

Spring
Boot

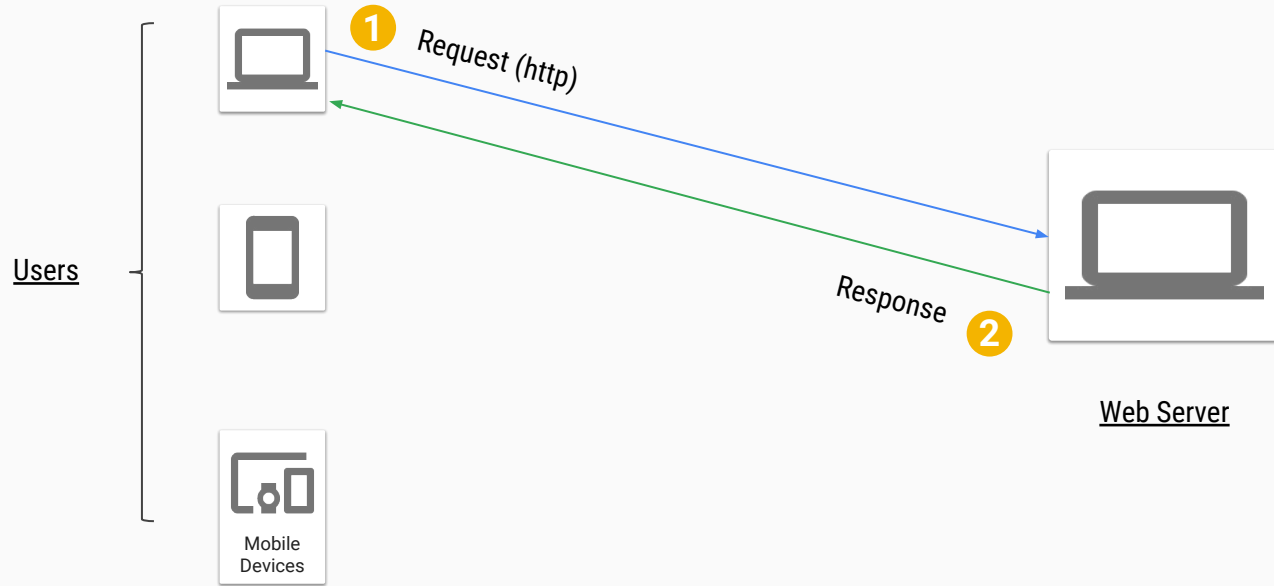
+

Web
Basics

Spring
Boot

+

Web
Basics



SPRING INITIALIZR bootstrap your application now

Generate a Maven Project ▾ with Java ▾ and Spring Boot 2.0.5 ▾

Project Metadata

Artifact coordinates

Group

com.example

Artifact

demo

Dependencies

Add Spring Boot Starters and dependencies to your application

Search for dependencies

Web, Security, JPA, Actuator, Devtools...

Selected Dependencies

Web ✕

DevTools ✕

Generate Project ⚙️ + ↵

Don't know what to look for? Want more options? [Switch to the full version.](#)

Pom.xml - Dependencies

```
<parent>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-parent</artifactId>
  <version>2.0.5.RELEASE</version>
  <relativePath/>
</parent>

<dependencies>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-web</artifactId>
  </dependency>
```

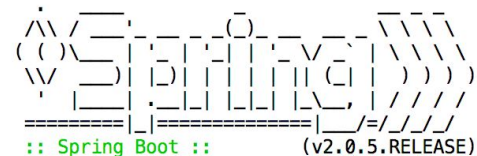
Makes application a web server

Application Main class

```
@SpringBootApplication
public class ToysApplication {

    public static void main(String[] args) {
        SpringApplication.run(ToysApplication.class, args);
    }
}
```

