**Техническое задание: Модуль “Шаблоны” административной панели PaaS-платформы**

## Общая информация

Модуль “Шаблоны” предназначен для управления типовыми конфигурациями рабочих пространств (workspace). Он состоит из трех вкладок: **Список шаблонов**, **Разделы платформы**, **Структура таблиц**.

Раздел доступен по адресу /admin/templates и предназначен только для глобальных администраторов платформы.

## Вкладка 1. Список шаблонов

### Назначение

Просмотр, создание и редактирование шаблонов workspace.

### UI (Frontend)

Таблица со следующими колонками:

* Название шаблона
* Тип (client, supplier)
* Версия
* Активен (да/нет)
* Применён по умолчанию (да/нет)
* Кнопка “Открыть / Редактировать”
* Кнопка “Создать шаблон”

### При открытии шаблона

Редактор шаблона содержит:

* Структура таблиц (на основе глобальных схем)
* Разделы (из вкладки 2)
* Список фич и ролей
* Привязанный тариф
* История версий (template\_versions)
* Кнопка “Применить к workspace”

## Вкладка 2. Разделы платформы

### Назначение

Управление разделами (views/features), которые могут быть включены в шаблоны и тарифы.

### UI

Таблица разделов со столбцами:

* Название
* Родительский раздел (если есть)
* Тип доступа (открытый, закрытый)
* Привязанная таблица
* Флаг “Системный”
* Статус (вкл/выкл)

### Форма создания/редактирования раздела

* Название (обязательно)
* Родительский раздел (nullable)
* Тип доступа: open / restricted
* Привязанная таблица (выбор из существующих)
* Описание (optional)

### Backend API

* GET /api/admin/sections
* POST /api/admin/sections
* PATCH /api/admin/sections/:id
* DELETE /api/admin/sections/:id

### Таблица: sections

CREATE TABLE sections (  
 id UUID PRIMARY KEY,  
 name TEXT NOT NULL,  
 parent\_id UUID REFERENCES sections(id),  
 table\_name TEXT,  
 is\_system BOOLEAN DEFAULT FALSE,  
 access\_type TEXT CHECK (access\_type IN ('open', 'restricted')),  
 description TEXT,  
 created\_at TIMESTAMP DEFAULT NOW()  
);

## Вкладка 3. Структура таблиц

### Назначение

Централизованное управление схемами таблиц, используемых в шаблонах и разделах.

### Категории таблиц (аналог 1С)

* 📗 Справочники
* 📘 Документы
* 📙 Регистры
* 📒 Журналы
* 📓 Отчёты
* 📕 Обработки

### UI

* Структура по категориям (accordion или табы)
* Список таблиц в каждой категории:
  + Название
  + Код
  + Тип
  + Кнопки [Редактировать] / [Удалить] / [Создать на основе]

### Форма создания/редактирования таблицы

* Название
* Код (уникальный slug)
* Тип (directory, document, register, …)
* Системные поля (галочки: created\_at, updated\_at и т.п.)
* Пользовательские поля:
  + Название
  + Тип (string, number, boolean, date, reference, select)
  + Обязательность
  + Связь с другой таблицей (если reference)
  + Флаг is\_system

### Backend API

* GET /api/admin/table-schemas
* POST /api/admin/table-schemas
* PATCH /api/admin/table-schemas/:id
* DELETE /api/admin/table-schemas/:id

### Таблицы БД:

CREATE TABLE global\_table\_schemas (  
 id UUID PRIMARY KEY,  
 name TEXT NOT NULL,  
 code TEXT UNIQUE NOT NULL,  
 type TEXT CHECK (type IN ('directory', 'document', 'register', 'journal', 'report', 'procedure')),  
 is\_system BOOLEAN DEFAULT FALSE,  
 created\_at TIMESTAMP DEFAULT NOW()  
);  
  
CREATE TABLE global\_table\_fields (  
 id UUID PRIMARY KEY,  
 schema\_id UUID REFERENCES global\_table\_schemas(id) ON DELETE CASCADE,  
 name TEXT NOT NULL,  
 label TEXT NOT NULL,  
 type TEXT,  
 is\_required BOOLEAN DEFAULT FALSE,  
 is\_system BOOLEAN DEFAULT FALSE,  
 reference\_table TEXT,  
 options JSONB  
);

## Зависимости и связи

| Объект | Связан с |
| --- | --- |
| section | global\_table\_schemas |
| template | sections, features, roles |
| workspace | template, plan, db\_uri |
| plan\_features | фильтруют sections |
| custom\_fields | применимы только в workspace |

## Разработать:

### Backend

* Модели и миграции sections, global\_table\_schemas, global\_table\_fields
* CRUD API для всех вкладок
* Логика версионирования шаблонов (template\_versions)

### Frontend

* UI для всех вкладок (таблицы, формы, модалки)
* State-менеджмент выбранного шаблона и его структур
* Версионирование шаблонов (опционально)

## Следующие шаги:

* Разработка API и UI прототипа
* Проверка сценариев копирования шаблонов в workspace
* Интеграция с тарифами и ACL