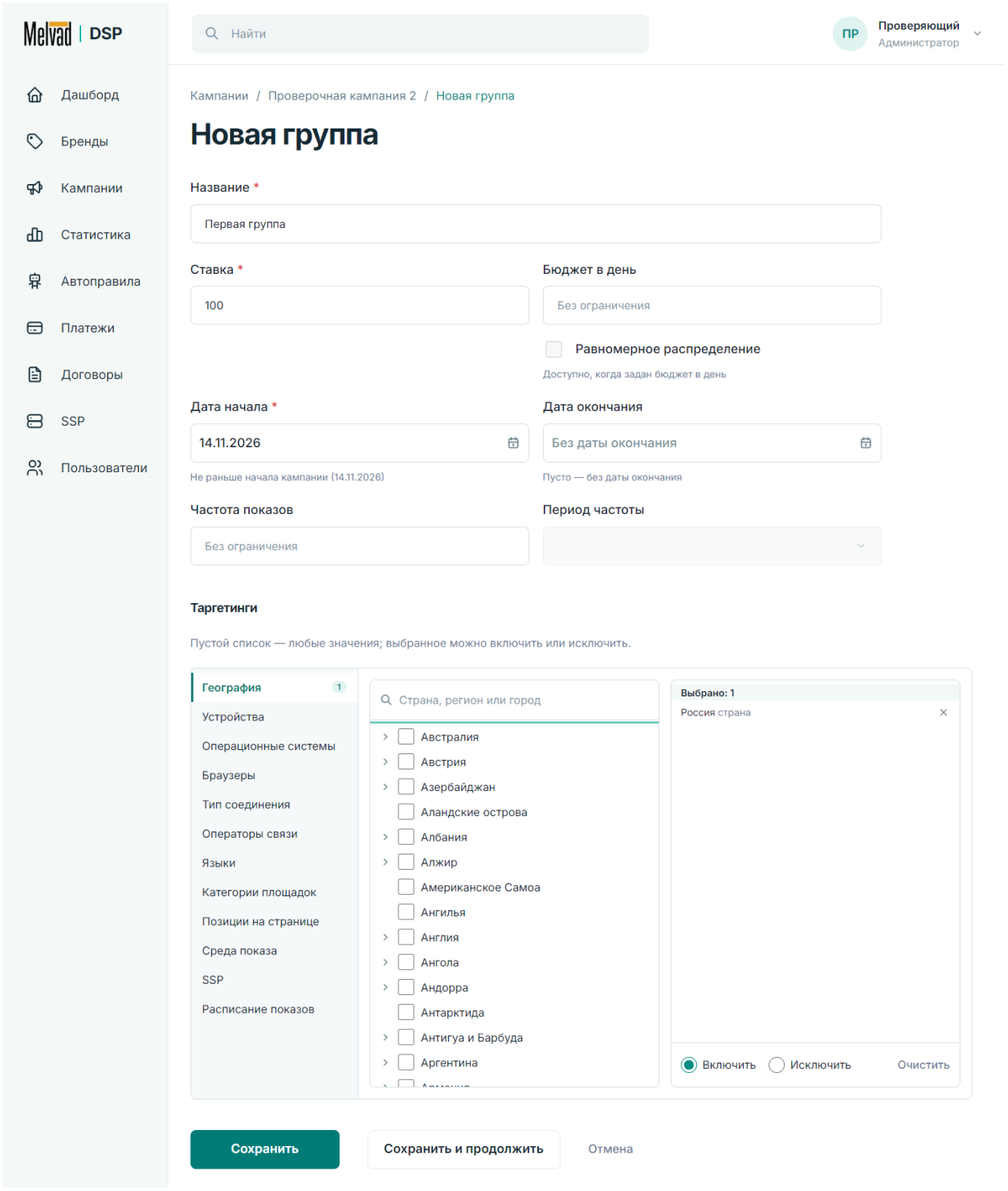


Рисунок 31 — Создание группы объявлений

Для создания объявления нужно выбрать тип «Баннер-изображение», затем заполнить следующие поля:

- Название – Баннер 240×400
- Кликовая ссылка – https://melvad.com
- Изображение – сохранить эту картинку (рисунок ниже, размер 240×400 пикселей) к себе на компьютер и затем нажать кнопку «Обзор» и загрузить ранее сохранённый файл с компьютера



