Form handling and validation

✅ Form handling is part of vue

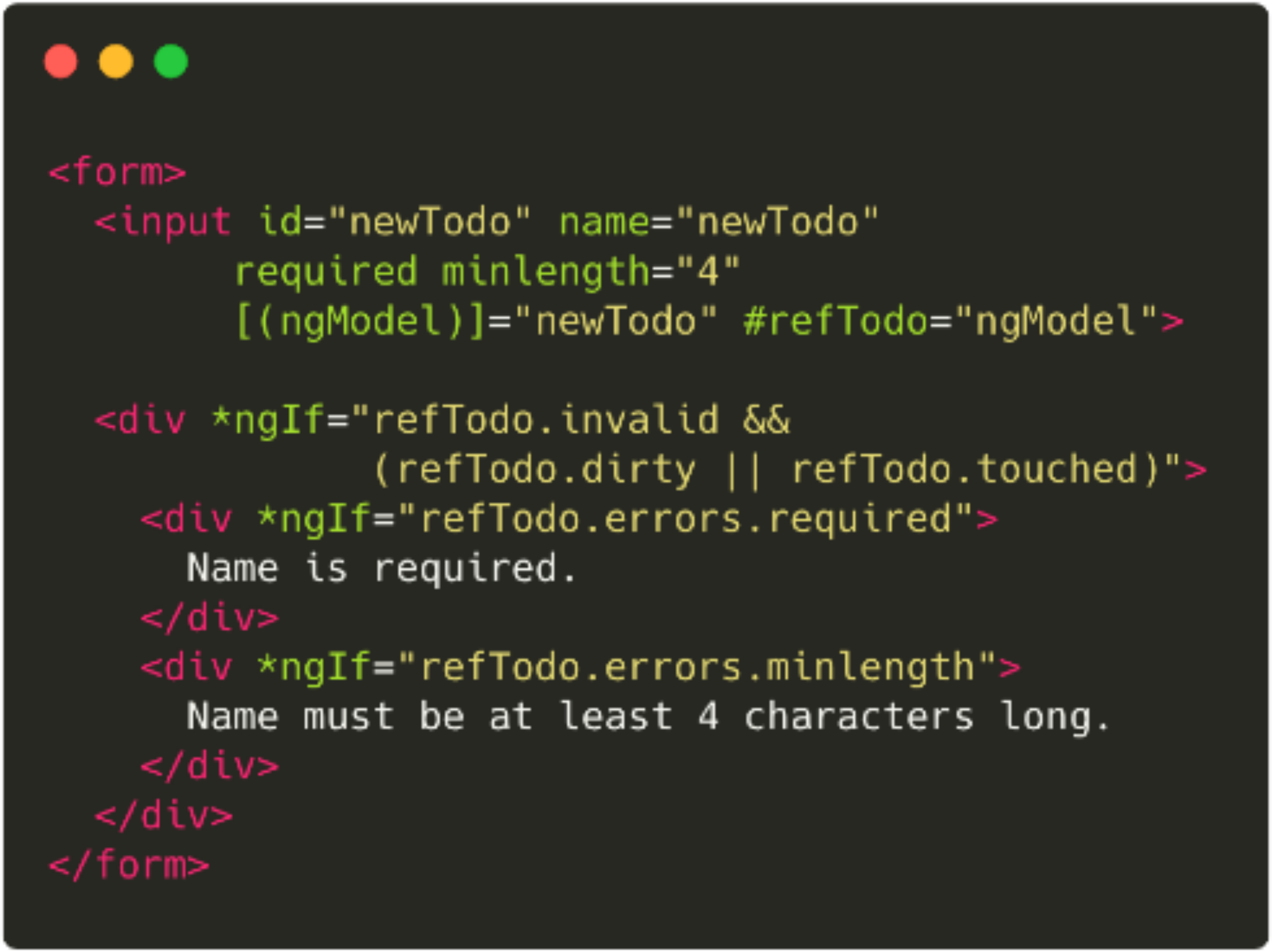
❌ Form validation is part of vue

📦 vue or vee-validate

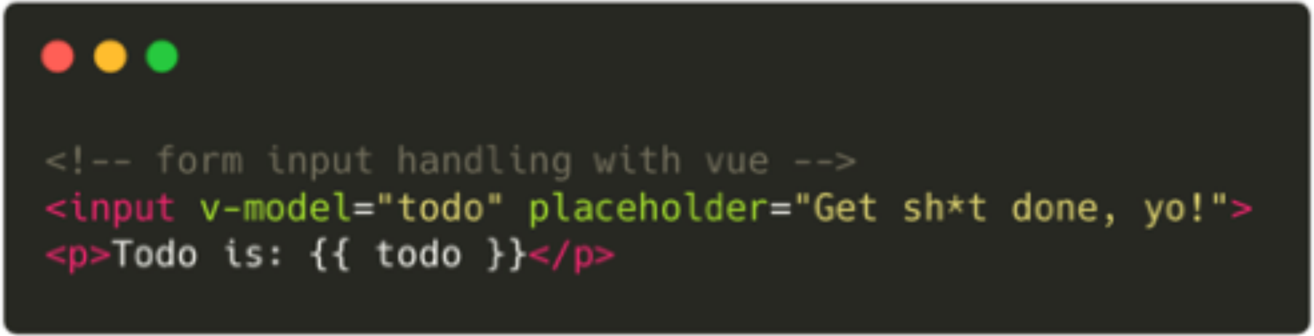
📋 https://github.com/vuejs/awesome-vue#validation

✅ Part of angular

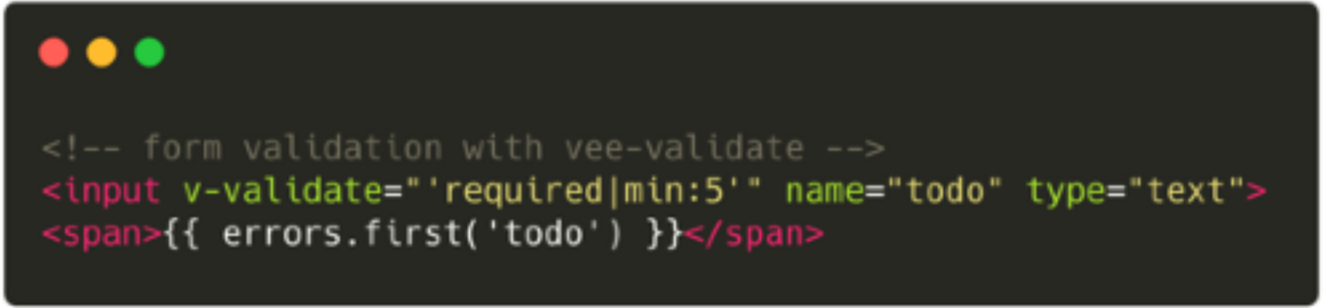
📦 @angular/core or @ngrx/platform



```
<form>
  <input id="newTodo" name="newTodo"
        required minlength="4"
        [(ngModel)]="newTodo" #refTodo="ngModel">

  <div *ngIf="refTodo.invalid &&
              (refTodo.dirty || refTodo.touched)">
    <div *ngIf="refTodo.errors.required">
      Name is required.
    </div>
    <div *ngIf="refTodo.errors.minlength">
      Name must be at least 4 characters long.
    </div>
  </div>
</form>
```

