

# Spring Boot

intermediate

# THE PACKAGE MANAGER FOR WINDOWS

## Modern Software Automation

Why Chocolatey

Get Started

Find Packages



WEBINAR

### Chocolatey for Business OVERVIEW AND DEMONSTRATION

📅 Every Friday at 3:00 AM GMT+9 / 6:00 PM GMT

Every Thursday one of our Product Consultants will host a live overview and demonstration of Chocolatey for

Now run the following command:

```
> Set-ExecutionPolicy Bypass -Scope Process -Force; [System.Net.ServicePointManager]::SecurityProtocol = [System.Net.ServicePointManager]
```



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6
PS C:\Users\mausti> choco -version
Parsing version resulted in error (converted to warning):
Cannot bundle unregistered option '-e'.
This is a listing of all of the different things you can pass to choco.

Commands

* list - lists remote or local packages
* find - searches remote or local packages (alias for search)
* search - searches remote or local packages (alias for list)
* info - retrieves package information. Shorthand for choco search pkgnname --exact --verbose
* install - installs packages from various sources
* pin - suppress upgrades for a package
* outdated - retrieves packages that are outdated. Similar to upgrade all --noop
* upgrade - upgrades packages from various sources
* uninstall - uninstalls a package
* pack - packages up a nuspec to a compiled nupkg
* push - pushes a compiled nupkg
* new - generates files necessary for a chocolatey package from a template
* sources - view and configure default sources (alias for source)
* source - view and configure default sources
* config - Retrieve and configure config file settings
* feature - view and configure choco features
* features - view and configure choco features (alias for feature)
* setapikey - retrieves, saves or deletes an apikey for a particular source (alias for apikey)
* apikey - retrieves, saves or deletes an apikey for a particular source
* unpackself - have chocolatey set itself up
* version - [DEPRECATED] will be removed in v1 - use `choco outdated` or `cup <pkgall> -whatif` instead
* update - [DEPRECATED] RESERVED for future use (you are looking for upgrade, these are not the droi
for)
```

```
Windows PowerShell
user name. Overrides the default proxy password (encrypted in settings if set). Available for config settings in 0.9.9.9+, this CLI option available in 0.10.4+.
--proxy-bypass-list=VALUE
ProxyBypassList - Comma separated list of regex locations to bypass on proxy. Requires explicit proxy ('--proxy' or config setting). Overrides the default proxy bypass list of ''. Available in 0.10.4+.

--proxy-bypass-on-local
Proxy Bypass On Local - Bypass proxy for local connections. Requires explicit proxy ('--proxy' or config setting). Overrides the default proxy bypass on local setting of 'True'. Available in 0.10.4+.

--log-file=VALUE
Log File to output to in addition to regular loggers. Available in 0.1-0.8+.
Chocolatey v0.10.15
PS C:\Users\mausti> choco install maven
Chocolatey v0.10.15
Chocolatey detected you are not running from an elevated command shell (cmd/powershell).

You may experience errors - many functions/packages require admin rights. Only advanced users should run choco w/out an elevated shell. When you open the command shell, you should ensure that you do so with "Run as Administrator" selected. If you are attempting to use Chocolatey in a non-administrator setting, you must select a different location other than the default install location. See https://chocolatey.org/install#non-administrative-install for details.

