

Структура метаданных

Регистрация метаданных

Метаданные по сути регистрируются в javascript после загрузки кода платформы EntUI и до его использования. Загрузка производится вызовами функции EntUI_add_entity, по одному вызову на каждую сущность.

EntUI_add_entity(entity_name, meta)

entity_name - имя сущности, строка. Должно быть уникальным, может содержать латинские буквы, цифры и символ подчеркивания. Регистрочувствительно.

meta - метаданные в формате json.

Структура метаданных

Структура метаданных во многом определяется используемыми плагинами. Ниже будет описана структура поддерживаемая набором плагинов из базовой поставки платформы EntUI.

Секции верхнего уровня.

```
{
  "table_header": "...",
  "window_header": "...",
  "form_header": "...",
  "form_header_new": "...",
  "label_template": "...",
  "button_label_create": "...",
  "opts": {},
  "cols": [],
  "filters": [],
  "form": [],
  "form_tabs": [],
  "includes": []
}
```

-
- **table_header** - заголовок над таблицей, строка.
 - **window_header** - шапка диалога с таблицей выбора, строка.
 - **form_header** - шапка диалога с формой редактирования, строка.
 - **form_header_new** - шапка диалога с формой создания, строка.
 - **label_template** - заголовок над формой редактирования, строка с подстановками в ruby-формате. В качестве подстановок допустимы только поля формы из секции form (в частности - невидимые).
 - **button_label_create** - кнопка создания новой сущности над таблицей, строка.
 - **opts** - общие настройки, хеш.
 - **cols** - колонки таблицы, массив.
 - **filters** - фильтры таблицы, массив.
 - **form** - поля формы, массив.
 - **form_tabs** - табы со списками сущностей следующего уровня, массив.
 - **includes** - оптимизация запросов списка, подробно описано в разделе интерфейса с RubyOnRails.

opts

Опции и "значения по умолчанию":

```
{
  "root": {"window": false, "create": true, "edit": true, "paging": true, "filters": true, "hidden_cols": [], "interface": "EntUIDataTable"},
  "uplink": {"window": true, "create": true, "edit": true, "paging": true, "filters": true, "hidden_cols": [], "interface": "EntUIDataTable"},
  "downlink": {"window": false, "create": true, "edit": true, "paging": true, "filters": false, "hidden_cols": [], "interface": "..."}
```

```
EntUIDataTable"},
  "form": {"interface": "EntUIEditForm"}
}
```

Разделы:

- **root** - таблица появляющаяся первой, при запуске EntUI, с указанными при запуске сущностями.
- **uplink** - таблица всплывающая в диалоге, при нажатии кнопки редактирования ссылки (внутри формы).
- **downlink** - таблицы связанных сущностей на вкладках под формой.
- **form** - опции формы.

Опции:

- **window** - Открывать во всплывающем окне (диалоге), или не указанном месте.
- **create** - Есть кнопка создания новой сущности.
- **edit** - Двойной клик на строке таблицы открывает форму редактирования.
- **paging** - Таблица показывается в постраничном режиме.
- **filters** - Показывать ли фильтры в данном режиме.
- **hidden_cols** - Список имен скрытых колонок.
- **interface** - Плагин который поддерживает работу с таблицей или формой.

cols

Массив описаний колонок таблицы.

```
"cols": [
  {"atr": "...", "label": "..."},
  ...
]
```

Для открытия карточек редактирования сущностей, должна существовать колонка с именем 'id'. Эта колонка может быть невидимой (указывается в списке **hidden_cols** в **opts**).

label - заголовок колонки. Он может быть не указан, тогда в качестве заголовка берется название атрибута.

filters

Массив описаний фильтров над таблицей. Эта секция может отсутствовать, если ее нет, то нет и фильтров.

```
"filters": [
  {"atr": "...", "label": "..."},
  ...
]
```

Имя атрибута (atr) должно присутствовать в списке колонок **cols**, label может быть не указано, тогда оно берется из списка cols.

form

Описание полей формы редактирования.