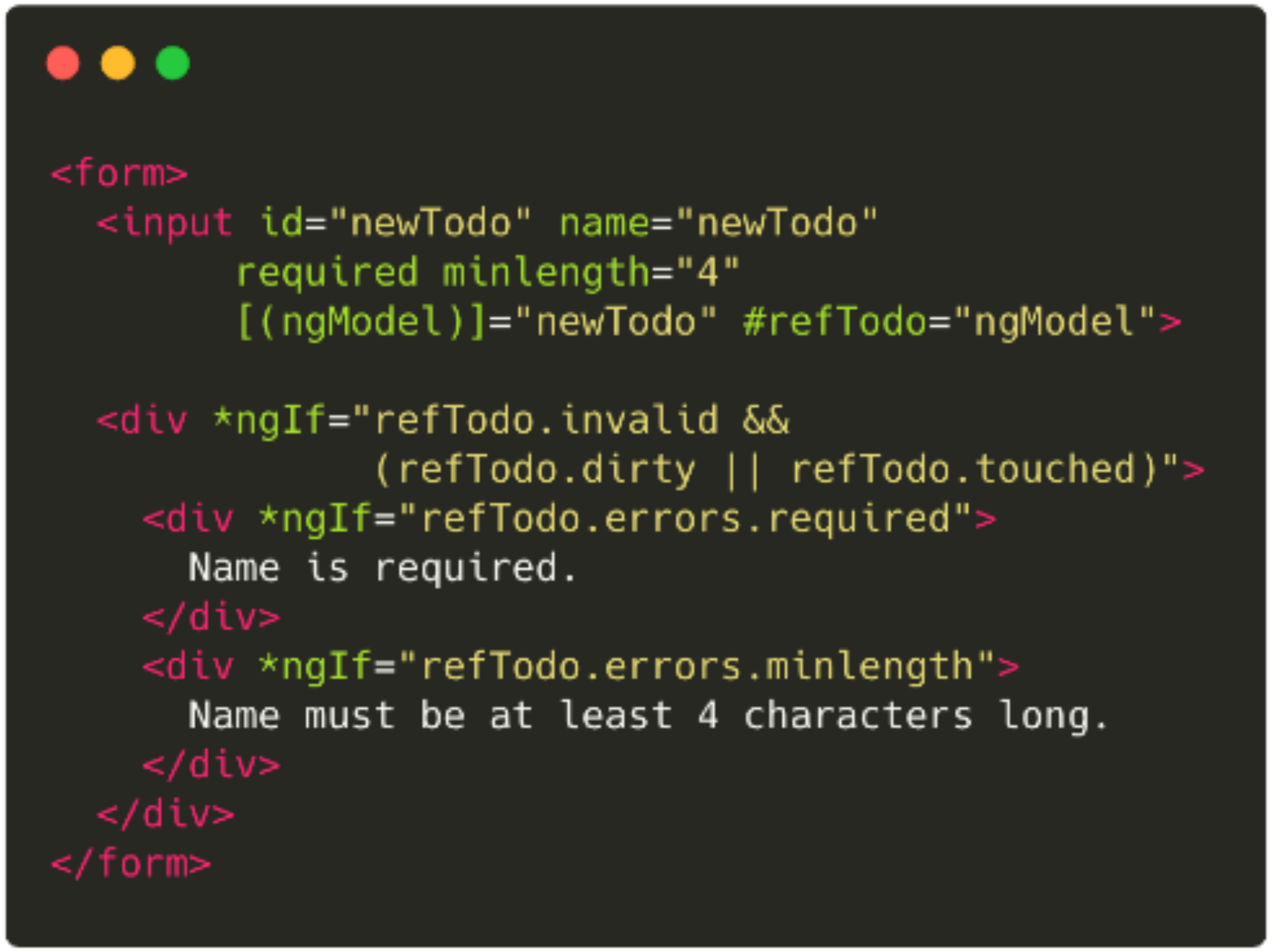


```
<!-- form input handling with vue -->
<input v-model="todo" placeholder="Get sh*t done, yo!">
<p>Todo is: {{ todo }}</p>
```



```
<!-- form validation with vee-validate -->
<input v-validate="'required|min:5'" name="todo" type="text">
<span>{{ errors.first('todo') }}</span>
```

