

**Instructor Notes:**

Add instructor notes here.



**Instructor Notes:****Lesson Objectives**

- Exception handling at controller level
- Exception handling at application level

**References:**

<http://www.baeldung.com/exception-handling-for-rest-with-spring>

**Instructor Notes:****6.1 Exception handling at controller level**

we will define a method to handle exceptions, and annotate that with `@ExceptionHandler`

```
@RestController
public class CountryController{

    @ResponseStatus(value=HttpStatus.NOT_FOUND,
reason="Country with this id not present")
    @ExceptionHandler({Exception.class})
    public void handleException() {

    }
}
```

`@ExceptionHandler` along with `@ResponseStatus` to map the exception to the custom method in controller which can handle all exception in that controller. In `@ExceptionHandler` annotation we can include the Exception classes which we need to handle for this controller

**Instructor Notes:**

These demos can be executed for better understanding

## 6.1 Exception handling at controller level



SpringRESTDemoWithException



## Instructor Notes:

## 6.2 Exception handling at application level

**Spring 3.2** brings support for a global `@ExceptionHandler` with the new `@ControllerAdvice` annotation.

```
@ControllerAdvice
public class DemoException {
    @ResponseBody
    @ResponseStatus(value=HttpStatus.NOT_FOUND)
    @ExceptionHandler(value = {Exception.class})
    protected ErrorInfo handleConflict(Exception ex,
        HttpServletRequest req) {
        String bodyOfResponse = ex.getMessage();// "Country
        with this id not present";
        String uri = req.getRequestURL().toString();
        return new ErrorInfo(uri,bodyOfResponse);
    }
}
```

The new annotation allows the multiple scattered `@ExceptionHandler` from before to be consolidated into a **single, global error handling component**. The actual mechanism is extremely simple but also very flexible: it allows full control over the body of the response as well as the status code it allows mapping of several exceptions to the same method, to be handled together it makes good use of the newer RESTful *ResponseEntity* response

```
public class ErrorInfo {
    private String url;
    private String message;
    public ErrorInfo(String url, String message) {
        this.url = url;
        this.message = message;
    }
    public String getUrl() {
        return url;
    }
    public void setUrl(String url) {
        this.url = url;
    }
    public String getMessage() {
        return message;
    }
    public void setMessage(String message) {
        this.message = message;
    }
}
```

**Instructor Notes:**

These demos can be executed for better understanding

## 6.2 Exception handling at application level



SpringRESTDemoWithGlobalException




**Instructor Notes:**

Summary of the lesson.

### Summary

We have so far learnt

- Exception handling at controller level
- Exception handling at application level



Add the notes here.

**Instructor Notes:**

Corresponding lab  
assignment

Lab

Lab3



Lab will be added after Lab Book creation



**Instructor Notes:**

Question 1: Option 2

Question 2: True

**Review Question**

Question 1: Which of the following annotation is used for handler method?

- Option1 : @Exception
- Option 2: @ExceptionHandler
- Option 3: @HandlerMapping
- Option 4: @HandlerMethod

Question 2: We can handle Custom Exception in Spring REST.

- True
- False



Add the notes here.