Run ApplicationMain



```
2018-10-14 12:29:18.260 INFO 1382 --- [ restartedMain] com.shop.toys.ToysApplication : Starting ToysApplication on Deepaks-MacBook-Pro.local with PID 1382 (/Use
2018-10-14 12:29:18.261 INFO 1382 --- [ restartedMain] com.shop.toys.ToysApplication : No active profile set, falling back to default profiles: default
2018-10-14 12:29:18.330 INFO 1382 --- [ restartedMain] ConfigServletWebServerApplicationContext : Refreshing org.springframework.boot.web.servlet.context.AnnotationConfigS
2018-10-14 12:29:19.747 INFO 1382 --- [ restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port(s): 8080 (http)
2018-10-14 12:29:19.767 INFO 1382 --- [ restartedMain] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2018-10-14 12:29:19.767 INFO 1382 --- [ restartedMain] org.apache.catalina.core.StandardEngine : Starting Servlet Engine: Apache Tomcat/8.5.34
2018-10-14 12:29:19.771 INFO 1382 --- [ost-startStop-1] o.a.catalina.core.AprLifecycleListener : The APR based Apache Tomcat Native library which allows optimal performan
2018-10-14 12:29:19.830 INFO 1382 --- [ost-startStop-1] o.a.c.c.C.[Tomcat].[/] : Initializing Spring embedded WebApplicationContext
2018-10-14 12:29:19.830 INFO 1382 --- [ost-startStop-1] o.s.web.context.ContextLoader : Root WebApplicationContext: initialization completed in 1505 ms
2018-10-14 12:29:19.901 INFO 1382 --- [ost-startStop-1] o.s.b.w.servlet.ServletRegistrationBean : Servlet dispatcherServlet mapped to [/]
2018-10-14 12:29:19.905 INFO 1382 --- [ost-startStop-1] o.s.b.w.servlet.FilterRegistrationBean : Mapping filter: 'characterEncodingFilter' to: [/]
2018-10-14 12:29:19.906 INFO 1382 --- [ost-startStop-1] o.s.b.w.servlet.FilterRegistrationBean : Mapping filter: 'hiddenHttpMethodFilter' to: [/]
2018-10-14 12:29:19.906 INFO 1382 --- [ost-startStop-1] o.s.b.w.servlet.FilterRegistrationBean : Mapping filter: 'httpPutFormContentFilter' to: [/]
2018-10-14 12:29:19.906 INFO 1382 --- [ost-startStop-1] o.s.b.w.servlet.FilterRegistrationBean : Mapping filter: 'requestContextFilter' to: [/]
2018-10-14 12:29:20.026 INFO 1382 --- [ restartedMain] o.s.w.s.handler.SimpleUrlHandlerMapping : Mapped URL path [/**/favicon.ico] onto handler of type [class org.springf
2018-10-14 12:29:20.228 INFO 1382 --- [ restartedMain] s.w.s.m.m.a.RequestMappingHandlerAdapter : Looking for @ControllerAdvice: org.springframework.boot.web.servlet.conte
2018-10-14 12:29:20.279 INFO 1382 --- [ restartedMain] s.w.s.m.m.a.RequestMappingHandlerMapping : Mapped "{[/greeting],methods=[GET]}" onto public java.lang.String com.sho
2018-10-14 12:29:20.284 INFO 1382 --- [ restartedMain] s.w.s.m.m.a.RequestMappingHandlerMapping : Mapped "{[/error]}" onto public org.springframework.http.ResponseEntity<j
2018-10-14 12:29:20.285 INFO 1382 --- [ restartedMain] s.w.s.m.m.a.RequestMappingHandlerMapping : Mapped "{[/error],produces=[text/html]}" onto public org.springframework
2018-10-14 12:29:20.305 INFO 1382 --- [ restartedMain] o.s.w.s.handler.SimpleUrlHandlerMapping : Mapped URL path [/webjars/**] onto handler of type [class org.springframe
2018-10-14 12:29:20.305 INFO 1382 --- [ restartedMain] o.s.w.s.handler.SimpleUrlHandlerMapping : Mapped URL path [/**] onto handler of type [class org.springframework.web
2018-10-14 12:29:20.432 INFO 1382 --- [ restartedMain] o.s.b.d.a.OptionalLiveReloadServer : LiveReload server is running on port 35729
2018-10-14 12:29:20.472 INFO 1382 --- [ restartedMain] o.s.j.e.a.AnnotationMBeanExporter : Registering beans for JMX exposure on startup
2018-10-14 12:29:20.525 INFO 1382 --- [ restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 8080 (http) with context path ''
2018-10-14 12:29:20.533 INFO 1382 --- [ restartedMain] com.shop.toys.ToysApplication : Started ToysApplication in 2.665 seconds (JVM running for 3.634)
2018-10-14 12:29:24.376 INFO 1382 --- [nio-8080-exec-1] o.a.c.c.C.[Tomcat].[/] : Initializing Spring FrameworkServlet 'dispatcherServlet'
2018-10-14 12:29:24.376 INFO 1382 --- [nio-8080-exec-1] o.s.web.servlet.DispatcherServlet : FrameworkServlet 'dispatcherServlet': initialization started
2018-10-14 12:29:24.392 INFO 1382 --- [nio-8080-exec-1] o.s.web.servlet.DispatcherServlet : FrameworkServlet 'dispatcherServlet': initialization completed in 16 ms
```

← → ↻ 🏠 ⓘ localhost:8080

Whitelabel Error Page

This application has no explicit mapping for /error, so you are seeing this as a fallback.

