

[\(http://baeldung.com\)](http://baeldung.com)

Model, ModelMap, and ModelAndView in Spring MVC

Last modified: October 26, 2017

by baeldung (<http://www.baeldung.com/author/baeldung/>)

Spring (<http://www.baeldung.com/category/spring/>) +

Spring MVC (<http://www.baeldung.com/category/spring-mvc/>)

I just announced the new *Spring 5* modules in REST With Spring:

>> CHECK OUT THE COURSE (</rest-with-spring-course#new-modules>)

1. Overview

In this article, we'll look at the use of the core `org.springframework.ui.Model`, `org.springframework.ui.ModelMap` and `org.springframework.web.servlet.ModelView` provided by Spring MVC.

2. Maven Dependencies

Let's start with the *spring-context* dependency in our *pom.xml* file:

```
1 <dependency>
2   <groupId>org.springframework</groupId>
3   <artifactId>spring-context</artifactId>
4   <version>4.3.11.RELEASE</version>
5 </dependency>
```

The latest version of spring-context dependency can be found here (<http://search.maven.org/#search%7Cga%7C1%7Ca%3A%22spring-context%22>).

For the *ModelAndView*, the *spring-web* dependency is required:

```
1 <dependency>
2   <groupId>org.springframework</groupId>
3   <artifactId>spring-web</artifactId>
4   <version>4.3.11.RELEASE</version>
5 </dependency>
```

The latest versions of spring-web dependency can be found here (<http://search.maven.org/#search%7Cga%7C1%7Ca%3A%22spring-web%22>).

And, if we use Thymeleaf as our view, we should add this dependency to pom.xml:

```
1 <dependency>
2   <groupId>org.thymeleaf</groupId>
3   <artifactId>thymeleaf-spring3</artifactId>
4   <version>3.0.8.RELEASE</version>
5 </dependency>
```

The latest version of Thymeleaf dependency can be found here (<https://mvnrepository.com/artifact/org.thymeleaf/thymeleaf-spring3>).

3. Model

Let's start with the most basic concept here – the *Model*.

Simply put, the model can supply attributes used for rendering views.

To provide a view with usable data, we simply add this data to its *Model* object. Additionally, maps with attributes can be merged with *Model* instances:

```
1 @GetMapping("/showViewPage")
2 public String passParametersWithModel(Model model) {
3     Map<String, String> map = new HashMap<>();
4     map.put("spring", "mvc");
5     model.addAttribute("message", "Baeldung");
6     model.mergeAttributes(map);
7     return "viewPage";
8 }
```

4. ModelMap

Just like the *Model* interface above, *ModelMap* is also used to pass values to render a view.

The advantage of *ModelMap* is it gives us the ability to pass a collection of values and treat these values as if they were within a *Map*:

```
1 @GetMapping("/printViewPage")
2 public String passParametersWithModelMap(ModelMap map) {
3     map.addAttribute("welcomeMessage", "welcome");
4     map.addAttribute("message", "Baeldung");
5     return "viewPage";
6 }
```

5. ModelAndView

The final interface to pass values to a view is the *ModelAndView*. This interface allows us to pass all the information required by Spring MVC in one return:

This interface allows us to pass all the information required by Spring MVC in one return:

```
1 @GetMapping("/goToViewPage")
2 public ModelAndView passParametersWithModelAndView() {
3     ModelAndView modelAndView = new ModelAndView("viewPage");
4     modelAndView.addObject("message", "Baeldung");
5     return modelAndView;
6 }
```

6. The View

All the data, we place within these models, is used by a view – in general, a templated view to render the web page.

If we have a Thymeleaf template file targeted by our controller's methods as their view. A parameter passed through the model will be accessible from within the thymeleaf HTML code:

```
1 <!DOCTYPE HTML>
2 <html xmlns:th="http://www.thymeleaf.org (http://www.thymeleaf.org)">
3 <head>
4   <title>Title</title>
5 </head>
6 <body>
7   <div>Web Application. Passed parameter : th:text="${message}"</div>
8 </body>
9 </html>
```

The parameter passed here is used through the syntax *`\${message}`*, which is known as a placeholder. The Thymeleaf template engine will replace this placeholder with an actual value from an attribute of the same name passed through the model.