Рисунок 32 — Картинка баннера 240×400 для загрузки

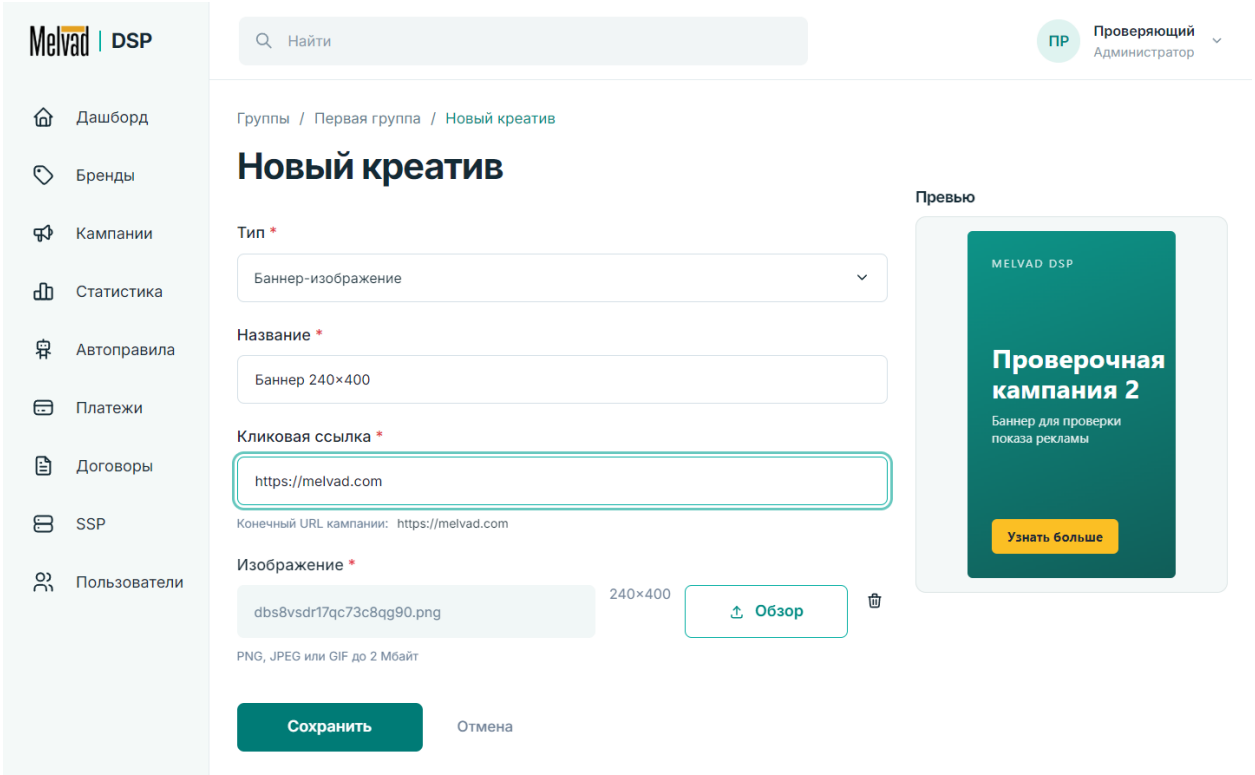


Рисунок 33 — Создание объявления

Далее нажать кнопку «Сохранить». Через несколько секунд креатив будет зарегистрирован в ОРД: в списке креативов в колонке «ОРД статус» появится «Зарегистрирован», а в карточке креатива — erid. Запомните ID креатива (первая колонка списка креативов) и его erid — они понадобятся в п. 3.3.

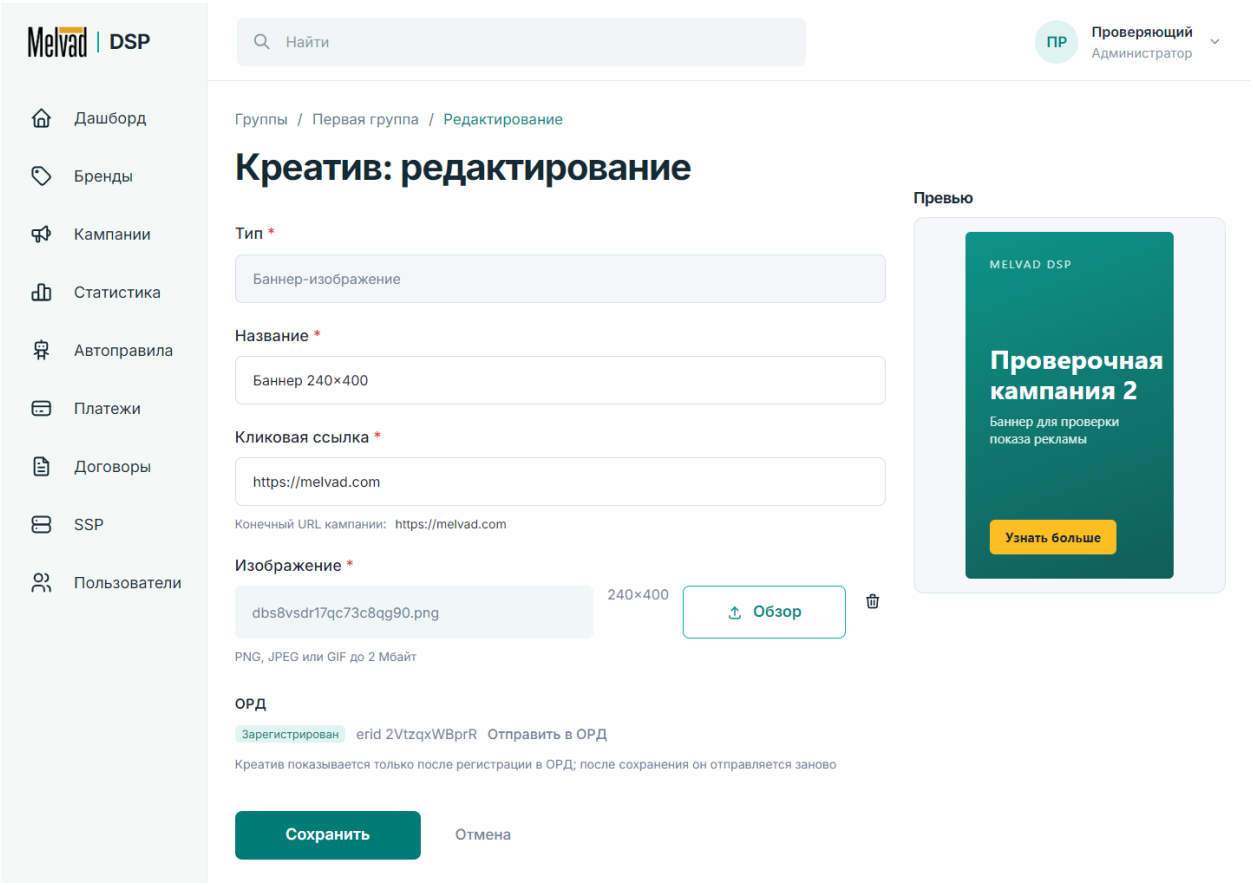


Рисунок 34 — Креатив зарегистрирован в ОРД (erid)

3.2.6. Создание SSP

Для создания SSP необходимо перейти в меню «SSP», нажать кнопку «Новая SSP» и заполнить поля:

- Имя – check-ssp (латиница, цифры, «-» и «_»)
- Домен – check-ssp.ru
- Тип контрагента – Юридическое лицо
- ИНН / налоговый номер – 7743001840
- Наименование – ООО «ВК»

Далее нажать кнопку «Сохранить». После создания нужно в списке SSP найти ранее созданную SSP и нажать иконку «Конечные точки» (значок </>). В окне будет RTB-ссылка вида https://bid.dsp.melvad.com/ssps/1/check-ssp, её нужно скопировать кнопкой «Копировать RTB ссылку» и использовать в Postman на следующем шаге. Регистрация SSP в ОРД (кнопка «Зарегистрировать в ОРД») для получения рекламы не обязательна.

