

# Система настроек

В системе, помимо кода приложения, присутствуют настройки, которые отвечают за бизнес-логику приложения и его визуальную составляющую. Эти настройки позволяют гибко управлять различными модулями системы.

## Основные возможности настроек:

Регулирование элементов системы:

- Отчеты
- Формы
- График
- Правила

## Технология настроек:

- JSONata: Настройки реализованы с использованием JSONata — языка для обработки и трансформации данных в формате JSON
- Rudderstack: Также используются настройки на RudderStack, который представляет собой язык программирования для трансформации JSON-данных. RudderStack включает в себя JSON Template Engine, который упрощает преобразование данных из одного формата в другой, минимизируя накладные расходы на выполнение и улучшая производительность. Он генерирует оптимизированный JavaScript-код из шаблонов трансформации, что позволяет легко добавлять новые шаблоны и поддерживать их.

## Доступ к настройкам:

Доступ к настройкам осуществляется через базу данных PostgreSQL с использованием DBEaver.

Настройки для различных модулей системы организованы в отдельные таблицы базы данных, что обеспечивает удобство управления и гибкость в настройках. Ниже представлены основные таблицы, используемые для хранения настроек:

- Таблица forms - Используется для настройки форм в системе.
- Таблица flex\_settings - Применяется для гибких настроек, позволяя адаптировать систему под специфические требования.
- Таблица mail\_reports - Служит для настройки рассылаемых актов, обеспечивая автоматизацию отчетности.
- Таблица reports - Используется для настройки печатных отчетов, позволяя формировать документы в нужном формате.
- Таблица settings - Применяется для настройки различных модулей системы, обеспечивая их корректную работу.
- Таблица settingTypes - Используется для хранения настроек по умолчанию, определяющих базовые параметры системы.

## Использование настроек:

Настройки в системе могут быть заданы на нескольких уровнях, что обеспечивает гибкость и индивидуальный подход к каждому пользователю.

- Настройки для ролей:
  - Каждая роль в системе может иметь свои собственные настройки, которые применяются ко всем пользователям, работающим под данной ролью. Это позволяет управлять доступом и функциональностью на уровне группы пользователей.
- Персональные настройки для пользователей:
  - Кроме настроек, связанных с ролями, каждая учетная запись пользователя может иметь свои персональные настройки. Эти настройки имеют приоритет над настройками, заданными для роли.
- Настройки по умолчанию:
  - Если для пользователя не заданы персональные настройки и для его роли отсутствуют соответствующие настройки, система применяет настройки по умолчанию. Эти настройки определяют базовые параметры и функциональность.

## Логика применения настроек:

При определении активных настроек система следует следующему порядку приоритета:

1. Персональные настройки пользователя:
  - Если у пользователя есть индивидуальные настройки, они применяются в первую очередь.

## 2. Настройки роли:

- Если персональные настройки отсутствуют, система проверяет настройки, связанные с ролью пользователя.

## 3. Настройки по умолчанию:

- Если ни персональные настройки, ни настройки роли не заданы, система использует настройки по умолчанию.

Таким образом, данная иерархия позволяет обеспечить максимальную гибкость и адаптивность системы, удовлетворяя потребности как отдельных пользователей, так и групп пользователей с одинаковыми ролями.

---

Revision #2

Created 28 July 2025 06:20:47 by Семен Долгов

Updated 28 July 2025 07:18:43 by Семен Долгов