## ****1.**@RequestBody**

Simply put, ****the** @RequestBody**annotation maps the**HttpRequest**body to a transfer or domain object, enabling automatic deserialization**** of the inbound HttpRequest body onto a Java object.

@PostMapping("/request")

public ResponseEntity postController(@RequestBody LoginForm loginForm) {

exampleService.fakeAuthenticate(loginForm);

return ResponseEntity.ok(HttpStatus.OK);

}

Spring automatically deserializes the JSON into a Java type, assuming an appropriate one is specified.

By default, ****the type we annotate with the**@RequestBody**annotation must correspond to the JSON sent from our client-side controller:****

1. ****@ResponseBody****

The @ResponseBody annotation tells a controller that the object returned is automatically serialized into JSON and passed back into the HttpResponse object.

Suppose we have a custom Response object:

public class ResponseTransfer {

private String text;

// standard getters/setters

}

Next, the associated controller can be implemented:

@Controller

@RequestMapping("/post")

public class ExamplePostController {

@Autowired

ExampleService exampleService;

@PostMapping("/response")

@ResponseBody

public ResponseTransfer postResponseController(

@RequestBody LoginForm loginForm) {

return new ResponseTransfer("Thanks For Posting!!!");

}

}

In the developer console of our browser or using a tool like Postman, we can see the following response:

{"text":"Thanks For Posting!!!"}

# **@RequestParam vs @PathVariable**

**@PathVariable**s extract values from the URI path.****

**E.g**

@GetMapping("/foos/{id}")

@ResponseBody

**public** String **getFooById**(@PathVariable String id) {

**return** "ID: " + id; }

**@PathVariable is extracting values from the URI path, it’s not encoded.**

E.g

http://localhost:8080/spring-mvc-basics/foos/ab+c

---- ID: ab+c

**@RequestParam**s extract values from the query string.****

****E.g.****

@GetMapping("/foos")

@ResponseBody **public** String **getFooByIdUsingQueryParam**(@RequestParam String id) { **return** "ID: " + id; }

**@RequestParam is encoded.**

http://localhost:8080/spring-mvc-basics/foos?id=ab+c

---- ID: ab c

We can make **@PathVariable** optional by using the required attribute.

4)