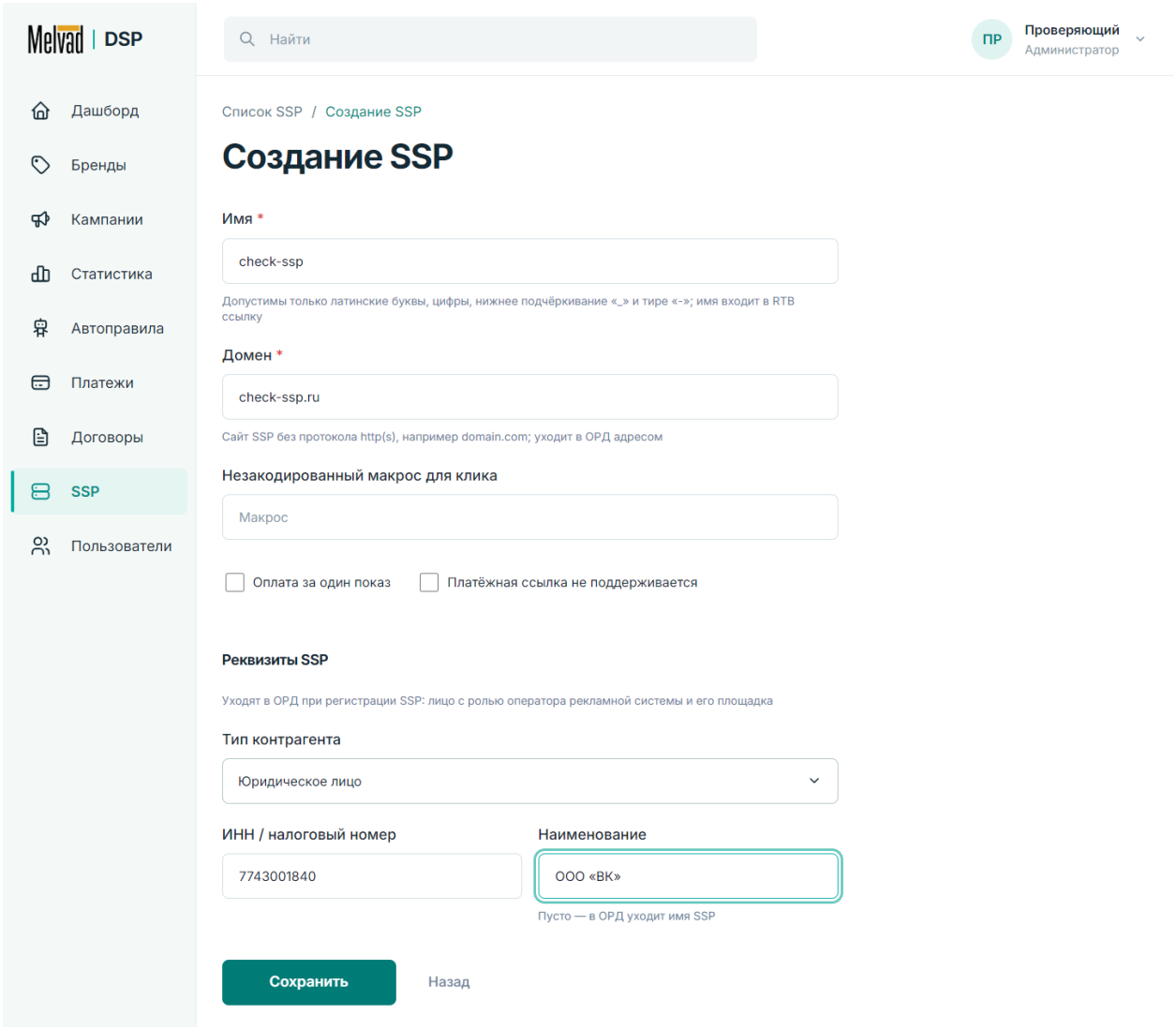


Рисунок 35 — Создание SSP

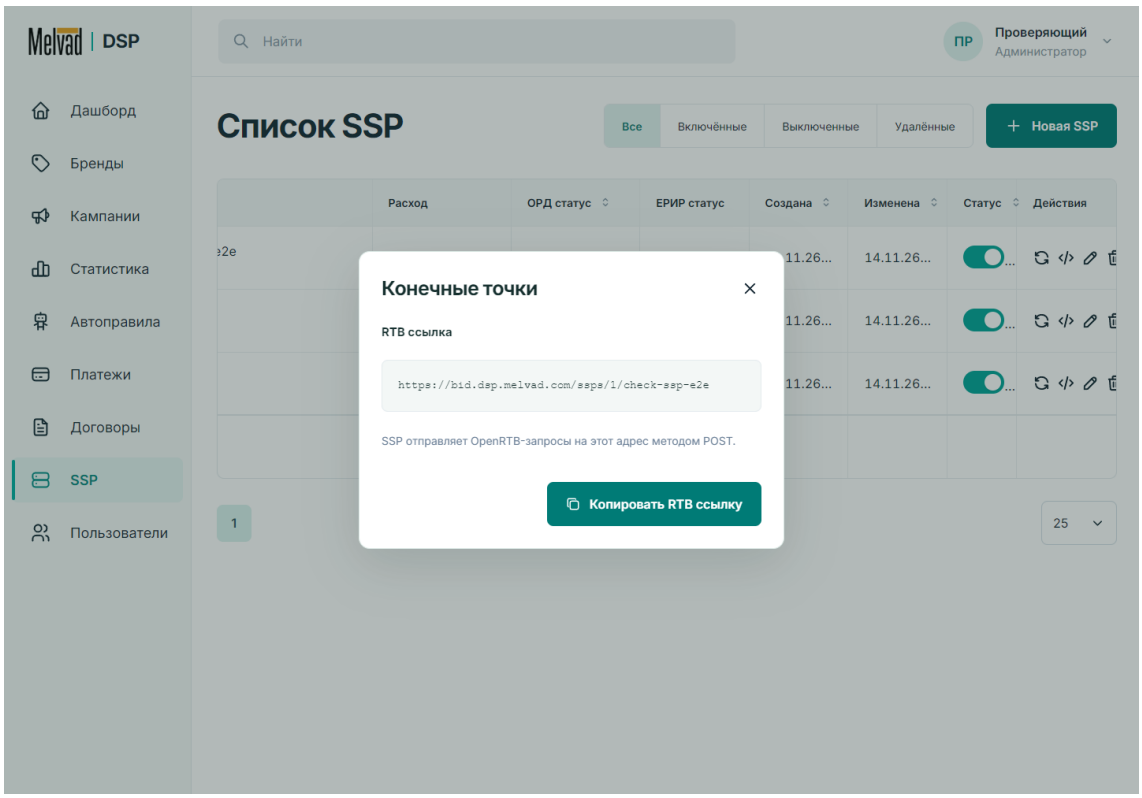


Рисунок 36 — RTB-ссылка созданной SSP (на рисунке — SSP из тестового прогона)

Список SSP можно посмотреть на вкладке «SSP».

3.3. Запрос на получение рекламы

С помощью приложения Postman (скачать можно на официальном сайте — https://www.postman.com) выполняем POST-запрос на RTB-ссылку SSP из п. 3.2.6 (например, https://bid.dsp.melvad.com/ssps/1/check-ssp) с заголовком Content-Type: application/json и телом (Body → raw → JSON):

```
{
  "id": "1234567890",
  "at": 1,
  "tmax": 500,
  "cur": ["RUB"],
  "imp": [
    {
      "id": "1",
      "bidfloor": 10,
      "bidfloorcur": "RUB",
      "secure": 1,
      "banner": {
        "w": 240,
        "h": 400,
        "format": [{ "w": 240, "h": 400 }],
        "mimes": ["image/jpeg", "image/png", "image/gif"]
      }
