Spring Framework 7

Beginner to Guru

Exception Handling Overview

HTTP Status Codes

- **100** Series - Informational Responses
- **200** Series - Successful Responses
- **300** Series - Redirection Responses
- **400** Series - Client Error Responses
- **500** Series - Server Error Responses

Use of HTTP Status Codes

- **200** Series are used when the request completes as expected
  - Common codes used: 200 Ok, 201 Created, 202 Accepted, 204 No Content
- **400** Series are used when the request is understood but fails an expected condition
  - **400 Bad Request** - Invalid data received
  - **401 Unauthorized** - User authentication required
  - **403 Forbidden** - User authenticated, not authorized
  - **404 Not Found** - Requested Resource Not Found
  - **405 Method Not Allowed** - Method Not Supported

Use of HTTP Status Codes

- **500** Series are used when there is a server side error
    - Typically from unexpected runtime errors
    - Null Pointer Errors, parse errors, database connection errors, etc
    - Code should log relevant information
    - Do not return stack trace to client
        - Be careful to not “leak” information to internet

Standard Spring MVC Exceptions

- Spring MVC does support a number of standard exceptions
- Standard Exceptions are handled by the DefaultHandlerExceptionResolver
- The DefaultHandlerExceptionResolver sets the appropriate HTTP status code
  - BUT does not write content to the body of the response
- Spring MVC does have robust support for customizing error responses

Spring Standard Exceptions

- **BindException** - 400 Bad Request
- **ConversionNotSupportedException** - 500 Internal Server Error
- **HttpMediaTypeNotAcceptableException** - 406 Not Acceptable
- **HttpMediaTypeNotSupportedException** - 415 Unsupported Media Type
- **HttpMessageNotReadableException** - 400 Bad Request
- **HttpMessageNotWritableException** - 500 Internal Server Error
- **HttpRequestMethodNotSupportedException** - 405 Method Not Allowed
- **MethodArgumentNotValidException** - 400 Bad Request

Spring Standard Exceptions

- **MissingServletRequestParameterException** - 400 Bad Request
- **MissingServletRequestPartException** - 400 Bad Request
- **NoSuchRequestHandlingMethodException** - 404 Not Found
- **TypeMismatchException** - 400 Bad Request

Spring MVC Exception Handling

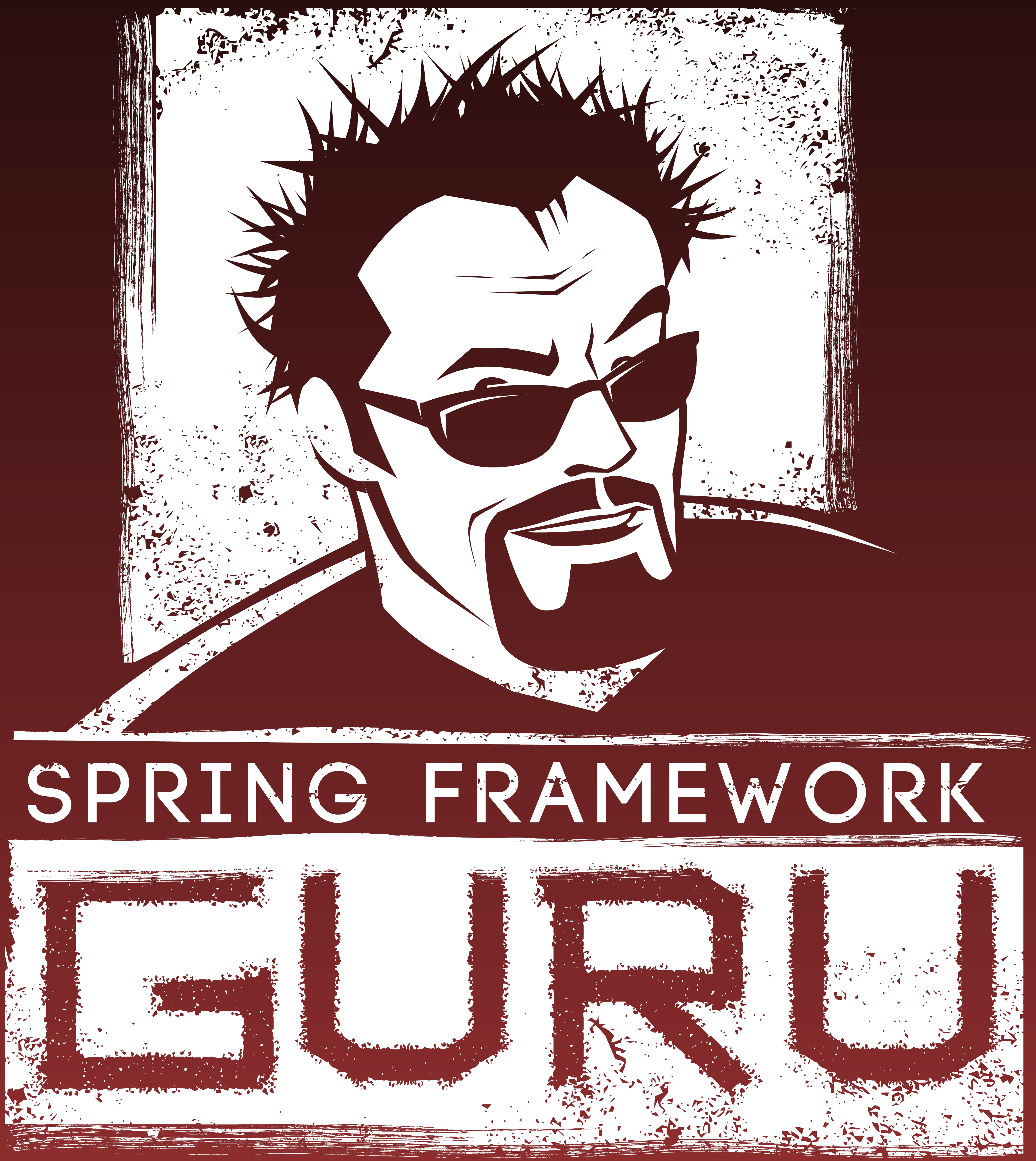
- @ExceptionHandler on controller method to handle specific Exception types
- @ReponseStatus - annotation for custom exception classes to set desired HTTP status
- Implement AbstractHandlerException Resolver - full control over response (including body)
- @ControllerAdvice - used to implement a global exception handler
- ResponseStatusException.class - (Spring 5+) Exception which can be thrown which allows setting the HTTP status and message in the constructor

Spring Boot ErrorController

- Provides Whitelabel Error Page for HTML requests, or JSON response for RESTful requests
- Properties:
  - **`server.error.include-binding-errors`** - default: never
  - **`server.error.include-exception`** - default: false
  - **`server.error.include-message`** - default: never
  - **`server.error.include-stacktrace`** - default: never
  - **`server.error.path`** - default: /error
  - **`server.error.whitelabel.enabled`** - default: true

Spring Boot BasicErrorController

- Spring Boot includes a BasicError Controller
- This class can be extended for additional error response customization
- Allows for wide support of the needs of various clients and situations
- Rarely used, but important to know it is available for use when needed


SPRING FRAMEWORK
GURU