Sun Oct 14 12:43:40 IST 2018

There was an unexpected error (type=Not Found, status=404).

No message available

Code to handle requests

```
@RestController ← tell Spring this class handles requests
public class GreetingController {

    @RequestMapping(value = "/greeting", method = RequestMethod.GET)
    public String greet() {
        return "Hello there!";
    }
}
```

Map the URL to method

Allow only GET requests

← → ↻ 🏠 ⓘ localhost:8080/greeting

Hello there!

Shorthand for RequestMapping

```
@RequestMapping(value = "/greeting", method = RequestMethod.GET)  
public String greet() {  
    return "Hello there!";  
}
```


```
@GetMapping("/greeting") ← Shorthand  
public String greet() {  
    return "Hello there!";  
}
```

Sending parameters

← → ↻ 🏠 ⓘ localhost:8080/greeting?name=Jackie

```
@RestController
public class GreetingController {

    @GetMapping("/greeting")
    public String greet(@RequestParam(value = "name") String userName) {
        return "Hello there, " + userName;
    }
}
```



Two red arrows originate from the code below. One arrow points from the `value = "name"` in the `@RequestParam` annotation to the `name` parameter in the URL `localhost:8080/greeting?name=Jackie`. The other arrow points from the `userName` parameter in the `greet` method signature to the `name` parameter in the same URL.

Sending parameters

← → ↻ 🏠 ⓘ localhost:8080/greeting?name=Jackie

Hello there, Jackie

Sending parameters

← → ↻ 🏠 ⓘ localhost:8080/greeting?name=Jackie

```
@RestController
public class GreetingController {

    @GetMapping("/greeting")
    public String greet(@RequestParam String name) {
        return "Hello there, " + name;
    }
}
```



Sending multiple parameters

```
@GetMapping("/greeting")
public String greet(@RequestParam String name,
                   @RequestParam String time) {
    return "Good " + time + ", " + name;
}
```

2 parameters


← → ↻ 🏠 ⓘ localhost:8080/greeting?name=Jackie&time=morning

Good morning, Jackie

Sending optional parameters

```
@GetMapping("/greeting")
public String greet(@RequestParam String name,
                   @RequestParam(required = false) String time) {

    if (time == null) {
        time = "day";
    }
    return "Good " + time + ", " + name;
}
```



optional

← → ↻ 🏠 ⓘ localhost:8080/greeting?name=Jackie&time=morning

Good morning, Jackie

← → ↻ 🏠 ⓘ localhost:8080/greeting?name=Jackie

Good day, Jackie

Setting default value

```
@GetMapping("/greeting")
public String greet(@RequestParam String name,
                   @RequestParam(required = false, defaultValue = "day") String time)
    return "Good " + time + ", " + name;
}
```

← → ↻ 🏠 ⓘ localhost:8080/greeting?name=Jackie&time=morning

Good morning, Jackie

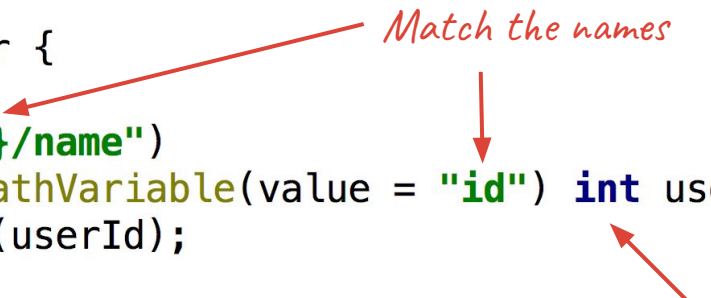
← → ↻ 🏠 ⓘ localhost:8080/greeting?name=Jackie

Good day, Jackie

Path variables

```
@RestController
public class UserController {

    @GetMapping("/user/{id}/name")
    public String greet(@PathVariable(value = "id") int userId) {
        return getUserName(userId);
    }
}
```