    }
  ],
  "site": {
    "id": "site-1",
    "domain": "news.example.ru",
    "page": "https://news.example.ru/article/1",
    "cat": ["IAB12"],
    "publisher": { "id": "pub-1" }
  },
  "device": {
    "ip": "77.88.55.80",
```

```
    "ua": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/130.0.0.0 Safari/537.36",
    "devicetype": 2
  },
  "user": { "id": "user-1" },
  "regs": { "ext": { "nroa": { "mandatory": true } } }
}
```


Запрос описывает показ баннера 240×400 на сайте посетителю из России (IP-адрес российского провайдера определяет страну для таргетинга группы); ставка не ниже 10 ₽ за тысячу показов в валюте RUB; regs.ext.nroa.mandatory = true просит вернуть данные маркировки рекламы.

И получаем ответ с рекламой (код ответа 200 OK):

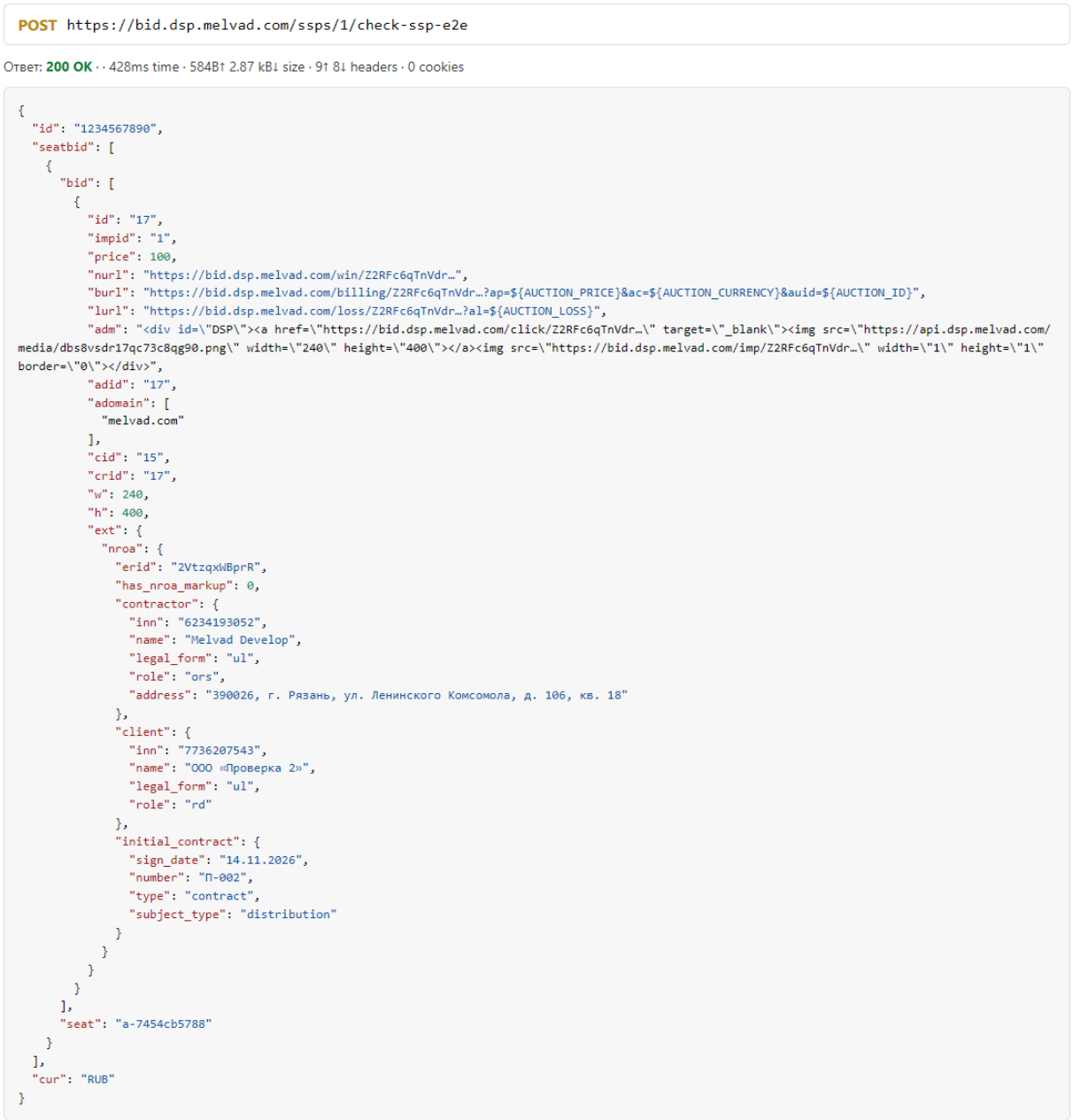


Рисунок 37 — Ответ DSP на запрос в Postman (длинные токены ссылок сокращены)