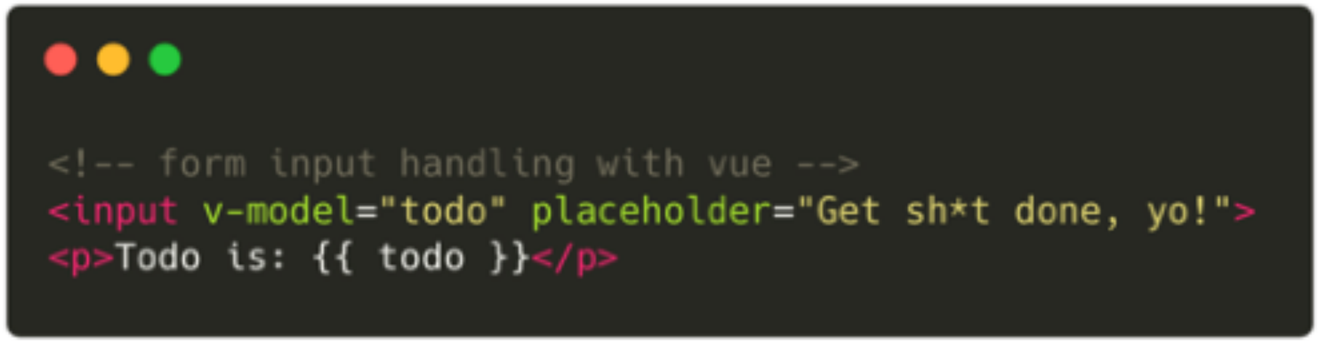
Form handling and validation



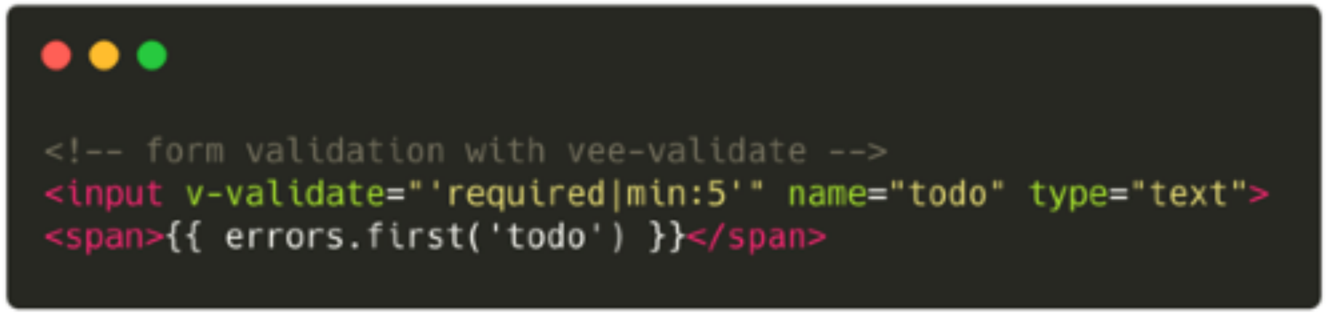
```html
<form>
  <input id="newTodo" name="newTodo"
        required minlength="4"
        [(ngModel)]="newTodo" #refTodo="ngModel">

  <div *ngIf="refTodo.invalid &&
              (refTodo.dirty || refTodo.touched)">
    <div *ngIf="refTodo.errors.required">
      Name is required.
    </div>
    <div *ngIf="refTodo.errors.minlength">
      Name must be at least 4 characters long.
    </div>
  </div>
</form>
```

✅ Part of angular

📦 @angular/core or @ngrx/platform



```html
<!-- form input handling with vue -->
<input v-model="todo" placeholder="Get sh*t done, yo!">
<p>Todo is: {{ todo }}</p>
```



```html
<!-- form validation with vee-validate -->
<input v-validate="'required|min:5'" name="todo" type="text">
<span>{{ errors.first('todo') }}</span>
```