← → ↻ 🏠 ⓘ localhost:8080/user/45/name

Jackie

*You can extract multiple
path variables at a time!*

Request Headers

```
@GetMapping("/request-header")  
public String language(@RequestHeader(value = "Accept-Language") String language) {  
    return language;  
}
```

← → ↻ 🏠 ⓘ localhost:8080/request-header

en-US,en;q=0.9,ca;q=0.8

Name of the header. It can also be a custom header eg: token


Handling multiple URLs in single controller


```
@RestController
public class UserController {


    @GetMapping("/user/{id}/name") ← ①
    public String name(@PathVariable(value = "id") int userId) {
        return getUserName(userId);
    }

    @GetMapping("/user/{id}/age") ← ②
    public int age(@PathVariable(value = "id") int userId) {
        return getUserAge(userId);
    }
}
```

Managing top level paths

```
@RestController
@RequestMapping("/user")  Extract common path here
public class UserController {

    @GetMapping("/{id}/name")  Path = /user + /{id}/name
    public String name(@PathVariable(value = "id") int userId) {
        return getUserName(userId);
    }

    @GetMapping("/{id}/age")  Path = /user + /{id}/age
    public int age(@PathVariable(value = "id") int userId) {
        return getUserAge(userId);
    }
}
```

Handling JSON

```
@RestController
public class ToyController {

    @GetMapping("/toy/elephant")
    public Toy elephant() {
        return new Toy(name: "eli", size: "medium", color: "gray");
    }
}
```

← → ↻ 🏠 ⓘ localhost:8080/toy/elephant

```
{"name": "eli", "size": "medium", "color": "gray"}
```

Post requests + Sending bodies

```
@PostMapping("/toy/add")
public void addToy(@RequestBody Toy newToy){
    addToDatabase(newToy);
}
```

```
fetch ('http://localhost:8080/toy/add', {
  method: "POST",
  headers: {
    "Content-Type": "application/json; charset=utf-8",
  },
  body: JSON.stringify({name: "Teddy", size: "big", color: "white"})
});
```

```
@GetMapping("/greeting")
public String greet(@RequestParam String name,
                   @RequestParam String time) {
    return "Good " + time + ", " + name;
}
```

← → ↻ 🏠 ⓘ localhost:8080/greeting?name=Jackie

Whitelabel Error Page

This application has no explicit mapping for /error, so you are seeing this as a fallback.

Sun Oct 14 19:35:47 IST 2018

There was an unexpected error (type=Bad Request, status=400).
Required String parameter 'time' is not present

Custom error checks

```
@RestController
public class GreetingController {

    List<String> times = Lists.newArrayList( ...elements: "evening", "morning");

    @GetMapping("/greeting")
    public ResponseEntity<String> greet(@RequestParam String name,
                                       @RequestParam String time) {


        if (!times.contains(time)) {
            return ResponseEntity.badRequest().body(t: "Time should be morning/evening");
        }

        return ResponseEntity.ok("Good " + time + ", " + name);
    }
}
```

Custom error checks

← → ↻ 🏠 ⓘ localhost:8080/greeting?name=Jackie&time=afternoon

Time should be morning/evening

Name	Status	Type
 greeting?name=Jacki...	400	document

Custom error checks

```
@RestController
public class GreetingController2 {


    @GetMapping("/secret-greeting")
    public ResponseEntity<String> greet(@RequestParam String name,
                                       @RequestParam String secret) {

        if (!"abc123".equals(secret)) {
            return ResponseEntity.status(HttpStatus.UNAUTHORIZED).body("Not authorized")
        }

        return ResponseEntity.ok("Welcome to the toy land, " + name);
    }
}
```

← → ↺ 🏠 ⓘ localhost:8080/secret-greeting?name=Jackie&secret=dontknow


Not authorized

Name	Status
 secret-greeting?nam...	401

Custom error checks

← → ↻ 🏠 ⓘ localhost:8080/secret-greeting?name=Jackie&secret=dontknow

Not authorized

Name	Status
 secret-greeting?nam...	401

← → ↻ 🏠 ⓘ localhost:8080/secret-greeting?name=Jackie&secret=abc123

Welcome to the toy land

Summary

- `@RestController` = servlet (handle incoming requests)
- `@RequestMapping (url)` = map URL to method
- `@GetMapping` / `@PostMapping` = shortcuts
- `required = false` = optional
- `@RequestBody`
- `@PathVariable` = extract path values
- Auto JSON mapping
- `ResponseEntity` = custom response status