Проверить, что подобралась именно ранее созданная реклама, можно сравнением идентификаторов в ответе и в интерфейсе Melvad DSP:

- «crid» и «adid» — ID креатива «Баннер 240×400» в списке креативов;
- «cid» — ID кампании «Проверочная кампания 2» в списке кампаний;
- «w» и «h» — 240 и 400, «adm» — HTML-код баннера с картинкой креатива и ссылкой клика;
- «ext.nroa.erid» — erid из блока «ОРД» карточки креатива; «ext.nroa.client» — рекламодатель «ООО «Проверка 2»» с ИНН 7736207543, «initial_contract» — договор № П-002.

Если вместо ответа с рекламой приходит код 204 (No Content), проверьте, что договор и креатив зарегистрированы в ОРД, баланс рекламодателя пополнен, кампания, группа и креатив включены, а дата начала не позже сегодняшнего дня; после изменений подождите около минуты. Ссылки «nurl», «burl» и ссылка показа в «adm» — уведомления о выигрыше и показе, их вызывает SSP; после их вызова показ и расход появляются на дашборде и в статистике.

4. Описание развернутого экземпляра инфраструктуры ПО

4.1. Общее описание

Проверочный стенд (окружение demo) представляет из себя несколько связанных между собой docker-контейнеров на сервере 139.100.213.132 (Ubuntu 24.04, Docker 29). Контейнеры хранилищ и API объединены docker-сетью dsp, биддер, UI, ResCleaner и nginx работают в сети хоста. Ниже представлен полный список контейнеров с их кратким описанием. Схема движения данных представлена в Приложении 1.

Сервисы:

Frontend

- ui-demo (Nginx) — веб-интерфейс на Vue 3, проксирует запросы /api/* к API.

Backend DSP:

- api-demo — API DSP: пользователи, рекламодатели, договоры, бренды, кампании, группы, креативы, SSP, платежи, автоправила, отчёты; конфигурация для биддера; регистрация в ОРД VK.
- bidder-demo — RTB-биддер, обрабатывает запросы SSP на показ рекламы (OpenRTB 2.5), события выигрыша, показа и клика.
- currator-demo — сервис курсов валют для пересчёта ставок, балансов и статистики.
- rescleaner-demo — сервис очистки истёкших резервов бюджетов в KeyDB.

Базы данных:

- dsp-postgres (PostgreSQL 17.5) — основная БД (база dsp) и БД сервиса курсов (база currator).
- dsp-clickhouse (ClickHouse 26.3) — аналитическая БД для статистики.
- dsp-keydb (KeyDB 6.0.18) — балансы рекламодателей, бюджеты и счётчики биддера, сессии UI.

Брокеры сообщений:

- dsp-kafka (Apache Kafka 4.3.2, режим KRaft без Zookeeper) — очередь сообщений для статистики (топик stats).

Публикация и развёртывание:

- nginx — принимает HTTP/HTTPS на портах 80 и 443 и проксирует домены DSP к контейнерам; сертификаты Let's Encrypt выпускает и обновляет lego (запускается по расписанию cron).
- gitlab-runner-dsp — раннер GitLab CI/CD, разворачивает новые версии сервисов на сервере.

Адреса:

Адрес	Назначение
https://ui.dsp.melvad.com	веб-интерфейс
https://api.dsp.melvad.com	API, файлы креативов (/media/)
https://bid.dsp.melvad.com	RTB-биддер: /ssps/1/<имя SSP>, события /win, /billing, /imp, /click

Порты, на которых запущены модули (снаружи сервера доступны только 80 и 443 через nginx и порт биддера 8081; остальные порты открыты только на 127.0.0.1):

- UI (ui-demo): 18083

- API (api-demo): 18082 (порт 8080 внутри контейнера)
- Bidder (bidder-demo): 8081 (RTB), 5060 (debug), 5070 (метрики)
- Currator (currator-demo): 18084 (API), 5370 (метрики)
- ResCleaner (rescleaner-demo): 5380 (метрики)
- PostgreSQL (dsp-postgres): 5432
- KeyDB (dsp-keydb): 6379
- ClickHouse (dsp-clickhouse): 8123 (HTTP), 9000 (native)
- Kafka (dsp-kafka): 9093 (для биддера), 9092 (внутри сети dsp)
- nginx: 80, 443

4.2. Подробное описание контейнеров сервисов, их процессов и компонентов

4.2.1. UI

Контейнер: ui-demo

Образ: gitlab.melvad.com:5050/dsp/ui/master

Описание

Веб-интерфейс на базе Vue 3, раздаваемый через Nginx (сеть хоста, 127.0.0.1:18083). Проксирует API-запросы к API того же сервера.

Процессы

- nginx (master) — главный процесс Nginx, управляет worker процессами.
- nginx (worker) — рабочие процессы, обрабатывают HTTP запросы.

Компоненты

- Nginx — веб-сервер.
- Vue 3 приложение (SPA), графики Chart.js, иконки Tabler.

Взаимодействие

- Принимает HTTP запросы от общего nginx сервера (домен ui.dsp.melvad.com).
- Статические файлы (JS, CSS, изображения) отдаёт напрямую.
- API запросы (/api/*) проксирует на api-demo (127.0.0.1:18082).