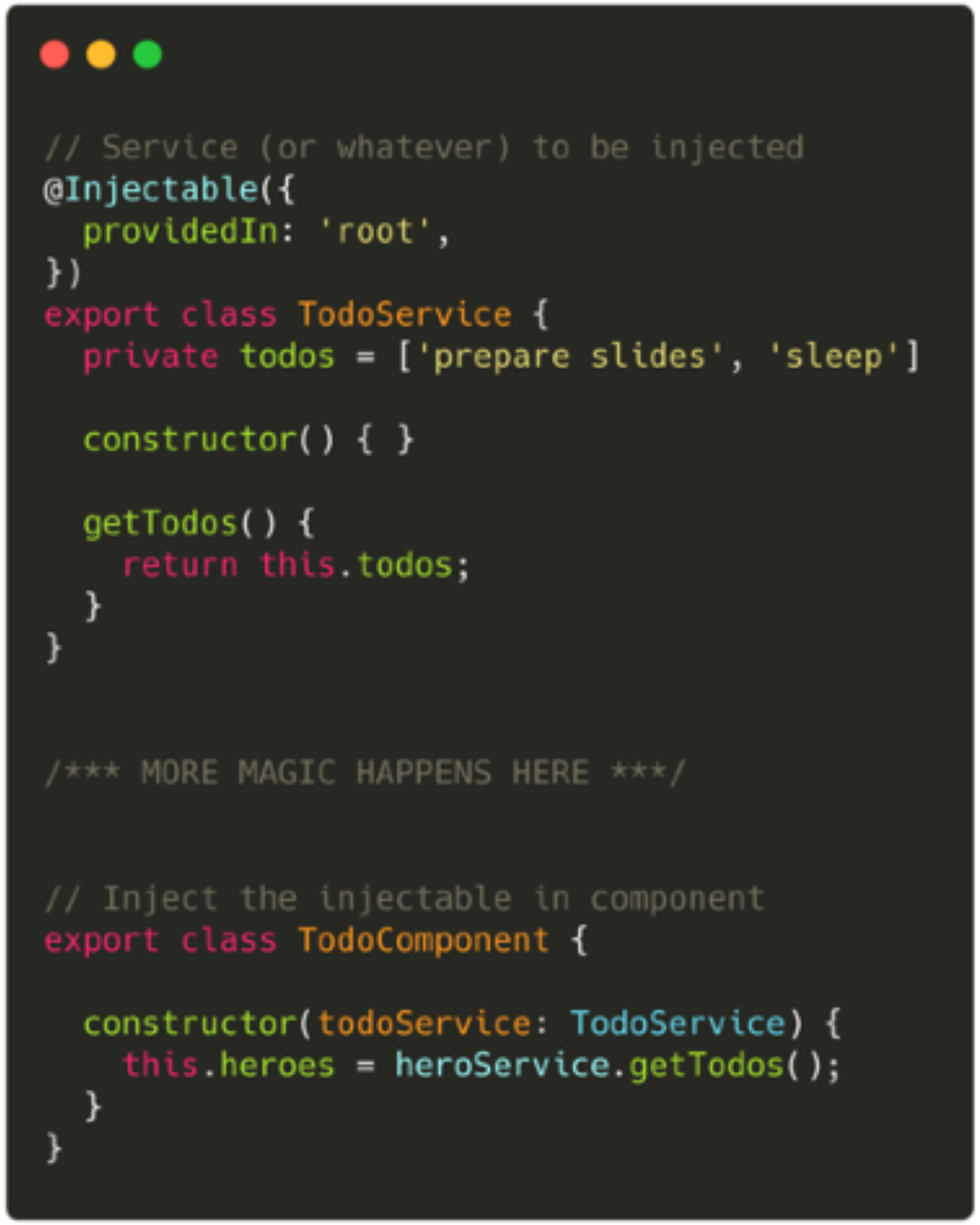
✅ Form handling is part of vue

❌ Form validation is part of vue

📦 vue or vee-validate

📋 https://github.com/vuejs/awesome-vue#validation

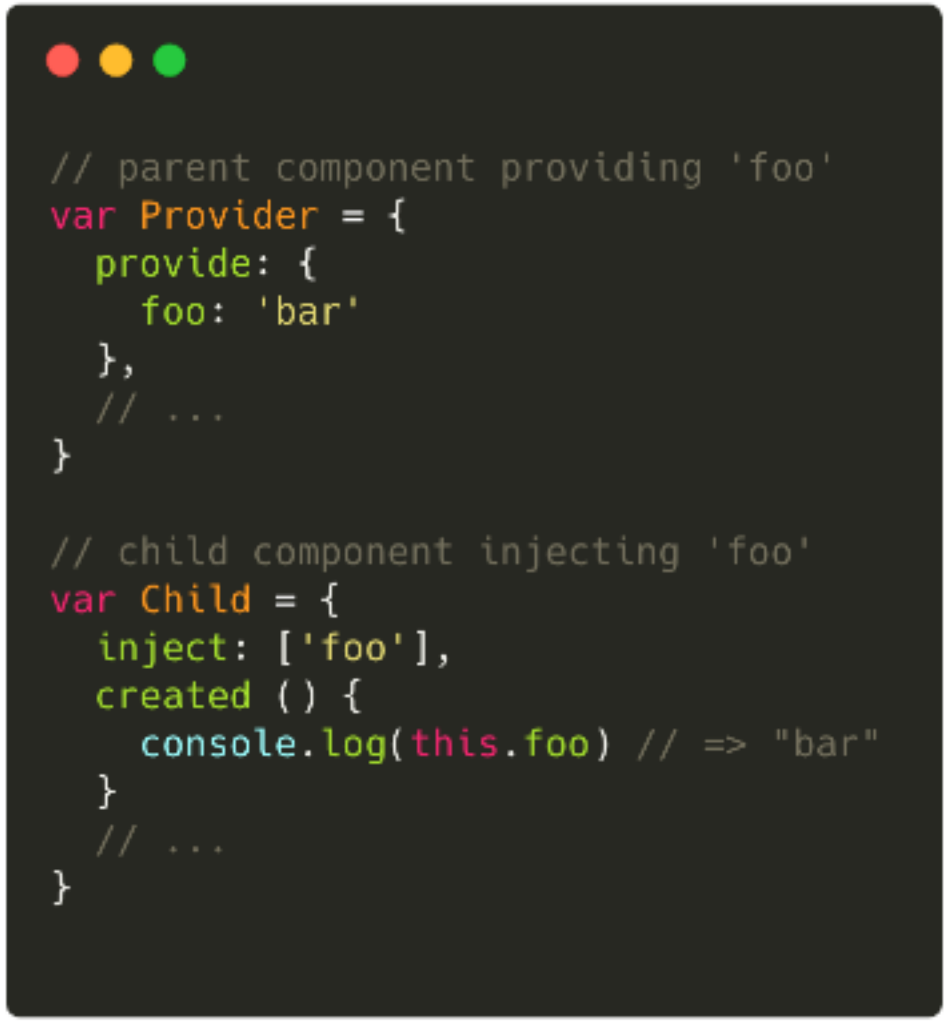
Dependency Injection



```
// Service (or whatever) to be injected
@Injectable({
  providedIn: 'root',
})
export class TodoService {
  private todos = ['prepare slides', 'sleep']

  constructor() { }

  getTodos() {
    return this.todos;
  }
}


/*** MORE MAGIC HAPPENS HERE ***/


// Inject the injectable in component
export class TodoComponent {

  constructor(todoService: TodoService) {
    this.heroes = heroService.getTodos();
  }
}
```

✅ Part of angular

📦 @angular/core



```
// parent component providing 'foo'
var Provider = {
  provide: {
    foo: 'bar'
  },
  // ...
}

// child component injecting 'foo'
var Child = {
  inject: ['foo'],
  created () {
    console.log(this.foo) // => "bar"
  }
  // ...
}
```

✅ Part of vue

📦 vue