



리액트와 함께 장고 시작하기 / 장고 Forms

Cross Site Request Forgery

여러분의 파이썬/장고 페이스메이커가 되겠습니다.

사이트 간 요청 위조 공격

Cross Site Request Forgery

사용자가 의도하지 않게 게시판에 글을 작성하거나, 쇼핑을 하게 하는 등의 공격

특정 웹사이트가 유저의 웹 브라우저를 신용하는 상태를 노린 것

<!-- 공격자 사이트의 웹페이지에 접속하면, 그 즉시 site-victim.com로의 POST 요청이 사용자 모르게 전달됩니다. -->

```
<body onload="document.attack_form.submit();">
  <form name="attack_form" method="post" action="http://site-victim.com/new/">
    <input type="hidden" name="title" value="스팸 제목" />
    <input type="hidden" name="content" value="스팸 내용" />
  </form>
</body>
```

```

```

<https://docs.djangoproject.com/en/3.0/ref/csrf>

요청을 받는 서버 입장에서,

공격을 막기 위해 Token을 통한 체크

POST 요청에 한해 CsrfViewMiddleware를 통한 체크

POST 요청을 받을 때 Token값이 없거나 유효하지 않다면, 403 Forbidden 응답

처리 순서

1) 입력 Form을 보여줄 때, CSRF Token값도 값이 할당

CSRF Token은 User마다 다르며, 계속 변경됩니다.

2) 그 입력 Form을 통해 Token값이 전달이 되면, Token 유효성 검증

이를 적용하기 위한 단 1줄의 코드

장고 프로젝트 기본 세팅으로 CsrfViewMiddleware가 적용되어있습니다.

```
<form action="" method="post">  
    {% csrf_token %}  
    <input type="text" name="title" />  
    <textarea name="content"></textarea>  
    <input type="submit" />  
</form>
```

```
<input type="hidden" name="csrfmiddlewaretoken" value="....." />
```

주의사항

CSRF Token != 유저인증 Token

CSRF Token != JWT (JSON Web Token)

CSRF Token 체크 기능을 끈다 ???

가급적이면 끄지 마세요.

기본 제공되는 보안기능이며,

이를 유지하는 데에 비용이 거의 들지 않는다.

특정 View에 한해, CSRF Token 체크에서 배제하려면?

해당 뷰에 @csrf_exempt 장식자를 적용

```
from django.views.decorators.csrf import csrf_exempt
```

```
@csrf_exempt
```

```
def post_new_for_api(request):  
    # blah blah blah ...
```

앱 API에서는 꺼주세요.

django-rest-framework의 APIView에서는 csrf_exempt가 적용되어있습니다.

```
class APIView(View):
    # ...
    @classmethod
    def as_view(cls, **initkwargs):
        # ...
        view = super().as_view(**initkwargs)
        view.cls = cls
        view.initkwargs = initkwargs

        # Note: session based authentication is explicitly CSRF validated,
        # all other authentication is CSRF exempt.
        return csrf_exempt(view)
```

Life is short.
You need Python and Django.

I will be your pacemaker.