4.2.2. API

Контейнер: api-demo

Образ: gitlab.melvad.com:5050/dsp/api/master

Описание

REST API на Go (Gin, go-pg) для управления пользователями, рекламодателями, договорами, брендами, кампаниями, группами, креативами, SSP, балансами, автоправилами и шаблонами отчётов через веб-интерфейс; отдаёт биддеру конфигурацию кампаний и регистрирует рекламу в ОРД VK.

Процессы

- api — обработка HTTP запросов и бизнес-логика.
- Фоновые задачи: движок автоправил (раз в минуту), регистрация рекламодателей и креативов в ОРД и повтор ошибок (раз в 10 минут), опрос статусов ЕРИР, опрос курсов Currator (раз в 5 минут).

Компоненты

- HTTP сервер на порту 8080 в контейнере (127.0.0.1:18082 на сервере).
- Маршруты UI (/auth, /users, /contracts, /brands, /campaigns, /groups, /creatives, /ssps, /autorules, /balances, /reports, /stats, /media) и маршруты биддера (/v1/adconfig, /v1/adconfig/dictionary, /v1/convconfig, /v1/rates) с ключом администратора.
- Миграции PostgreSQL (golang-migrate) и ClickHouse, выполняются командами api migrate и api chmigrate при развёртывании.
- Клиент ОРД VK, выгрузка отчётов в CSV и XLSX.
- Утилита create-user — создание администратора.

Взаимодействие

- PostgreSQL — основная БД (пользователи, рекламодатели, договоры, бренды, кампании, группы, креативы, SSP, автоправила, журнал баланса, шаблоны отчётов); после изменений — уведомление NOTIFY для биддера.
- KeyDB — балансы рекламодателей (общие с биддером), сессии UI.
- ClickHouse — чтение статистики для дашборда, отчётов, автоправил и списка SSP.
- Currator — курсы валют.
- ОРД VK — регистрация рекламодателей, договоров, креативов и SSP, статусы ЕРИР.
- Каталог /opt/dsp/media — загруженные файлы креативов (отдаёт nginx по адресу api.dsp.melvad.com/media/).

4.2.3. Bidder

Контейнер: bidder-demo

Образ: gitlab.melvad.com:5050/dsp/bidder/master

Описание

RTB (Real-Time Bidding) биддер на Go — обрабатывает запросы SSP на показ рекламы в реальном времени, подбирает креативы по таргетингам групп, назначает ставку, формирует разметку объявления с данными маркировки (erid) и учитывает события показа и клика.

Процессы

- bidder — обработка RTB запросов и событий (выигрыш, оплата, проигрыш, показ, клик).
- Загрузка конфигурации кампаний из API и подписка на её изменения (LISTEN в PostgreSQL).
- Отправка статистики в ClickHouse через Kafka.

Компоненты

- HTTP сервер на порту 8081 (fasthttp) — /ssps/<клиент>/<SSP>, /win, /billing, /loss, /imp, /click.
- Debug сервер на порту 5060 и сервер метрик на порту 5070 (только 127.0.0.1).
- Клиент KeyDB — балансы, бюджеты, резервы, ограничения частоты.
- Kafka producer — отправка статистики (топик stats).
- Базы MaxMind GeoIP — определение страны, региона и города по IP.

Взаимодействие

- API (127.0.0.1:18082) — конфигурация кампаний, групп, креативов, SSP и справочники.

- PostgreSQL (база dsp) — уведомления об изменении конфигурации и курсов (eventbus).
- KeyDB — бюджеты, балансы и резервы.
- Kafka — запись событий в топик stats.
- Currator — курсы валют.

4.2.4. ResCleaner

Контейнер: rescleaner-demo

Образ: gitlab.melvad.com:5050/dsp/rescleaner/master

Описание

Сервис для очистки устаревших данных в KeyDB: снимает истёкшие резервы бюджетов, которые биддер ставит на время аукциона, и возвращает их в доступный бюджет.

Процессы

- rescleaner — периодический обход резервов и очистка истёкших.

Компоненты

- Клиент KeyDB (база 1 биддера).
- Сервер метрик на порту 5380 (127.0.0.1).

Взаимодействие

- KeyDB — очистка устаревших резервов.

4.2.5. Currator

Контейнер: currator-demo

Образ: gitlab.melvad.com:5050/dsp/currator/master

Описание

Сервис курсов валют: раз в час получает курсы из openexchangerates.org (резервный источник — курсы ЦБ РФ), хранит их и отдаёт API и биддеру.

Процессы

- currator — обновление и выдача курсов.

Компоненты

- HTTP API на порту 18084, метрики на порту 5370 (127.0.0.1).

Взаимодействие

- PostgreSQL (база currator) — хранение курсов; уведомление NOTIFY rates в базе dsp.
- openexchangerates.org, ЦБ РФ — источники курсов.
- API и Bidder — получатели курсов.

4.2.6. PostgreSQL

Контейнер: dsp-postgres

Образ: postgres:17.5

Описание

Реляционная БД PostgreSQL 17.5: база dsp (все сущности DSP, справочники таргетингов, журнал баланса, шаблоны отчётов) и база currator (курсы валют).

Процессы

- postgres — главный процесс и служебные процессы (checkpointer, background writer, walwriter, autovacuum).

Компоненты

- Порт 5432 на 127.0.0.1.

Взаимодействие

- api-demo — чтение и запись, NOTIFY при изменениях.
- bidder-demo — уведомления LISTEN.
- currator-demo — курсы валют.

4.2.7. KeyDB

Контейнер: dsp-keydb

Образ: eqalpha/keydb:alpine_x86_64_v6.0.18

Описание

In-memory хранилище KeyDB (совместимо с Redis) с сохранением на диск (appendonly): балансы рекламодателей, бюджеты, резервы и счётчики биддера, сессии UI.

Процессы

- keydb-server — сервер хранилища.

Компоненты

- Порт 6379 на 127.0.0.1, ограничение памяти 4 ГБ.

Взаимодействие

- api-demo — балансы и сессии.
- bidder-demo — бюджеты, резервы, частота показов.
- rescleaner-demo — очистка резервов.

