

OzgeContract API Documentation

Обзор

OzgeContract - это backend API для системы управления контрактами с цифровыми подписями. API предоставляет функционал для управления компаниями, шаблонами контрактов, подписями и платежными запросами.

Base URL: `http://localhost:4000`

Аутентификация

На данный момент API не требует аутентификации. В будущих версиях планируется добавить JWT токены.

Структуры данных

Company (Компания)

json

```
{
  "id": 1,
  "name": "ТОО Рога и Копыта",
  "email": "company@example.com",
  "phone": "+77771234567",
  "password": "hashed_password"
}
```

Template (Шаблон)

json

```
{
  "id": 1,
  "company_id": 1,
  "name": "Договор поставки",
  "file_path": "uploads/templates/company_1_template.docx",
  "created_at": "2024-01-15T10:30:00Z"
}
```

Contract (Контракт)

json

```
{
  "id": 1,
  "company_id": 1,
  "template_id": 1,
  "contract_token": "uuid-string",
  "generated_pdf_path": "path/to/generated.pdf",
  "client_filled": false,
  "created_at": "2024-01-15T10:30:00Z"
}
```

Signature (Подпись)

json

```
{
  "id": 1,
  "contract_id": 1,
  "client_name": "Иванов Иван",
  "client_iin": "123456789012",
  "client_phone": "+77771234567",
  "signed_at": "2024-01-15T10:30:00Z",
  "method": "sms"
}
```

API Endpoints



Companies (Компании)

Регистрация компании

http

POST /companies/register

Content-Type: application/json

```
{
  "name": "ТОО Рога и Копыта",
  "email": "company@example.com",
  "phone": "+77771234567",
  "password": "password123"
}
```

Ответ: 201 Created

Получить все компании

http

GET /companies

Ответ:

json

```
[
  {
    "id": 1,
    "name": "ТОО Рога и Копыта",
    "email": "company@example.com",
    "phone": "+77771234567"
  }
]
```

Получить компанию по ID

http

GET /companies/id/{id}

Получить компанию по телефону

http

GET /companies/phone/{phone}

Обновить компанию

http

PUT /companies/{id}

Content-Type: application/json

```
{
  "name": "Новое название",
  "email": "new@example.com",
  "phone": "+77771234567",
  "password": "newpassword"
}
```

Удалить компанию

http

DELETE /companies/{id}

Templates (Шаблоны)

Загрузить шаблон

http

POST /templates/upload

Content-Type: multipart/form-data

Form fields:

- file: template.docx (файл)
- company_id: 1
- name: "Договор поставки"

Ответ:

json

```
{
  "id": 1,
  "company_id": 1,
  "name": "Договор поставки",
  "file_path": "uploads/templates/company_1_template.docx",
  "created_at": "2024-01-15T10:30:00Z"
}
```

Получить шаблон по ID

http

GET /templates/{id}

Получить шаблоны компании

http

GET /templates/company/{company_id}

Обновить шаблон

http

PUT /templates/{id}

Content-Type: application/json

```
{
  "name": "Новое название шаблона",
  "file_path": "new/path/to/template.docx"
}
```

Удалить шаблон

http

DELETE /templates/{id}



Contracts (Контракты)

Создать контракт

http

POST /contracts

Content-Type: application/json

```
{
  "company_id": 1,
  "template_id": 1,
  "generated_pdf_path": "path/to/generated.pdf",
  "client_filled": false
}
```

Ответ:

json

```
{
  "id": 1,
  "company_id": 1,
  "template_id": 1,
  "contract_token": "generated-uuid",
  "generated_pdf_path": "path/to/generated.pdf",
  "client_filled": false,
  "created_at": "2024-01-15T10:30:00Z"
}
```

Получить контракт по ID

http

GET /contracts/{id}

Получить контракт по токену

http

GET /contracts/token/{token}

Получить контракты компании

http

GET /contracts/company/{company_id}

Обновить контракт

http

PUT /contracts/{id}

Content-Type: application/json

```
{  
  "generated_pdf_path": "new/path/to/generated.pdf",  
  "client_filled": true  
}
```

Удалить контракт

http

DELETE /contracts/{id}

Signatures (Подписи)

Создать подпись

http

POST /signatures

Content-Type: application/json

```
{
  "contract_id": 1,
  "client_name": "Иванов Иван",
  "client_iin": "123456789012",
  "client_phone": "+77771234567",
  "method": "sms"
}
```

Подписать контракт

http

POST /signatures/sign

Content-Type: application/json

```
{
  "contract_id": 1,
  "client_name": "Иванов Иван",
  "client_iin": "123456789012",
  "client_phone": "+77771234567",
  "method": "sms",
  "company_id": 1
}
```

Получить подпись по ID

http

GET /signatures/{id}

Получить подписи по контракту

http

GET /signatures/contract/{contract_id}

Удалить подпись

http

DELETE /signatures/{id}

Contract Fields (Поля контракта)

Создать поле контракта

http

POST /contract-fields

Content-Type: application/json

```
{
  "contract_id": 1,
  "field_name": "client_name",
  "field_type": "text"
}
```

