

# Field

На примере TextField

## Данные

```
[  
  "title" => "MoonShine",  
  "role_id" => 1  
];
```

## Данные

- **toValue** - значение для вывода в форме
- **value** - toValue, но с применением old() хелпера при валидации
- **toRawValue** - значение, полученное в результате работы метода resolveFill, без каких-либо модификаторов (отлично подходит для импорта/экспорта)
- **toFormattedValue** - значение для превью (везде кроме формы), полученное в результате трансформации через formattedCallback

```
Text::make(  
  label: "Title",  
  column: "title",  
  formatted: fn(array $data => $data['role_id'].$data['title']  
)  
)
```

```
value = null  
toValue = null  
toRawValue = null  
toFormattedValue = null
```

## Ввод значения

### resolveFill

Заполняем поле значением (Данные)

### prepareFill

Определяем, что именно из входных данных подходит для текущего поля (выбираем ключ массива, равный column поля)

### reformatFilledValue

Подготавливаем значение (выбираем нужное и трансформируем при необходимости)

Результат

```
value = "MoonShine"  
toValue = "MoonShine"  
toRawValue = "MoonShine"  
toFormattedValue = "1MoonShine"
```

## Вывод значения

### resolveValue

Трансформация значения перед выводом в формы

### value

Значение для формы, с учетом хелпера old

## Форма

Title

Role

- Метод **value** рекомендуется использовать только во **view**, так как при наличии значения **old**, вернется именно оно
- Во всех остальных случаях для получения значения формы необходимо использовать метод **toValue**

### resolvePreview

Трансформация значения перед выводом в таблицы (всюду кроме форм)

### preview

Вывод значения в таблицы

## Таблица и все что угодно кроме формы

TITLE	ROLE
1MoonShine	1

- **Preview** строится на основе **formattedValue**, но если оно не задано, то берется **toValue**

## Apply (Сохранение с учетом http запроса)

### requestValue

Получение значения из Http рекевста из формы на основе ключа column

## Данные из рекевста

```
[  
  "title" => 'MoonShine 2.0'  
];
```

### resolveOnApply

Получение данных и изменение их, с учетом данных из запроса (сохранение формы)

### resolveBeforeApply, resolveAfterApply, resolveAfterDestroy

Дополнительные события до сохранения, после и при удалении родительской сущности данных

## Apply

Входные данные трансформируются с учетом запроса и resolve методов

## Итоговые данные

```
[  
  "title" => "MoonShine 2.0",  
  "role_id" => 1  
];
```