

# Checkout Form Custom Validator Rule



# White Space

- Our form currently has a problem with white space
- In the required fields
  - If you enter ONLY whitespace ... it **passes** ... YIKES!
  - Should **fail!**

# Custom Validator

- We can resolve this with a custom validator: **notOnlyWhitespace**
  - Check the field value
  - If it ONLY has whitespace
    - then return an error (failed)
    - else return null (passed)

# Development Process

*Step-By-Step*

1. Define custom validator rule
2. Specify custom validator rule for the form controls
3. Update HTML template to display error messages

# Step 1: Define custom validator rule

Don't allow only whitespace

File: luv2-shopValidators.ts

```
import { FormControl, ValidationErrors } from '@angular/forms';
export class Luv2ShopValidators {

    // whitespace validation
    static notOnlyWhitespace(control: FormControl) : ValidationErrors {
        // check if string only contains whitespace
        if ((control.value != null) && (control.value.trim().length === 0)) {

            // invalid, return error object
            return { 'notOnlyWhitespace': true };
        }
        else {
            // valid, return null
            return null;
        }
    }
}
```

If validation check fails  
then return validation error(s)

If validation check passes  
then return null

Validation error key

The HTML template will check for this error key

# Step 2: Specify custom validator rule for form controls

File: checkout.component.ts

```
this.checkoutFormGroup = this.formBuilder.group({  
  customer: this.formBuilder.group({  
    'firstName': new FormControl('',  
      [Validators.required, Validators.minLength(2), Luv2ShopValidators.notOnlyWhitespace]  
    ),  
    ...  
  }),  
  ...  
});
```

Our custom validator  
method name

```
export class Luv2ShopValidators {  
  // whitespace validation  
  static notOnlyWhitespace(control: FormControl) : ValidationErrors {  
    // check if string only contains whitespace  
    if ((control.value != null) && (control.value.trim().length === 0)) {
```

# Step 3: Update HTML template to display error messages

File: checkout.component.html

```
<label>First Name</label>
<input formControlName="firstName" type="text">

<div *ngIf="firstName.invalid && (firstName.dirty || firstName.touched)" class="alert alert-danger">

  <div *ngIf="firstName.errors.required || firstName.errors.notOnlyWhitespace">
    First Name is required
  </div>

  <div *ngIf="firstName.errors.minLength">
    First Name must be at least 2 characters long
  </div>

</div>
```

If validation fails  
then display error messages

Reference to our custom  
validation error key

# Step 3: Update HTML template to display error messages

File: checkout.component.html

```
<label>First Name</label>
<input formControlName="firstName" type="text">

<div *ngIf="firstName.invalid && (firstName.dirty || firstName.touched)" class="alert alert-danger">

  <div *ngIf="firstName.errors.required || firstName.errors.notOnlyWhitespace">
    First Name is required
  </div>

  <div *ngIf="firstName.errors.minLength">
    // whitespace validation
    static notOnlyWhitespace(control: FormControl) : ValidationErrors {
      // check if string only contains whitespace
      if ((control.value != null) && (control.value.trim().length === 0)) {
        // invalid, return error object
        return { 'notOnlyWhitespace': true };
      } else {
        // valid, return null
      }
    }
  </div>
</div>
```

If validation fails  
then display error messages

Reference to our custom  
validation error key