Annotation used for Spring Boot Error Handling

- **@RestController :** is the base annotation for classes that handle REST operations.

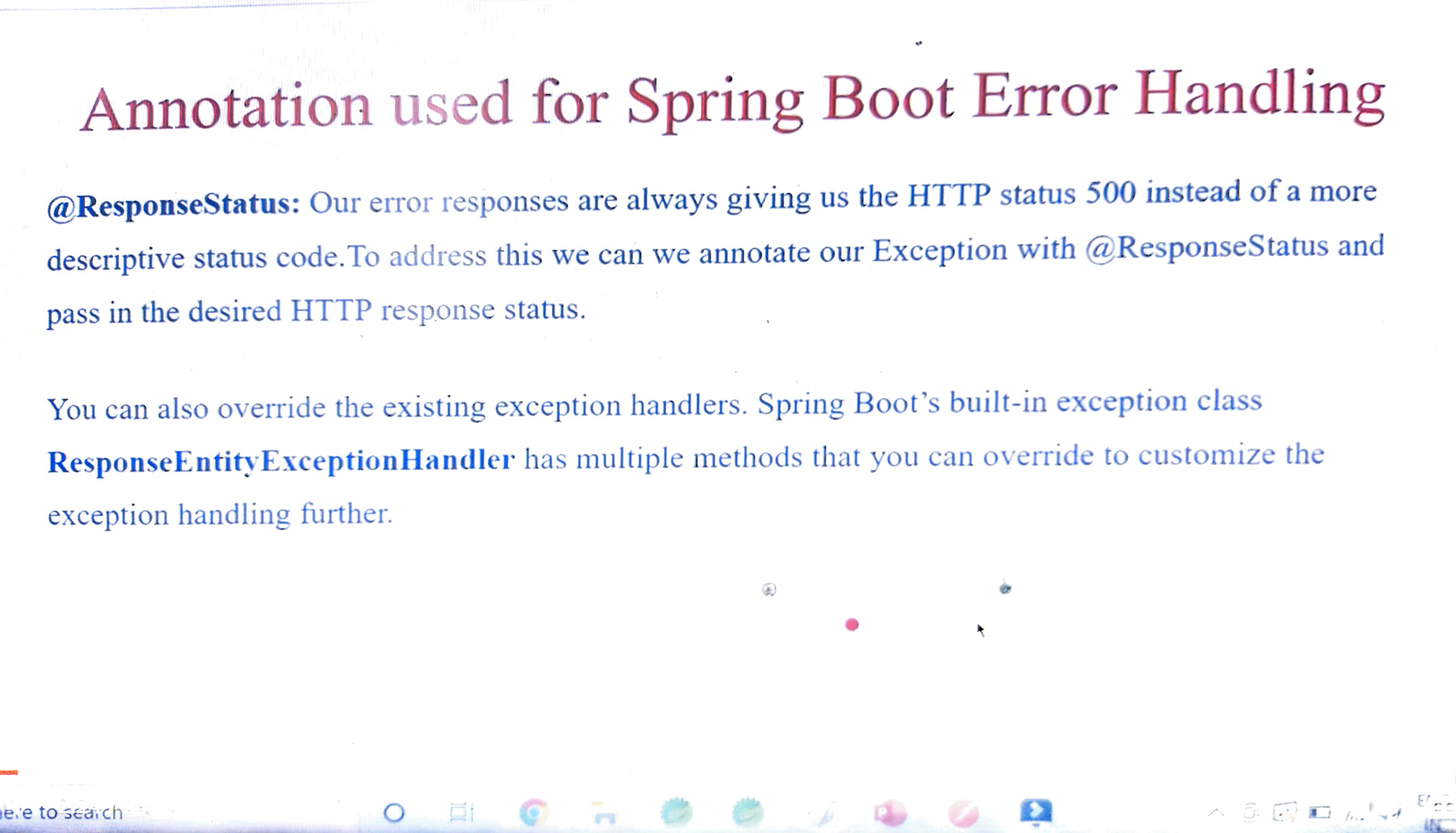
- **@ControllerAdvice :** The @ControllerAdvice annotation handles exceptions globally – it allows you to use the same ExceptionHandler for multiple controllers. This way, we can define how to treat an exception in just one place because this handler will be called whenever the exception is thrown from classes that are covered by ControllerAdvice.
- as the name suggests, is “Advice” for multiple controllers.
- allows our class to be a global interceptor of exceptions thrown by methods annotated by @RequestMapping.

Annotation used for Spring Boot Error Handling

- **@ExceptionHandler** : Spring annotation that provides a mechanism to treat exceptions that are thrown during execution of handlers (Controller operations). This annotation, if used on methods of controller classes, will serve as the entry point for handling exceptions thrown within this controller only.

- Altogether, the most common way is to use @ExceptionHandler on methods of @ControllerAdvice classes so that the exception handling will be applied globally or to a subset of controllers.

@ExceptionHandler and @ControllerAdvice are used to define a central point for treating exceptions and wrapping them up in a class.



Annotation used for Spring Boot Error Handling

@ResponseStatus: Our error responses are always giving us the HTTP status 500 instead of a more descriptive status code.To address this we can we annotate our Exception with @ResponseStatus and pass in the desired HTTP response status.

You can also override the existing exception handlers. Spring Boot's built-in exception class **ResponseEntityExceptionHandler** has multiple methods that you can override to customize the exception handling further.