4.2.8. ClickHouse

Контейнер: dsp-clickhouse

Образ: clickhouse/clickhouse-server:26.3.62.3

Описание

Аналитическая БД ClickHouse 26.3, база dsp: таблица stats_queue (движок Kafka) читает топик stats, материализованное представление stats_view пишет события в таблицу stats.

Процессы

- clickhouse-server — сервер БД.

Компоненты

- HTTP на порту 8123 и native на порту 9000 (127.0.0.1).

Взаимодействие

- dsp-kafka — чтение топика stats.
- api-demo — запросы статистики.

4.2.9. Kafka

Контейнер: dsp-kafka

Образ: apache/kafka:4.3.2

Описание

Брокер сообщений Apache Kafka 4.3.2 в режиме KRaft (без Zookeeper), один узел: очередь событий биддера (топик stats).

Процессы

- java (kafka.Kafka) — брокер и контроллер KRaft.

Компоненты

- Слушатель 9093 на 127.0.0.1 — для биддера в сети хоста.
- Слушатель 9092 — внутри docker-сети dsp для ClickHouse.

Взаимодействие

- bidder-demo — запись событий.
- dsp-clickhouse — чтение событий.

4.2.10. nginx и lego

Контейнер: nginx

Образ: nginx:1.30.4-alpine

Описание

Общий веб-сервер стенда: принимает HTTP и HTTPS запросы, перенаправляет HTTP на HTTPS и проксирует домены ui.dsp.melvad.com, api.dsp.melvad.com и bid.dsp.melvad.com к контейнерам; отдаёт файлы креативов. Сертификат Let's Encrypt выпускает утилита lego (контейнер goacme/lego запускается скриптом /opt/lego_renew.sh по cron ежедневно в 03:30 UTC).

Процессы

- nginx (master) и nginx (worker).

Компоненты

- Порты 80 и 443 (сеть хоста).
- Конфигурация /home/nginx/config, сертификаты /home/nginx/certs.

Взаимодействие

- ui-demo, api-demo, bidder-demo — проксирование запросов.
- Let's Encrypt — проверка домена (HTTP-01) и выпуск сертификата.

4.3. Настройка удаленного доступа к развернутому экземпляру ПО

Для проверки создан отдельный пользователь сервера **dsp-review**. Он не имеет прав администратора и прямого доступа к Docker: состояние, логи и данные Melvad DSP он смотрит командой `sudo dsp`, которая работает только с контейнерами DSP и только на чтение. Подключение можно осуществить двумя способами.

- Через пароль.

IP-адрес – 139.100.213.132
Логин – dsp-review
Пароль – xvJ60iXylb8BS546JHMg
Пример подключения: `ssh dsp-review@139.100.213.132`

- Через ssh-ключ.

Для подключения по ключу необходимо скачать файл с именем `id_ed25519_dsp_review`, представленный по ссылке. Файл может быть перемещен в рабочий каталог (или в текущий рабочий каталог вашего терминала).

Затем необходимо выставить ему минимальные права:

```
chmod 0400 id_ed25519_dsp_review
```

После этого можно подключаться (ключ -i позволяет задать путь и имя файла с ключом):

```
ssh -i id_ed25519_dsp_review dsp-review@139.100.213.132
```

Команды на сервере (пароль для sudo не нужен):

Команда	Что показывает
`sudo dsp ps`	контейнеры DSP: состояние, число перезапусков, образ
`sudo dsp stats`	CPU, память, сеть и диск контейнеров DSP
`sudo dsp inspect <контейнер>`	образ, состояние, сеть, порты, тома и ресурсы контейнера
`sudo dsp top <контейнер>`	процессы контейнера
`sudo dsp logs <контейнер> [N]`	последние N строк лога (по умолчанию 100)
`sudo dsp volumes`	каталоги данных DSP и их размер
`sudo dsp health`	проверки доступности сервисов
`sudo dsp sql "<запрос>"`	запрос к PostgreSQL DSP (база dsp), только чтение
`sudo dsp ch "<запрос>"`	запрос SELECT, SHOW или DESCRIBE к ClickHouse DSP (база dsp)

Примеры:

```
sudo dsp ps
sudo dsp logs api-demo 50
sudo dsp sql "select id, name, status, start_date, end_date from campaigns order by id desc
limit 10"
sudo dsp ch "show tables"
sudo dsp ch "select toDate(dateTime) d, sum(impressions), sum(clicks) from stats group by d
order by d desc limit 7"
```

4.4. Расположение файлов приложения

Приложение работает в Docker-контейнерах (команда docker ps на сервере покажет список, описанный в пункте 4.1), данные хранятся в смонтированных каталогах хоста (/opt/dsp/data) и в Docker volumes. Хранилища описаны в /opt/dsp/docker-compose.yml, биддер — в /opt/dsp/bidder/docker-compose.yml; UI, API, Currator и ResCleaner разворачиваются из GitLab CI/CD.

4.4.1. Список volumes

Volume	Контейнер	Назначение	Путь внутри контейнера
bidder-demo_bidder-cache	bidder-demo	Кэш биддера	/usr/local/bin/cache
анонимный том образа (8aa78628…)	dsp-kafka	Каталог секретов образа Kafka (не используется)	/etc/kafka/secrets

Volume	Контейнер	Назначение	Путь внутри контейнера
анонимный том образа (9fdfd7fa…)	dsp-kafka	Общий каталог конфигурации образа Kafka	/mnt/shared/config
анонимный том (b6d04207…)	gitlab-runner-dsp	Домашний каталог раннера	/home/gitlab-runner
runner-…-cache-…	задания CI/CD	Кэш заданий сборки и развёртывания	/cache

