laravel XSS prevention

Jump to bottom

Dharmvijay Patel edited this page 3 minutes ago · 1 revision

- instead of using {{}} The {{{ }}} escape the string. this clear output, what about saving data in database, clearing sql injection codes, javascript code from form feilds XSS is solved by escaping the output ({{{ $var }}}), so the html/js doesn't get executed.
- Create a middeleware to filter out tags from request.

```
namespace App\Http\Middleware;
use Closure;
use Illuminate\Http\Request;
class XSSMiddleware
{
    public function handle(Request $request, Closure $next)
    {
        $input = $request->all();
        array_walk_recursive($input, function(&$input) {
            $input = strip_tags($input);
        });
        $request->merge($input);
        return $next($request);
    }
}
```

- Laravel query builder uses PDO paramater binding, you don't have to worry about SQL injection. Of course If you use raw queries, you have to deal with them by yourself.

// You are safe $results = DB::select("select * from users where id = ?", array(Input::get('id')));

// You are NOT safe $results = DB::select(DB::raw("select * from users where id =".Input::get('id'))); XSS is more complex than SQL injection. You need a third party library. I suggest HTML Purifier

Clone this wiki locally

```
https://github.com/dharmvijay/Laravel.wiki.git
```