Formify - Seeded Data & Admin Guide

🚀 Quick Start

1. Seed Sample Data

```
# Clear existing data and seed new sample data
python manage.py seed_data --clear

# Or just add data without clearing existing
python manage.py seed_data
```

2. Start Development Server

```
python manage.py runserver
```

3. Access Admin Interface

- **URL**: http://localhost:8000/admin/
- **Admin User**: admin@example.com / testpass123
- **Test Users**: test1@example.com, test2@example.com, test3@example.com / testpass123

4. Access API Documentation

- **Swagger UI**: http://localhost:8000/api/docs/
- **ReDoc**: http://localhost:8000/api/redoc/

📊 Seeded Data Overview

👥 Users (4 users)

- **admin@example.com** - Admin user with full permissions
- **test1@example.com** - Regular user (created forms and processes)
- **test2@example.com** - Regular user (created employee surveys)
- **test3@example.com** - Regular user (created event management)

📋 Forms (3+ forms)

1. **Customer Feedback Form** (Public)
    - Overall Satisfaction (Select)
    - Product Quality Rating (Select)
    - Comments (Text)
    - Would you recommend us? (Checkbox)

2. **Employee Satisfaction Survey** (Private - password: employee2024)
    - Job Satisfaction (Select)
    - Work-Life Balance (Select)
    - Management Support (Select)
    - Additional Comments (Text)

3. **Product Feature Request** (Public)
    - Feature Category (Select)
    - Feature Description (Text)
    - Priority Level (Select)
    - Contact for Follow-up (Checkbox)

4. **Employee Self Assessment** (Private - password: selfeval2024)
    - Self Rating (Select)
    - Strengths (Text)
    - Areas for Improvement (Text)
    - Goals for Next Period (Text)

🔄 Processes (2 processes)

1. **Customer Onboarding Process** (Linear, Public)
    - Step 1: Account Setup (Customer Feedback Form)
    - Step 2: Profile Completion (Product Feature Request)

2. **Employee Performance Evaluation** (Free, Private - password: eval2024)
    - Step 1: Self Assessment (Employee Satisfaction Survey)
    - Step 2: Detailed Self Review (Employee Self Assessment)

📂 Categories (4 categories)

- **Customer Feedback** - Forms and processes for customer feedback
- **Employee Surveys** - Internal surveys and feedback forms
- **Product Research** - Research forms for product development
- **Event Management** - Forms for managing events and registrations

📝 Sample Data

- **9 Responses** with complete answers
- **15 Form Views** for analytics
- **5 Entity-Category Associations**

🛠️ Admin Interface Features

Enhanced Admin Features

- **Form Management**: View forms with field counts, response counts, and view statistics

- **Field Management**: Preview field options and choices
- **Process Management**: Manage processes with step counts and workflow
- **Category Management**: Organize forms and processes by categories
- **Response Management**: View responses with inline answers
- **Analytics**: Track form views and response statistics

Admin Interface Improvements

- **Better List Views**: More informative columns with counts and statistics
- **Enhanced Fieldsets**: Organized sections for better data management
- **Inline Editing**: Edit related objects directly from parent objects
- **Search & Filtering**: Advanced search and filtering capabilities
- **Read-only Fields**: Protected fields that show calculated values
- **API Links**: Direct links to admin pages for easy navigation

🔌 API Testing

Test Script

Run the included API test script:

```
python test_apis.py
```

Manual API Testing

1. **Get JWT Token**:

   ```
   curl -X POST http://localhost:8000/api/v1/accounts/login/ \
     -H "Content-Type: application/json" \
     -d '{"email": "test1@example.com", "password": "testpass123"}'
   ```

2. **List Forms**:

   ```
   curl -X GET http://localhost:8000/api/v1/forms/forms/ \
     -H "Authorization: Bearer YOUR_TOKEN"
   ```

3. **Submit Public Form**:

   ```
   curl -X POST
   http://localhost:8000/api/v1/forms/public/forms/FORM_ID/submit/ \
     -H "Content-Type: application/json" \
     -d '{"answers": [{"field_id": "FIELD_ID", "value": "Test Answer"}]}'
   ```

Available API Endpoints

Authentication

- `POST /api/v1/accounts/register/` - User registration
- `POST /api/v1/accounts/login/` - User login
- `POST /api/v1/accounts/logout/` - User logout
- `POST /api/v1/accounts/token/refresh/` - Refresh JWT token

Forms Management

- `GET /api/v1/forms/forms/` - List user's forms
- `POST /api/v1/forms/forms/` - Create new form
- `GET /api/v1/forms/forms/{id}/` - Get form details
- `PUT /api/v1/forms/forms/{id}/` - Update form
- `DELETE /api/v1/forms/forms/{id}/` - Delete form

Fields Management

- `GET /api/v1/forms/fields/` - List user's fields
- `POST /api/v1/forms/fields/` - Create new field
- `GET /api/v1/forms/fields/{id}/` - Get field details
- `PUT /api/v1/forms/fields/{id}/` - Update field
- `DELETE /api/v1/forms/fields/{id}/` - Delete field

Public Forms

- `GET /api/v1/forms/public/forms/` - List public forms
- `GET /api/v1/forms/public/forms/{id}/` - Get public form details
- `POST /api/v1/forms/public/forms/{id}/submit/` - Submit form response

Private Forms

- `POST /api/v1/forms/private/forms/validate/` - Validate private form access

Processes

- `GET /api/v1/forms/processes/` - List user's processes
- `POST /api/v1/forms/processes/` - Create new process
- `GET /api/v1/forms/processes/{id}/` - Get process details

Process Workflow

- `GET /api/v1/forms/workflow/process-steps/` - Get process steps
- `GET /api/v1/forms/workflow/current-step/` - Get current step
- `POST /api/v1/forms/workflow/complete-step/` - Complete process step
- `GET /api/v1/forms/workflow/progress/` - Get process progress

Categories

- `GET /api/v1/forms/categories/` - List user's categories
- `POST /api/v1/forms/categories/` - Create new category
- `GET /api/v1/forms/categories/{id}/` - Get category details

Analytics

- `GET /api/v1/forms/responses/by_form/` - Get responses by form
- `GET /api/v1/forms/answers/by_response/` - Get answers by response
- `GET /api/v1/forms/answers/field_statistics/` - Get field statistics

🎯 Testing Scenarios

1. Form Creation & Management

- Create forms with different field types
- Test public/private form access
- Submit responses and view analytics

2. Process Workflow

- Create linear and free processes
- Test step completion workflow
- Monitor process progress

3. Category Organization

- Organize forms and processes by categories
- Test category-based filtering and management

4. Analytics & Reporting

- View form view statistics
- Analyze response data
- Generate field-level statistics

🔧 Development Commands

```bash
# Run tests
python manage.py test

# Create migrations
python manage.py makemigrations

# Apply migrations
python manage.py migrate

# Create superuser
python manage.py createsuperuser
```

```
# Seed data
python manage.py seed_data --clear

# Generate API schema
python manage.py spectacular --file schema.yml
```

📖 Documentation

- **API Documentation**: http://localhost:8000/api/docs/
- **Admin Interface**: http://localhost:8000/admin/
- **Project README**: See main README.md for project overview

---

Happy Testing! 🎉