

Модели данных (EntityModel)

В данной главе описаны базовые понятия что такое модель данных, структура модели и принцип работы с ними через API.

Модель данных может использоваться:

1. во внешних приложениях, как основной источник данных;
2. в системе синхронизации, как источник данных для описания объектов;
3. при формировании выходных документов, как источник данных для шаблонов.

- [Создание модели данных](#)
- [Предварительный просмотр](#)
- [Работа с моделью данных через API](#)
- [Модель данных для печати бланков](#)

Создание модели данных

Модель данных представляет собой описание объекта предметной области, которого можно представить в виде структуры в формате JSON. Все узлы структуры должны быть названы латинскими буквами, слова без пробелом в нижнем регистре.

Для создания новой **модели данных** перейдите в раздел разработчика в проекте [Сервисы ТПУ](#)

Вы видите только модели данных, которые вы создали (авторы) или с вами поделились другие разработчики.

В правом верхнем углу нажмите кнопку "Добавить" и заполните все основные поля.

Параметр	Значение
Название	Краткое название модели данных
Алиас	Уникальный алиас модели, будет использовать в API и в синхронизации
Описание	Полное описание модели данных
Вид доступа к модели данных	Уровень доступа <ol style="list-style-type: none">1. Всем2. Только пользователям (требуется авторизация)3. Только корпоративным пользователям (действующий студент или сотрудник)
Время жизни кеша в секундах	Время жизни кэша сформированного объекта модели данных в памяти, для ускорения работы. Если кэширование не требуется - значение в этом поле нужно указать 0.
Приложения модели	Перечень приложений, которые могут обращаться к указанной модели данных. Каждое приложение должно передавать свой <u>публичный ключ доступа</u> .

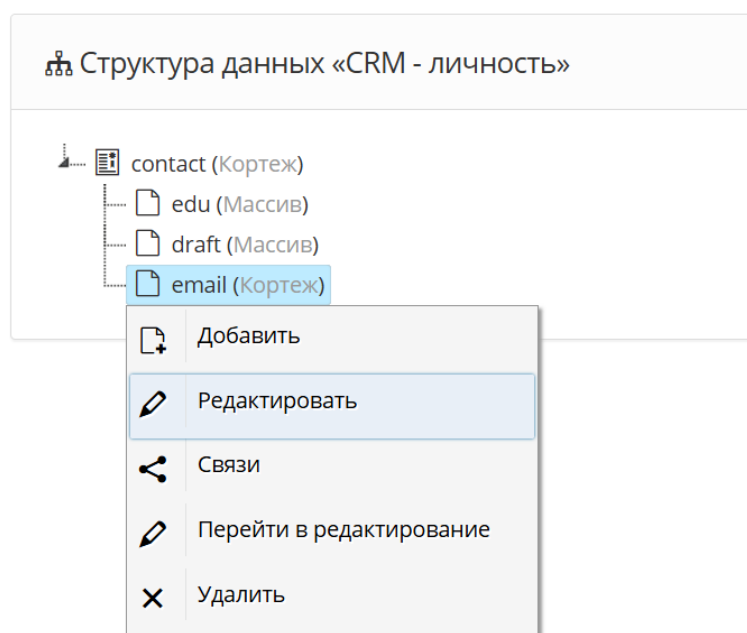
Параметр	Значение
Источник для корневого узла	<p>По умолчанию для структуры модели данных создается древовидное представление с корневым элементом (root). Данное значение в дальнейшем можно будет поменять.</p> <p>У одного объекта модели данных может быть только один корневой элемент.</p>

В качестве источников данных для узлов структуры могут быть использованы только [3 типа динамических источников данных](#).

1. Кортеж
2. Массив
3. Список

Редактирование описания структуры модели данных осуществляется через контекстное меню в режиме редактирования. Для добавления / редактирование / удаления узлов - требуется воспользоваться соответствующими элементами меню.

