**@RestController**: Marks a class as a RESTful controller. It combines the **@Controller** and **@ResponseBody** annotations, indicating that the class handles HTTP requests and returns the response in the body of the HTTP response.

**@RequestMapping**: Specifies the base URL or path for the controller. It can be applied at both the class and method levels. You can also use more specific annotations like @GetMapping, **@PostMapping, @PutMapping, @DeleteMapping**, etc., which are shortcuts for the **@RequestMapping** annotation with specific HTTP methods.

**@PathVariable**: Binds a method parameter to a path variable in the URL. It is used to extract dynamic values from the URL.

**@RequestParam**: Binds a method parameter to a query parameter or form data. It is used to extract values from the request URL or form data.

**@RequestBody**: Binds the request body to a method parameter. It is used to deserialize the request body into an object.

**@ResponseBody**: Specifies that the return value of a method should be serialized and included as the response body.

These are just a few of the many annotations available in Spring Boot for building RESTful APIs. Depending on your specific use case, you might use additional annotations such as @RequestHeader, **@RequestAttribute**, or @ResponseStatus to handle headers, attributes, or set the response status code, respectively.