

CSRF Protection

The CSRF token of the forms, and the errors of a form as a whole

Anonymous · 08/10/2026

CSRF Protection

A form is **protected with a CSRF token** unless it says otherwise. The form renders a hidden field `_token` with a token, and when its data is processed the token that comes back is checked. `derafu/form` does not know where tokens are kept, nor what a session or a request is: that is what a **CSRF token manager** knows, and the application gives one.

The token manager

```
namespace Derafu\Form\Contract\Csrf;

interface CsrfTokenManagerInterface
{
    // The token for an id (the form it is for).
    public function getToken(string $id): string;

    // Whether a token that came with a form is the one of the id for this user.
    public function isValid(string $id, string $token): bool;
}
```

The form asks for a token every time it is shown, and checks the one that comes with the data. How long a token lasts, and whether checking it uses it up, is up to each implementation.

The id is the **name of the schema** of the form, or `form` if it has none, so a token of one form is not good for another.

An implementation is specific to each environment, because it is the one that knows which session the tokens belong to (for example, one that wraps the token manager of Symfony, or one that works over the session of a Mezzio request). The package defines the contract and does not bring any.

Wiring

With the services of the package (`form-services.yaml`) the manager is **optional**: if the application has a service that implements `CsrfTokenManagerInterface`, the renderer and the processor use it. When the renderer is created by hand, it is given in the options:

```
$renderer = FormRendererFactory::create(['csrf_token_manager' => $manager]);

$processor = new FormDataProcessor($resolver, $dataProcessor, csrfTokenManager:
$manager);
```

A form that is protected needs a manager

Rendering or processing a form that is protected, **without a manager, is an error** (an exception with the name of the form), not a form that is silently left open. Either register a manager or turn the protection off for the form.

Turning it off

It is an option of the form, which both the renderer and the processor read, so they always agree:

```
# resources/forms/search.form.yaml
options:
  csrf_protection: false
```

A form that is not protected renders no token, and its data is processed without one. Use it for forms that do not need it, for example a search form with the method GET, or a form that is posted by clients that are not browsers.

When the token is not valid

The token is **not part of the data**: it is taken out before the fields are processed. If it is missing, or it is not valid, the result is not valid and it has a message in the **errors of the form**:

```
$result = $processor->process($form, $_POST);

$result->isValid();           // false
$result->getFormErrors();    // ['The form is not valid or has expired. ...']
```

The fields are processed all the same, so their own errors come along. The message is translated if the processor has a translator (the text is in the domain `errors`).

The errors of a form as a whole

Some errors do not belong to any field, like this one. They are in `ProcessResult::getFormErrors()`, they count in `hasErrors()` and they come first in `getAllErrors()`. The form that the result gives (`getForm()`) has them (`$form->getErrors()`), and the template of the form shows them **before its**

fields with `form_global_errors(form)`:

```
{{ form_global_errors(form) }}
```

The errors of the fields are still shown in their rows, as before.

Last updated on 08/10/2026