Главная / Модели данных / Структура данных «CRM - личность»



При редактировании узла требуется указать алиас (должен быть обязательно совместим со стандартом JSON) и [источник данных](#) (должен быть подготовлен заранее)

Редактирование узла

Динамическая модель: CRM - личность

Путь до узла: contact (Кортеж)

Алиас узла *

Латинскими буквами (JSON совместимые символы)

Источник данных *

Источники доступные для приложения модели данных

Заполните форму и нажмите «Сохранить»

* Поля обязательные для заполнения

Для связи дочерних элементов с родительскими требуется указать хотя бы один внешний ключ, пересечение может быть организовано по составному ключу.

Связи в источниках данных

Динамическая модель: CRM - личность

Путь до узла: contact (Кортеж) > email (Кортеж)

Источник данных: Email абитуриента (синхронизация)

Вышестоящий источник данных: Личность (синхронизация)

Ключ *

Внешний ключ *

Заполните форму и нажмите «Сохранить»

* Поля обязательные для заполнения

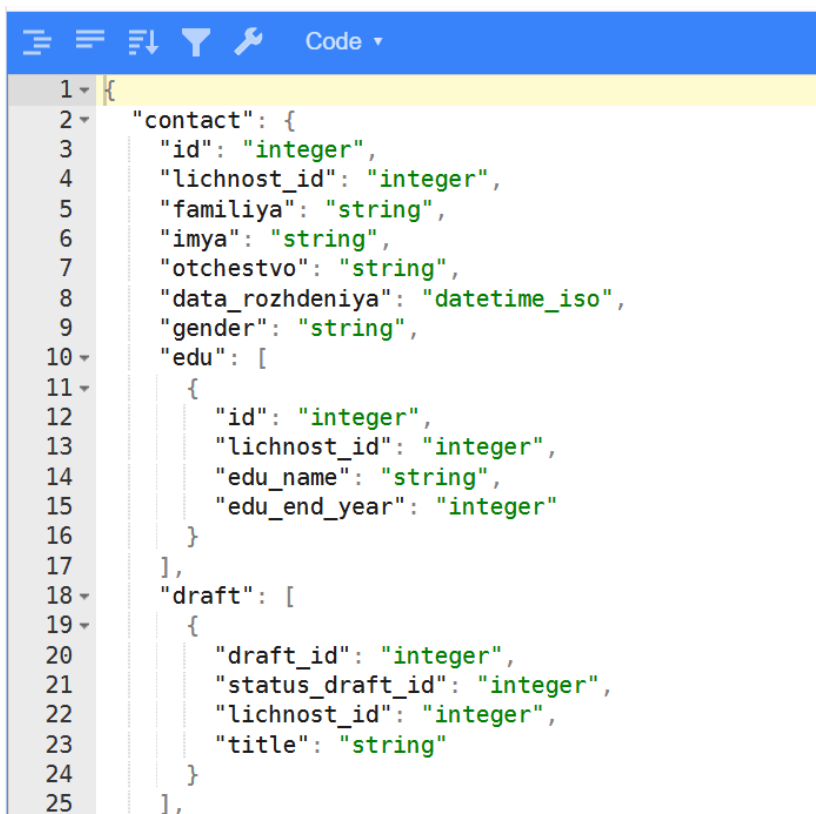
Для оперативной корректировки источника данных в контекстном меню есть ссылка на переход в режим редактирование. Источник данных можно редактировать, если вы его автор или с вами поделились доступом.

Предварительный просмотр

После создания описание модели данных, для проверки работы можно перейти в режим предварительный просмотр.

Предварительный просмотр формирует 3 различных описания.

