

Docs » Data Handling Category » Data Processor Project » Custom Rules

Custom Rules

Creating Custom Rules

Anonymous · 09/09/2026

Creating Custom Rules

Custom Cast Rule

```
use Derafu\DataProcessor\Contract\CasterRuleInterface;

final class CustomCastRule implements CasterRuleInterface
{
    public function cast(mixed $value, array $parameters = []): mixed
    {
        // Your casting logic here.
    }
}

// Register the rule.
$registry->addCasterRule('custom_cast', CustomCastRule::class);
```

Custom Transform Rule

```
use Derafu\DataProcessor\Contract\TransformerRuleInterface;

final class CustomTransformRule implements TransformerRuleInterface
{
    public function transform(mixed $value, array $parameters = []): mixed
    {
        // Your transformation logic here.
    }
}

// Register the rule.
$registry->addTransformerRule('custom_transform', CustomTransformRule::class);
```

Custom Sanitizer Rule

```
use Derafu\DataProcessor\Contract\SanitizerRuleInterface;
```

```
final class CustomSanitizerRule implements SanitizerRuleInterface
{
    public function sanitize(mixed $value, array $parameters = []): mixed
    {
        // Your sanitization logic here.
    }
}

// Register the rule.
$registry->addSanitizerRule('custom_sanitize', CustomSanitizerRule::class);
```

Custom Validator Rule

```
use Derafu\DataProcessor\Contract\ValidatorRuleInterface;

final class CustomValidatorRule implements ValidatorRuleInterface
{
    public function validate(mixed $value, array $parameters = []): void
    {
        // Your validation logic here.
    }
}

// Register the rule.
$registry->addValidatorRule('custom_validate', CustomValidatorRule::class);
```

Using Custom Rules

Once registered, you can use your custom rules just like built-in ones:

```
$processor->process('field', $value, [
    'cast' => 'custom_cast',
    'transform' => ['custom_transform'],
    'sanitize' => ['custom_sanitize'],
    'validate' => ['custom_validate'],
]);
```

Last updated on 09/09/2026