{{ title }} (403)

{{ main }}

{% if no\_referer %}

{{ no\_referer1 }}

{{ no\_referer2 }}

{{ no\_referer3 }}

{% endif %} {% if no\_cookie %}

{{ no\_cookie1 }}

{{ no\_cookie2 }}

{% endif %}

{% if DEBUG %}

## Help

{% if reason %}

Reason given for failure:

{{ reason }}

{% endif %}

In general, this can occur when there is a genuine Cross Site Request Forgery, or when [Django’s CSRF mechanism](https://docs.djangoproject.com/en/%7B%7B%20docs_version%20%7D%7D/ref/csrf/) has not been used correctly. For POST forms, you need to ensure:

* Your browser is accepting cookies.
* The view function passes a request to the template’s [render](https://docs.djangoproject.com/en/%7B%7B%20docs_version%20%7D%7D/topics/templates/#django.template.backends.base.Template.render) method.
* In the template, there is a {% templatetag openblock %} csrf\_token {% templatetag closeblock %} template tag inside each POST form that targets an internal URL.
* If you are not using CsrfViewMiddleware, then you must use csrf\_protect on any views that use the csrf\_token template tag, as well as those that accept the POST data.
* The form has a valid CSRF token. After logging in in another browser tab or hitting the back button after a login, you may need to reload the page with the form, because the token is rotated after a login.

You’re seeing the help section of this page because you have DEBUG = True in your Django settings file. Change that to False, and only the initial error message will be displayed.

You can customize this page using the CSRF\_FAILURE\_VIEW setting.

{% else %}

{{ more }}

{% endif %}