7. Conclusion

In this quick tutorial, we've discussed three core concepts in Spring MVC – the *Model*, the *ModelMap* and the *ModelAndView*. We also had a look at examples of how the view can make use of these values.

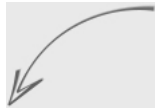
As always, the implementation of all these examples and code snippets can be found over on Github (<https://github.com/eugenp/tutorials/tree/master/spring-all>).

I just announced the new Spring 5 modules in REST With Spring:

>> CHECK OUT THE LESSONS (/rest-with-spring-course#new-modules)



Build your Microservice Architecture with Spring Boot and Spring Cloud



Enter your email Address

Download Now

Sort by: newest



Ernest

Guest

Is there any advantages of using one over the other?? Or it only really depends on ones use case?



1 month 3 days ago



Grzegorz Piwowarek
(<http://4comprehension.com>)

(<http://www.baeldung.com/author/grzegorz-author/>)

Editor



Totally depends on the use-case

1 month 3 days ago

CATEGORIES

[SPRING \(HTTP://WWW.BAELDUNG.COM/CATEGORY/SPRING/\)](http://www.baeldung.com/category/spring/)
[REST \(HTTP://WWW.BAELDUNG.COM/CATEGORY/REST/\)](http://www.baeldung.com/category/rest/)
[JAVA \(HTTP://WWW.BAELDUNG.COM/CATEGORY/JAVA/\)](http://www.baeldung.com/category/java/)
[SECURITY \(HTTP://WWW.BAELDUNG.COM/CATEGORY/SECURITY-2/\)](http://www.baeldung.com/category/security-2/)
[PERSISTENCE \(HTTP://WWW.BAELDUNG.COM/CATEGORY/PERSISTENCE/\)](http://www.baeldung.com/category/persistence/)
[JACKSON \(HTTP://WWW.BAELDUNG.COM/CATEGORY/JACKSON/\)](http://www.baeldung.com/category/jackson/)
[HTTPCLIENT \(HTTP://WWW.BAELDUNG.COM/CATEGORY/HTTP/\)](http://www.baeldung.com/category/http/)

ABOUT

[ABOUT BAELDUNG \(HTTP://WWW.BAELDUNG.COM/ABOUT/\)](http://www.baeldung.com/about/)
[THE COURSES \(HTTP://COURSES.BAELDUNG.COM\)](http://courses.baeldung.com)
[META BAELDUNG \(HTTP://META.BAELDUNG.COM/\)](http://meta.baeldung.com/)
[THE FULL ARCHIVE \(HTTP://WWW.BAELDUNG.COM/FULL_ARCHIVE\)](http://www.baeldung.com/full_archive)
[WRITE FOR BAELDUNG \(HTTP://WWW.BAELDUNG.COM/CONTRIBUTION-GUIDELINES\)](http://www.baeldung.com/contribution-guidelines)
[PRIVACY POLICY \(HTTP://WWW.BAELDUNG.COM/PRIVACY-POLICY\)](http://www.baeldung.com/privacy-policy)
[TERMS OF SERVICE \(HTTP://WWW.BAELDUNG.COM/TERMS-OF-SERVICE\)](http://www.baeldung.com/terms-of-service)
[CONTACT \(HTTP://WWW.BAELDUNG.COM/CONTACT\)](http://www.baeldung.com/contact)
[COMPANY INFO \(HTTP://WWW.BAELDUNG.COM/BAELDUNG-COMPANY-INFO\)](http://www.baeldung.com/baeldung-company-info)
[ADVERTISE ON THE JAVA WEEKLY \(HTTP://WWW.BAELDUNG.COM/JAVA-WEEKLY-SPONSORSHIP\)](http://www.baeldung.com/java-weekly-sponsorship)
[CONSULTING WORK \(HTTP://WWW.BAELDUNG.COM/CONSULTING\)](http://www.baeldung.com/consulting)