

Working with Validation and Logging



Kevin Dockx

ARCHITECT

@KevinDockx <https://www.kevindockx.com>



Coming Up



Working with validation in a RESTful world

Validating on create & update

Logging faults, errors and additional information



Working with Validation in a RESTful World



Define validation rules

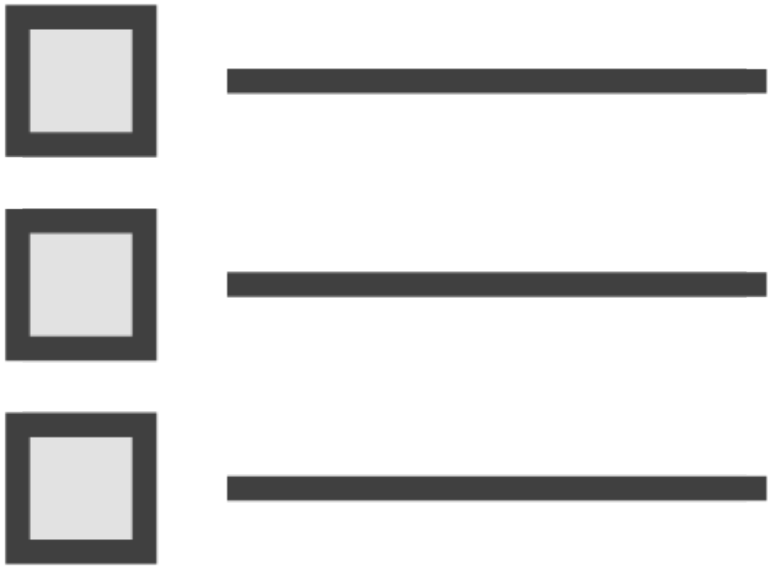


Check validation rules



Report validation
errors

Defining Validation Rules

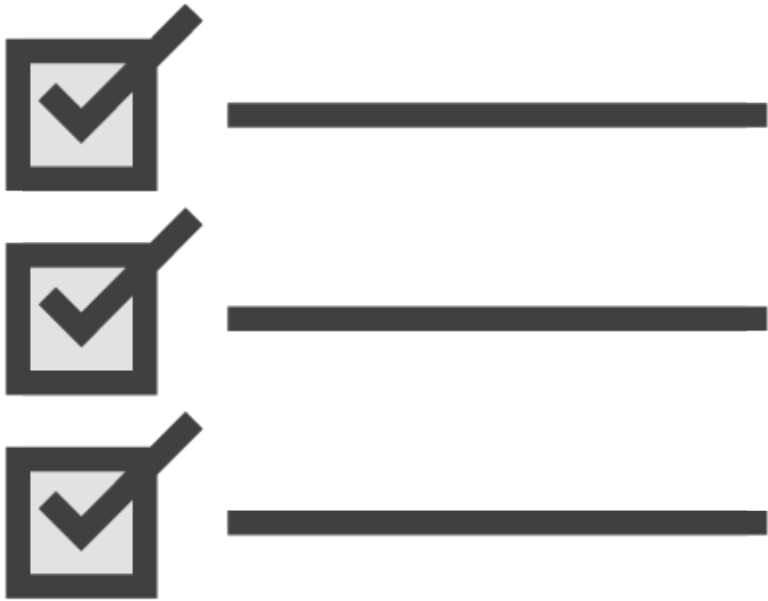


In ASP.NET Core, rules are by default defined by using

- Data annotations
- Custom code for when the annotations aren't sufficient

Validate input, not output

Checking Validation Rules



ModelState

- A dictionary containing both the state of the model, and model binding validation
- Contains a collection of error messages for each property value submitted

ModelState.IsValid()



Reporting Validation Errors



**Response status should be 422 -
Unprocessable entity**

**Response body should contain validation
errors**

```
{  
  ...  
  "Title": ["You should fill out a title."],  
  "Description": ["The description shouldn't have more  
                  than 500 characters."]  
}
```

Reporting Validation Errors

Properties & their related error message(s)

Often with an additional, custom error code



Demo



Working with Validation on POST



Demo



Working with Validation on PUT



Demo



Working with Validation on PATCH



Demo



Logging Faults



Demo



Logging Errors and Other Information



Demo



Logging to a File



Summary



Validation

- Defining validation rules
- Checking validation rules
- Reporting validation errors

Log faults, errors or additional information by using the built-in logger

Extend the logger with third-party middleware