

# Livewiring with a Light Touch

**Nick Evans** 

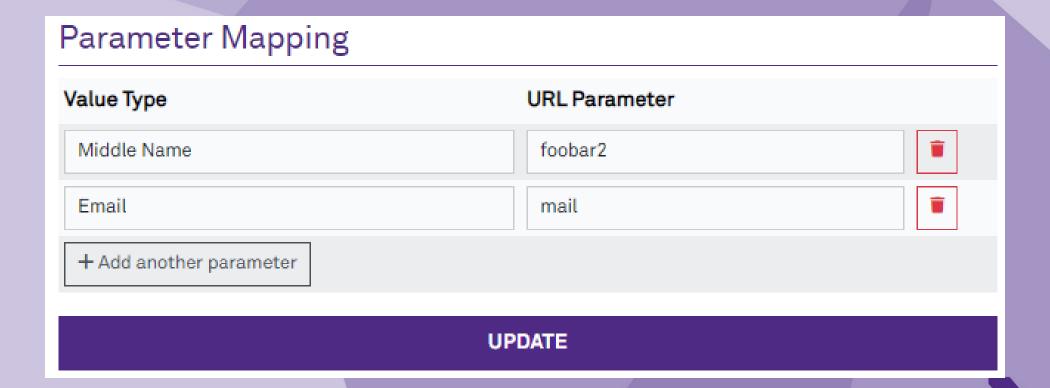
## Objective

- A little webdev history
- Showcase some different ways to use Livewire
  - ... and then my favorite way to use Livewire



#### **Basic Forms**

```
<form method="POST" action="/some-url">
   <fieldset id="mapping-1">
       <legend>Field #1</legend>
       <label for="origin-1">Origin Field</label>
        <input type="text" name="origin-1" id="mapping-1">
       <label for="dest-1">Destination Field</label>
        <input type="text" name="dest-1" id="dest-1">
   </fieldset>
   <button type="button" id="add">Add Another Field/button>
   <button type="submit">Save</button>
</form>
```





## Olde Schoole Way: jQuery

```
// <button type="button" id="add">Add Another Field</button>
$(document).ready(() => {
   $('add').click(() => {
       const numFieldSet = $('form fieldset').length;
       $('form').append(`<fieldset>
            <legend>Field #${numFieldSet + 1}</legend>
           . . . more tags . . .
   });
```



#### **Problems**

- Spreading your logic
  - Backend views
  - Front-end's DOM manipulations
- HTML in a big JS string typically costs you:
  - Syntax highlighting
  - Linting/analysis
  - Typehinting (my IDE even typehints CSS classes!)



## Livewiring the Form

```
<form method="POST" action="/some-url">
    @foreach ($mappings as $i => $mappings)
    <fieldset id="mapping-{{ $i }}">
        <legend>Field #{{ $i }}</legend>
        <label for="origin-{{ $i }}">Origin Field</label>
        <input type="text" name="..." id="..." wire:model="mappings.{{ $i }}.origin">
        <label for="dest-{{ $loop->index }}">Destination Field</label>
        <input type="text" name="..." id="..." wire:model="mappings.{{ $i }}.dest">
    </fieldset>
    @endforeach
    <button type="button" wire:click="add">Add Another Field</button>
    <button type="submit">Save</button>
</form>
```



```
class MappingForm extends Component
    public array $mappings = [];
    public function render(): View
       if (count($this->mappings) === 0) {
            $this->add(origin: '', dest: '');
        // Our HTML from before!
        return view('mapping-form');
    public function add(string $origin, string $dest): void
       $this->mappings[] = [
            'origin' => $origin,
            'dest' => $dest,
```

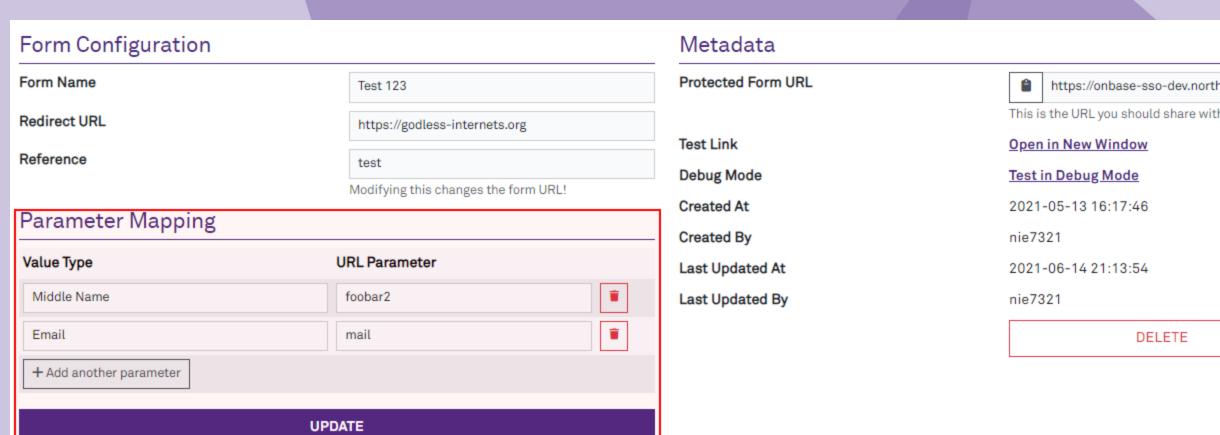


```
class MappingForm extends Component
   public ParentModel $model;
   protected $rules = [
        'mappings.*' => 'required|array',
        'mappings.origin' => 'required|string',
        'mappings.dest' => 'required|string',
   ];
   // render() & add() omitted for brevity, but they're still here!
   public function save(): Redirect
       $this->validate();
        $this->model->mappings()->each->save();
        request()->session()->flash('status', 'Mappings saved, woo!');
        return redirect('/homepage');
```



#### **Analysis of "Heavy Touch"**

• Consider: this isn't the only part of the form.





## Nick's (typical) Approach

- 1. Build page
- 2. Add Livewire spice to the already-working page
- 3. ???
- 4. Profit!



## Counterpoint

- Everything is situational
- Some times, you DO want Livewire to take total control of a page
- Gotta think it through





## Thank you!

Questions?

Scan the QR code for the deck source code!

