

[dev](#), [Инвентаризация](#), [forms](#)

Инвентаризация: DEV: Forms

ActiveField

Поскольку у нас шаблон ввода в форме не стандартный, а {Label} с иконкой и {hint} завернутым в tooltip, {input}, {error}

Переделываем (наследуем) Activefield (от \yii\bootstrap5\ActiveField) следующим образом

- методы label и hint не заполняют `$this->parts['{label}']` и `$this->parts['{hint}']` отрендеренным HTML, а сохраняют текстовую часть в labelText и hintText атрибутах
- в методе render мы уже формируем `$this->parts['{label}']` на основании labelText и hintText. `$this->parts['{hint}']` остается пустой
 - на самом деле в \yii\bootstrap5\ActiveField есть метод renderLabel и делаем всю магию там
- `$this->parts['{hint}']` можно заполнить методом classicHint, тогда можно отобразить и классическую подсказку после Input

TextArea

Modal Popups

Была следующая проблема: если у формы явно не задан ID, то он (т.к. форма наследуется от виджета) формируется как префикс (w) + индекс (0), поэтому по умолчанию у формы ID = w0. И если на страницу, в которой уже есть форма, через Popup + Ajax подгрузить еще одну форму, то она соответственно тоже будет w0, т.к. она была первой на подгруженной странице. И тут возникает проблема с инициализацией JavaScript, т.к. к форме прилагается скрипт инициализации, и если формы называются одинаково, то возникает коллизия.

[пример скрипта инициализации](#)

```
jQuery('#w0').yiiActiveForm([{\n    "id": "aces-users_ids",\n    "name": "users_ids",\n    "container": ".field-aces-users_ids",\n    "input": "#aces-users_ids",\n    "error": ".invalid-feedback",\n    "enableAjaxValidation": true\n}, {\n    "id": "aces-comps_ids",\n    "name": "comps_ids",\n    "container": ".field-aces-comps_ids",\n    "input": "#aces-comps_ids",\n    "error": ".invalid-feedback",\n    "enableAjaxValidation": true\n}]
```

```
    }, {
      "id": "aces-services_ids",
      "name": "services_ids",
      "container": ".field-aces-services_ids",
      "input": "#aces-services_ids",
      "error": ".invalid-feedback",
      "enableAjaxValidation": true
    }, {
      "id": "aces-ips",
      "name": "ips",
      "container": ".field-aces-ips",
      "input": "#aces-ips",
      "error": ".invalid-feedback",
      "enableAjaxValidation": true
    }, {
      "id": "aces-comment",
      "name": "comment",
      "container": ".field-aces-comment",
      "input": "#aces-comment",
      "error": ".invalid-feedback",
      "enableAjaxValidation": true
    }, {
      "id": "aces-name",
      "name": "name",
      "container": ".field-aces-name",
      "input": "#aces-name",
      "error": ".invalid-feedback",
      "enableAjaxValidation": true
    }, {
      "id": "aces-access_types_ids",
      "name": "access_types_ids",
      "container": ".field-aces-access_types_ids",
      "input": "#aces-access_types_ids",
      "error": ".invalid-feedback",
      "enableAjaxValidation": true
    }, {
      "id": "aces-notepad",
      "name": "notepad",
      "container": ".field-aces-notepad",
      "input": "#aces-notepad",
      "error": ".invalid-feedback",
      "enableAjaxValidation": true
    }, {
      "id": "aces-acls_id",
      "name": "acls_id",
      "container": ".field-aces-acls_id",
      "input": "#aces-acls_id",
      "error": ".invalid-feedback",
      "enableAjaxValidation": true
    }
  ]
}
```

```
], {  
  "errorSummary": ".alert.alert-danger",  
  "errorCssClass": "is-invalid",  
  "successCssClass": "is-valid",  
  "validationStateOn": "input",  
  "validationUrl": "\/web\/aces\/validate?id=675"  
});
```

во избежание подобных коллизий мы делаем в init по умолчанию id вида className-id-form

From:
<http://wiki.reviakin.net/> - Wiki

Permanent link:
<http://wiki.reviakin.net/%D0%B8%D0%BD%D0%B2%D0%B5%D0%BD%D1%82%D0%B0%D1%80%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F:dev:forms?rev=1746118307>

Last update: 2025/05/01 16:51