В большинстве случаев эта секция должна быть. Если для какой-то сущности мы не хотим разрешать появление формы, то в секции opts следует установить:

```
"opts":{
  "root": {"create": false, "edit": false},
  "uplink": {"create": false, "edit": false},
  "downlink": {"create": false, "edit": false}
}
```

Формат описания формы:

```
"form": [
  [[ _fields_ ], { _row_opts_ } ],
  [[ _fields_ ]],
  ...
]
```

]

Форма описывается по рядам формирующей таблицы.

row_opts - html-опции ряда (то есть вешаются на <tr>). Могут отсутствовать, как показано во второй строчке примера.

[*fields*] - список метаданных по полям ряда. [{ *field* }, { *field* }, ...]

Каждое поле добавляет две ячейки: ячейку названия и ячейку контроля. Соответственно во всех рядах должно быть одинаковое количество полей. При необходимости список полей можно дополнить пустым хешем: [{ *field* }, {}]

Описание поля { *field* }, общие опции:

```
{"atr": "...", "kind": "...", "label": "...", "html": {}}
```

И специальные случаи:

```
{"atr": "...", "kind": "sel", "label": "...", "html": {}, "selector": ["название", "значение"], ["название оно же значение"], ... }
```

```
{"atr": "...", "kind": "link", "label": "...", "html": {}, "entity": "...", "id_atr": "id", "name_atr": "name" }
```

Любой параметр, кроме **atr**, может отсутствовать.

Параметры:

- **atr** - имя атрибута, приходящее с сервера.
- **kind** - тип поля (по умолчанию "text"). Сейчас поддерживаются:
 - **text** - ввод строки
 - **frozentext** - текстовое read only поле.
 - **date** - ввод даты (с календарем).
 - **datetime** - ввод даты и времени.
 - **sel** - выбор из перечисленных вариантов.
 - **span** - отображение текста без контроля ввода в чем-то аналогично frozentext.
 - **hidden** - невидимое поле. **Внимание!** Наличие в строке hidden-поля делает невидимой всю строку. Поэтому, hidden-поля лучше выносить в отдельную строку.
- **link** - ссылка на лежащую выше по иерархии сущность (uplink)
- **check** - чекбокс. Поддерживается отдельным плагином EntUIFldCheck.
- **label** - подпись к полю. Если отсутствует, используется значение **atr**.
- **html** - html-параметры. Имеет вид хеша: "html": {"td": {}, "input": {}}.
 - **td** - html-параметры устанавливаемые на ячейку формирующей таблицы, которая содержит контрол.
 - **input** - html-параметры устанавливаемые на контрол (чем бы он ни был, даже если это не input а что-то другое).
- **selector** - используется для полей типа sel, содержит список вариантов. ["название", "значение"], ["название оно же значение"], ...]
- **entity** - используется для полей типа link, содержит имя сущности на которую указывает ссылка. Если параметр не указан, имя сущности считается совпадающим с **atr**.
- **id_atr** - используется для полей типа link, содержит имя поля в таблице сущностей, которое берется в качестве идентификатора в ссылке. По умолчанию - "id".
- **name_atr** - используется для полей типа link, содержит имя поля в таблице сущностей, которое берется в качестве текста в ссылке. По умолчанию - "name".

Список типов полей может быть расширен с помощью плагинов. Пример - плагин EntUIFldCheck.

form_tabs

Табы на карточке под формой. Формат описания табов:

```
"form_tabs": [  
  {"name": "...", "kind": "downlink", "label": "...", "entity": "..."},  
  {"name": "...", "kind": "text", "label": "...", "text": "..."},  
  ...  
]
```

Параметры:

- **name** - название таба. Должно быть уникальным в рамках списка.
- **kind** - тип таба. В настоящий момент поддерживается только два (возможно, позднее добавим механизм плагинов):
 - **downlink** - таблица сущностей ссылающихся на данную (расположенных ниже по иерархии).

- **text** - просто текстовое поле на табе.
- **label** - подпись на ярлычке таба.
- **text** - текст на поле, для таба типа text.
- **entity** - имя связанной сущности, для таба типа downlink. Если не указано, считается равным параметру **name**.