4.4.2. Смонтированные каталоги хоста

Путь на хосте	Контейнер	Путь в контейнере	Назначение
/opt/dsp/data/postgres	dsp-postgres	/var/lib/postgresql/data	Данные PostgreSQL
/opt/dsp/data/keydb	dsp-keydb	/data	Данные KeyDB (AOF, RDB)
/opt/dsp/data/clickhouse	dsp-clickhouse	/var/lib/clickhouse	Данные ClickHouse
/opt/dsp/clickhouse/config.d	dsp-clickhouse	/etc/clickhouse-server/config.d	Конфигурация ClickHouse (UTC, память)
/opt/dsp/data/kafka	dsp-kafka	/var/lib/kafka/data	Данные Kafka
/opt/dsp/api/settings.yaml	api-demo	/etc/dsp-api/settings.yaml	Настройки API (только чтение)
/opt/dsp/media	api-demo, nginx	/var/lib/dsp-api/media, /opt/dsp/media	Файлы креативов
/opt/dsp/bidder/bidder.settings.json	bidder-demo	/usr/local/bin/bidder.settings.json	Настройки биддера (только чтение)
/home/maxmind	bidder-demo	/usr/local/bin/assets/maxmind	Базы MaxMind GeoIP
/opt/dsp/currator/currator.settings.json	currator-demo	/opt/currator/bin/currator.settings.json	Настройки Currator
/opt/dsp/rescleaner/rescleaner.settings.json	rescleaner-demo	/opt/rescleaner/bin/rescleaner.settings.json	Настройки ResCleaner
/home/nginx/config/nginx.conf, conf.d, dhparams.pem	nginx	/etc/nginx/nginx.conf, /etc/nginx/conf.d, /etc/nginx/dhparams.pem	Конфигурация nginx (домены DSP — conf.d/dsp.conf)
/home/nginx/certs	nginx	/etc/ssl/certs	Сертификаты Let's Encrypt (lego)
/home/nginx/html	nginx	/usr/share/nginx/html	Служебные страницы
/home/gitlab-runner-dsp/config	gitlab-runner-dsp	/etc/gitlab-runner	Конфигурация раннера
/var/run/docker.sock	gitlab-runner-dsp	/var/run/docker.sock	Управление Docker при развёртывании

Контейнер ui-demo не использует каталоги хоста: статика встроена в образ. Пароли и ключи сервисов хранятся только в файлах настроек на сервере (/opt/dsp/.env и файлы settings), доступных пользователю root.

Приложение 1. Общая схема контейнеров и движения данных между ними

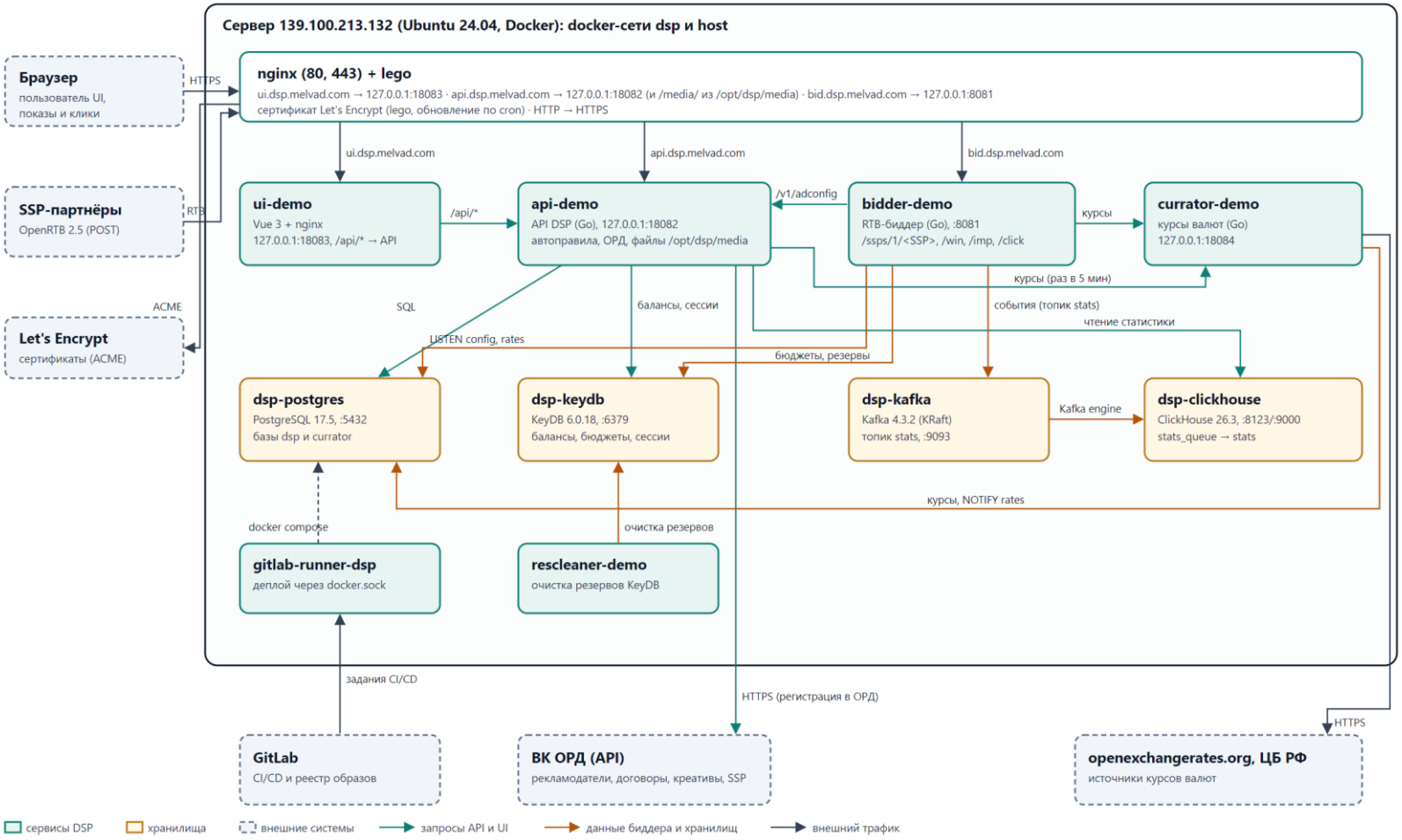


Рисунок 38 — Схема контейнеров Melvad DSP на сервере