1. Структура (тут представлены все экспортируемые поля с указанием типа данных)



```
1 {
2   "contact": {
3     "id": "integer",
4     "lichnost_id": "integer",
5     "familiya": "string",
6     "imya": "string",
7     "otchestvo": "string",
8     "data_rozhdeniya": "datetime_iso",
9     "gender": "string",
10    "edu": [
11      {
12        "id": "integer",
13        "lichnost_id": "integer",
14        "edu_name": "string",
15        "edu_end_year": "integer"
16      }
17    ],
18    "draft": [
19      {
20        "draft_id": "integer",
21        "status_draft_id": "integer",
22        "lichnost_id": "integer",
23        "title": "string"
24      }
25    ],
```

2. Комментарии (в этом режиме все поля содержат описательную часть, комментарий к назначению данного поля)

```
1 {
2   "contact": {
3     "id": "object_id",
4     "lichnost_id": "ИД Личности",
5     "familiya": "Фамилия",
6     "imya": "Имя",
7     "otchestvo": "Отчество",
8     "data_rozhdeniya": "Дата рождения",
9     "gender": "Пол",
10    "edu": [
11      {
12        "id": "ИД Обучения",
13        "lichnost_id": "ИД Личности",
14        "edu_name": "Название учебного заведения",
15        "edu_end_year": "Год окончания"
16      }
17    ],
18    "draft": [
19      {
20        "draft_id": "ИД заявки на поступление",
21        "status_draft_id": "ИД статуса заявки",
22        "lichnost_id": "ИД Личности",
23        "title": "Название сервиса"
24      }
25    ],

```

3. Пример (Загруженный объект по умолчанию из базы, с реальными значениями)

```
1 {
2   "contact": {
3     "id": 7487,
4     "lichnost_id": 193,
5     "familiya": "Иванов",
6     "imya": "Иван",
7     "otchestvo": "Иванович",
8     "data_rozhdeniya": "1994-09-01T00:00:00+0800",
9     "gender": "Мужской",
10    "edu": [
11      {
12        "id": 22642,
13        "lichnost_id": 193,
14        "edu_name": "Томский политехнический университет",
15        "edu_end_year": 2018
16      },
17      {
18        "id": 1570,
19        "lichnost_id": 193,
20        "edu_name": "Томский политехнический университет",
21        "edu_end_year": 2005
22      },
23      {
24        "id": 239,
25        "lichnost_id": 193,

```

После проверки модели данных, её можно использовать через API интеграцию или получать данные через сервер Синхронизации, а так же использовать как источник данных при формировании документов для СОУД.

по API можно запросить:

1. список объектов
2. один объект по первичному ключу

Через сервер синхронизации:

1. при изменении свойств объект - будет вызван API во внешней системе, куда будет передан один объект или список объектов.

При создании шаблона документа можно указать модель данных, как основной источник данных при формировании документа в формате PDF (шаблон может быть в формате Word или LaTeX).

Работа с моделью данных через API

С моделями данных можно работать через API только на чтение.

Для получения списка доступных моделей данных, необходимо вызвать следующий метод API, указав [публичный ключ приложения](#).

```
https://api.tpu.ru/v2/entity
```

Для получения [структуры/описания модели данных](#) можно использовать следующие методы API.

```
https://api.tpu.ru/v2/entity/structure/<id>
```

Для получения возможного списка параметров, которые можно использовать в модели данных (*данный список может быть пуст*)

```
https://api.tpu.ru/v2/entity/parametr/<id>
```

Для получения возможных фильтров, которые используются при работе с корневым элементом модели данных.

```
https://api.tpu.ru/v2/entity/filter/<id>
```

Фильтры доступны только при работе с моделями данных организованных на источниках данных [типа массив](#).

Сортировка выборки данных

```
sort=[direction][column]
```

Для обратной сортировки добавьте минус перед названием столбца

Пример:

```
?sort=title,-created
```

Фильтрация выборки данных:

```
filter[type][op]=value
```

Допустимые сравнения:

Оператор	Сравнение	Описание
eq	equal (=)	Эквивалентно (равно) указанному значению
like	text search	Поиск по тексту
gte	>=	Больше и равно
lte	<=	Меньше и равно
gt	>	Больше
lt	<	Меньше

Пример:

```
?filter[created][eq]=2019-07-16
```

Все параметры можно комбинировать

Модель данных для печати бланков