Получить поля контракта

http

GET /contract-fields/{contract_id}

Company Balances (Балансы компаний)

Создать баланс

http

POST /company-balances

Content-Type: application/json

```
{
  "company_id": 1,
  "sms_signatures": 10,
  "ecp_signatures": 5
}
```

Получить баланс компании

http

GET /company-balances/{company_id}

Обновить баланс

http

PUT /company-balances/{company_id}

Content-Type: application/json

```
{  
  "sms_signatures": 15,  
  "ecp_signatures": 8  
}
```

Удалить баланс

http

DELETE /company-balances/{company_id}

Tariff Plans (Тарифные планы)

Создать тарифный план

http

POST /tariff-plans

Content-Type: application/json

```
{  
  "name": "Базовый",  
  "discount": 10.00,  
  "from_count": 10,  
  "to_count": 50,  
  "is_active": true  
}
```

Получить все тарифные планы

http

GET /tariff-plans

Получить тарифный план по ID

http

GET /tariff-plans/{id}

Обновить тарифный план

http

PUT /tariff-plans/{id}

Content-Type: application/json

```
{
  "name": "Премиум",
  "discount": 15.00,
  "from_count": 10,
  "to_count": 50,
  "is_active": true
}
```

Удалить тарифный план

http

DELETE /tariff-plans/{id}

Payment Requests (Платежные запросы)

Создать платежный запрос

http

POST /payment_requests

Content-Type: application/json

```
{
  "company_id": 1,
  "tariff_plan_id": 1,
  "sms_count": 10,
  "ecp_count": 5,
  "total_amount": 2500.00,
  "status": "pending",
  "payment_url": "https://payment.gateway.com/pay/123",
  "payment_ref": "PAY123456"
}
```

Получить платежный запрос по ID

http

GET /payment_requests/{id}

Получить платежные запросы компании

http

GET /payment_requests/company/{company_id}

Обновить платежный запрос

http

PUT /payment_requests/{id}

Content-Type: application/json

```
{  
  "status": "paid",  
  "paid_at": "2024-01-15T10:30:00Z"  
}
```

Удалить платежный запрос

http

DELETE /payment_requests/{id}



Statistics (Статистика)

Получить статистику компании

http

GET /stats/company/{company_id}

Ответ:

json

```
{
  "company_id": 1,
  "total_signed": 150,
  "active_contracts": 25,
  "signed_this_month": 12,
  "last_signed_contracts": [
    {
      "contract_id": 1,
      "client_name": "Иванов Иван",
      "template_name": "Договор поставки",
      "signed_at": "2024-01-15T10:30:00Z",
      "status": "Подписан"
    }
  ]
}
```

Коды ошибок

Код	Описание
200	OK - Успешный запрос
201	Created - Ресурс создан
400	Bad Request - Неверный запрос
404	Not Found - Ресурс не найден
500	Internal Server Error - Внутренняя ошибка сервера

Примеры использования

Типичный флоу создания и подписания контракта:

1. Регистрация компании

```
http
POST /companies/register
```

2. Загрузка шаблона

```
http
POST /templates/upload
```

3. Создание контракта

http

POST /contracts

4. Добавление полей контракта

http

POST /contract-fields

5. Подписание контракта

http

POST /signatures/sign

Пример фронтенд компонента (React)

// Пример компонента для создания компании

```
const CompanyRegistration = () => {
  const [formData, setFormData] = useState({
    name: '',
    email: '',
    phone: '',
    password: ''
  });

  const handleSubmit = async (e) => {
    e.preventDefault();
    try {
      const response = await fetch('http://localhost:4000/companies/register', {
        method: 'POST',
        headers: {
          'Content-Type': 'application/json',
        },
        body: JSON.stringify(formData)
      });

      if (response.ok) {
        alert('Компания успешно зарегистрирована!');
      }
    } catch (error) {
      console.error('Ошибка:', error);
    }
  };

  return (
    <form onSubmit={handleSubmit}>
      <input
        type="text"
        placeholder="Название компании"
        value={formData.name}
        onChange={(e) => setFormData({...formData, name: e.target.value})}
      />
      <input
        type="email"
        placeholder="Email"
        value={formData.email}
        onChange={(e) => setFormData({...formData, email: e.target.value})}
      />
      <input
        type="tel"
        placeholder="Телефон"
        value={formData.phone}
```

```
    onChange={(e) => setFormData({...formData, phone: e.target.value})}
  />
  <input
    type="password"
    placeholder="Пароль"
    value={formData.password}
    onChange={(e) => setFormData({...formData, password: e.target.value})}
  />
  <button type="submit">Зарегистрировать</button>
</form>
);
};
```

Примечания

- Все даты возвращаются в формате ISO 8601
- Файлы загружаются через multipart/form-data
- API поддерживает CORS для всех доменов
- Методы подписи: "sms" или "esp"
- Статусы платежей: "pending", "paid", "cancelled", "expired"

Поддержка

Для вопросов по API обращайтесь к команде разработки.