

Отчёт о выполненных задачах (18.12.2024)

Задача #2: Поле "Сдаёт авто с..."

Файл: `app/serializers.py`

Изменение: Добавлено поле `renting_since` в `LessorDetailSerializer`

Описание: Возвращает дату первого добавленного транспорта арендодателя. Если транспорта нет — `null`.

Ответ API:

```
{
  "lessor": {
    "renting_since": "2021-11-18"
  }
}
```

Коммит: `Feat: add renting_since field to LessorDetailSerializer`

Задача #3: Запрет отмены поездки после оплаты

Файл: `chat/views.py`

Изменение: В endpoint `cancel-by-client` добавлена проверка на успешную оплату

Описание: После успешной оплаты клиент не может отменить поездку самостоятельно. Возвращается ошибка 403 с сообщением "Поездка уже оплачена. Для отмены обратитесь в техподдержку."

Ответ API при попытке отмены оплаченной поездки:

```
{
  "detail": "Поездка уже оплачена. Для отмены обратитесь в техподдержку."
}
```

Коммит: `Feat: block trip cancellation after payment - redirect to support`

Задача #4: Ошибка при смене типа аренды (price_deposit)

Статус: Исправлено

Ошибка:

```
{'price_deposit': [ErrorDetail(string='Убедитесь, что вы ввели не более 10
цифры.', code='max_digits')]}
```

Решение: Увеличен лимит `max_digits` с 10 до 15.

Было:

```
price_deposit = DecimalField(decimal_places=2, max_digits=10) # макс: 99999999.99
```

Стало:

```
price_deposit = DecimalField(decimal_places=2, max_digits=15) # макс:
99999999999999.99
```

Файл: `vehicle/models.py`

Миграция: `0033_increase_price_deposit_max_digits.py`

Коммит: `Feat: increase price_deposit max_digits from 10 to 15`

Задача #6: Уведомление при отмене заявки арендодателем

Статус: Уже реализовано на бэкенде

Текущая логика (`chat/views.py`, метод `perform_update`):

При установке `status='denied'` арендодателем:

1. Проверяется, что это владелец транспорта
2. Требуется указать причину отказа (`denied_reason`)
3. Создаётся уведомление арендатору: "Вам отказано в аренде транспорта {vehicle} по причине: {reason}"
4. Заявка сохраняется со статусом `denied`

API запрос:

```
PATCH /chat/request_rents/{id}/
{
  "status": "denied",
  "denied_reason": "Транспорт недоступен в указанные даты"
}
```

Доработки бэкенда: Не требуются

Задача #7: Цены на машины меняются при обновлении

Статус: Исправлено

Проблема: При обновлении транспорта использовался `bulk_create()` для создания `RentPrice`, который **не вызывает метод `save()`** модели. Из-за этого поле `total` не пересчитывалось корректно.

Было:

```
RentPrice.objects.bulk_create([
    RentPrice(vehicle=instance, name=p['name'], price=..., discount=...)
    for p in prices_data
])
```

Стало:

```
for p in prices_data:
    RentPrice.objects.create(
        vehicle=instance, name=p['name'], price=..., discount=...
    )
```

Файл: `vehicle/serializers/base.py`

Коммит: `Fix: use create() instead of bulk_create() for RentPrice to calculate total`

Задача #11: Сортировка транспорта после верификации

Статус: Выполнено

Требование: Новый транспорт должен быть вверху неverified, после одобрения — вверху verified.

Изменения:

1. **Добавлено поле `verified_at`** в модель `Vehicle` (`vehicle/models.py`)
 - Заполняется при верификации админом/менеджером
 - Сбрасывается при снятии верификации
2. **Обновлена логика верификации** (`vehicle/serializers/base.py`)
 - При `verified=True` устанавливается `verified_at=datetime.now()`
 - При сбросе верификации `verified_at=None`
3. **Обновлена сортировка** для всех ViewSet (`Auto`, `Bike`, `Ship`, `Helicopter`, `SpecialTechnic`):
 - Неverified: сортировка по `-created_at` (новые первыми)
 - Verified: сортировка по `-verified_at` (недавно одобренные первыми)

Файлы: `vehicle/models.py`, `vehicle/views.py`, `vehicle/serializers/base.py`

Миграция: `0032_vehicle_verified_at.py`

Коммит: `Feat: add verified_at field and sort verified vehicles by verification date`

Проверка: Бэкенд возвращает данные в правильном порядке:

```
SELECT id, created_at FROM vehicle_vehicle WHERE verified = false ORDER BY
created_at DESC LIMIT 5;
-- Результат: 153, 151, 150, 149, 145 (новые первыми)
```

Если на фронте порядок другой — проблема на стороне фронтенда (возможно, список переворачивается при отображении или передаётся параметр `ordering`).

Задача #12: Комиссия на стоимость доставки

Статус: Исправлено (⚠ требует уточнения)

Проблема: Комиссия рассчитывалась от `total_cost`, который включает стоимость доставки. При комиссии 20% и доставке 750₽ — лишние 150₽.

Было:

```
def calculate_amount(self):
    commission = self.vehicle.owner.lessor.commission
    return Decimal(self.total_cost) * commission / Decimal(100)
```

Стало:

```
def calculate_amount(self):
    commission = self.vehicle.owner.lessor.commission
    rent_cost = Decimal(self.total_cost) - Decimal(self.delivery_cost or 0)
    return rent_cost * commission / Decimal(100)
```

Файл: `chat/models.py`

Коммит: `Fix: exclude delivery cost from commission calculation`

⚠ **ТРЕБУЕТ УТОЧНЕНИЯ:**

Текущая логика:

- `Payment.amount` = комиссия (отправляется в Tinkoff)
- `Payment.delivery` = стоимость доставки (хранится отдельно, НЕ отправляется в Tinkoff)

Вопросы:

1. Доставка должна оплачиваться через Tinkoff вместе с комиссией?
2. Или доставка оплачивается отдельно (наличными арендодателю)?

Если доставка должна оплачиваться через платформу — нужно изменить `payment/views.py`:

```
total_to_pay = payment.amount + payment.delivery
```
