

Csrp Project

CSRF token manager for the forms of derafu/form, working over the session of Mezzio with one token for each form

Anonymous · 10/10/2026 · #php

Derafu: Csrp

[github](#) [derafu/csrp](#)[last commit](#) [last wednesday](#) CI [passing](#)[downloads](#) [97](#)

The CSRF token manager that [derafu/form](#) asks for, over the session of [Mezzio](#). [derafu/form](#) renders the token in its forms and checks it when their data is processed; this package gives the tokens and keeps them in the session of the user. It does not depend on [derafu/http](#) nor on [derafu/auth](#), so a site with a contact form and no login uses it too.

An application that uses Symfony does not need it: it gives [derafu/form](#) a manager over its own.

What it gives

- [SessionCsrpTokenManager](#), the implementation of [Derafu\Form\Contract\Csrp\CsrpTokenManagerInterface](#).
- [CsrpSessionMiddleware](#), that gives the session of the request to the manager while the request is handled.
- [csrpf-services.yaml](#), with both, and the translations of the package (Spanish ships with it).

Install

```
composer require derafu/csrp
```

The session itself comes from [derafu/session](#) (a site that uses [derafu/foundation](#) has both). Import the services and add the middleware **after** the session middleware:

```
imports:
    - { resource: '../vendor/derafu/csrp/resources/config/csrpf-services.yaml' }

services:
    Psr\Http\Server\RequestHandlerInterface:
        class: Derafu\Http\Service\RequestHandler
        arguments:
            $middlewares:
```

```
- '@Derafu\Http\Middleware\RouterMiddleware'  
- '@Mezzio\Session\SessionMiddleware'  
- '@Derafu\Csrf\CsrfSessionMiddleware'  
# ...  
- '@Derafu\Http\Middleware\DispatcherMiddleware'
```

The services register the manager under the interface of `derafu/form`, so the renderer and the processor of the forms use it without anything else. Forms are protected unless they turn it off (`csrf_protection: false`), and a form that is protected can not be rendered nor processed without a manager (see [CSRF Protection](#)).

How the tokens are kept

Each id (the name of the schema of the form, or `form`) has a **secret** in the session of the user, in the key `derafu.csrf`. It is created the first time that a token is asked for, and it lasts as long as the session.

- **The token is not the secret.** It is a masked copy: random bytes followed by the secret combined with them, in base64 for URLs. It is different in every response, so the token that travels in the HTML can not be guessed by comparing compressed responses (BREACH). Checking it takes the mask out and compares in constant time.
- **Checking does not use it up.** Two tabs of the same form, the back button, or a form that is shown again after an error keep working.
- **A token is for one id and one session.** The token of another form, or of another visitor, is not valid.
- **What is not a token is not valid:** a token that is empty, that is not base64, that is shorter or longer than it must be, or that was changed, is rejected without an error.

A visitor without session

The session is written the first time that a token is asked for, and that creates it: a visitor that opens a form for the first time gets a session (and its cookie) with the page. A page whose forms are not protected does not create one.

Limits

- The manager needs the session of the request. Without `CsrfSessionMiddleware`, or in a request that has no session, asking for or checking a token is an error with the message that says what to add.
- The secrets are not renewed when the user logs in: the session id is renewed (see [Derafu Auth](#)), but the data of the session, the secrets included, stays.

Last updated on 10/10/2026