

Подключение EntUI на страницу

Подключение метаданных

Метаданные должны быть подключены после загрузки кода EntUI до начала его использования. Подключение производится для каждой сущности отдельно, с помощью метода EntUI_add_entity:

```
EntUI_add_entity("имя сущности", { [[EntUI_meta|json-представление метаданных]] });
```

Регистрация каллбэков

Как и метаданные, каллбэки регистрируются после загрузки кода EntUI до начала его использования. Регистрация производится с помощью метода EntUI_add_callback:

```
EntUI_add_callback("имя сущности", "[[EntUI_callbacks|имя каллбэка]]", функция);
```

Например:

```
EntUI_add_callback("enum", "form_after_open", function(prefix, visible_data, target){...});
```

Размещение EntUI на странице

Размещение производится в два шага:

- Создается объект EntUI;
 - Строится корневая таблица сущностей на основе этого объекта.
- ```
var obj = new EntUI();
obj.start(target, "имя_сущности", "id_сущности");
```

Или например так:

```
(new EntUI()).start($(".place"), "имя_сущности");
```

В предложении одновременно может существовать несколько объектов EntUI. Они могут располагаться даже на одной отображаемой странице. Важно только, чтобы target у них был разный.

Если параметр "id сущности" не указан, то на базовом уровне будет выведена таблица. Если указан, то форма.

Замечание: элемент target не должен содержать ничего ценного, потому как при вызове метода start он очищается.

### Размещение как jQuery-плагин

---

То же самое, что в разделе выше можно записать в укороченном виде:

```
$(".place").EntUI("имя_сущности");
```

Если у метода jQuery.EntUI указан параметр "имя\_сущности", то объект создается и помещается в данные соответствующего узла.

Если "имя\_сущности" не указано, то извлекается существующий объект:

```
var obj = $(".place").EntUI();
```

### Используемые библиотеки

---

Javascript-библиотеки, которые должны быть подключены до EntUI:

- jQuery
- jQuery UI
- плагины jQuery: DataTables, DatePicker
- underscore.js