Do you want to continue? [Y]es, [N]o: -
```

# spring initializr

## Project

Maven Project

Gradle Project

## Language

Java

Kotlin

Groovy

## Spring Boot

2.5.1 (SNAPSHOT)

2.5.0

2.4.7 (SNAPSHOT)

2.4.6

2.3.12 (SNAPSHOT)

2.3.11

## Project Metadata

Group com.austin

Artifact home

Name home

Description Demo project for Spring Boot

Package name com.austin.home

Packaging  Jar

War

Java  16

11

8

## Dependencies

[ADD DEPENDENCIES...](#) CTRL +

### Spring Boot DevTools

DEVELOPER TOOLS

Provides fast application restarts, LiveReload, and configurations for enhanced development experience.

### Lombok

DEVELOPER TOOLS

Java annotation library which helps to reduce boilerplate code.

### Spring Web

WEB

Build web, including RESTful, applications using Spring MVC. Uses Apache Tomcat as the default embedded container.

### Thymeleaf

TEMPLATE ENGINES

A modern server-side Java template engine for both web and standalone environments. Allows HTML to be correctly displayed in browsers and as static prototypes.

### Spring Security

SECURITY

Highly customizable authentication and access-control framework for Spring applications.

### MariaDB Driver

SQL

MariaDB JDBC and R2DBC driver.

[GENERATE](#) CTRL + ↵

[EXPLORE](#) CTRL + SPACE

[SHARE...](#)



# Build fast, responsive sites with Bootstrap

Quickly design and customize responsive mobile-first sites with Bootstrap, the world's most popular front-end open source toolkit, featuring Sass variables and mixins, responsive grid system, extensive prebuilt components, and powerful JavaScript plugins.



## Get started

## Download

# Starter template

Be sure to have your pages set up with the latest design and development standards. That means using an HTML5 doctype and including a viewport meta tag for proper responsive behaviors. Put it all together and your pages should look like this:

```
<!doctype html>
<html lang="en">
  <head>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">

    <!-- Bootstrap CSS -->
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.1/dist/css/bootstrap.min.css" rel="stylesheet">

    <title>Hello, world!</title>
  </head>
  <body>
    <h1>Hello, world!</h1>

    <!-- Optional JavaScript; choose one of the two! -->

    <!-- Option 1: Bootstrap Bundle with Popper -->
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.1/dist/js/bootstrap.bundle.min.js" integrity="sha384-MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/tWtIaxVXM" crossorigin="anonymous"></script>

    <!-- Option 2: Separate Popper and Bootstrap JS -->
    <!--
    <script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.9.2/dist/umd/popper.min.js" integrity="sha384-IQsoaAHFAlZIUoiUQJ08xZvTLwSkAJqhiARqPcI9aLYyqMIcp+Q9K9" crossorigin="anonymous"></script>
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.1/dist/js/bootstrap.min.js" integrity="sha384-64fVh2uX8gA4FYmQ2G8vE9sd9h80rO2EUUqCvB4OyMzTn9w70Ooqtcw8L5IuqC" crossorigin="anonymous"></script>
  </body>
</html>
```

Copy



# Containers

[View on GitHub](#)

Containers are a fundamental building block of Bootstrap that contain, pad, and align your content within a given device or viewport.

## How they work

Containers are the most basic layout element in Bootstrap and are **required when using our default grid system**. Containers are used to contain, pad, and (sometimes) center the content within them. While containers *can* be nested, most layouts do not require a nested container.

Bootstrap comes with three different containers:

- `.container`, which sets a `max-width` at each responsive breakpoint
- `.container-fluid`, which is `width: 100%` at all breakpoints
- `.container-{breakpoint}`, which is `width: 100%` until the specified breakpoint

The table below illustrates how each container's `max-width` compares to the original `.container` and `.container-fluid` across each breakpoint.

See them in action and compare them in our [Grid example](#).

# Grid system

[View on GitHub](#)

Use our powerful mobile-first flexbox grid to build layouts of all shapes and sizes thanks to a twelve column system, six default responsive tiers, Sass variables and mixins, and dozens of predefined classes.

## Example

Bootstrap's grid system uses a series of containers, rows, and columns to layout and align content. It's built with [flexbox](#) and is fully responsive. Below is an example and an in-depth explanation for how the grid system comes together.

**New to or unfamiliar with flexbox?** [Read this CSS Tricks flexbox guide](#) for background, terminology, guidelines, and code snippets.

Column

Column

Column

# Examples

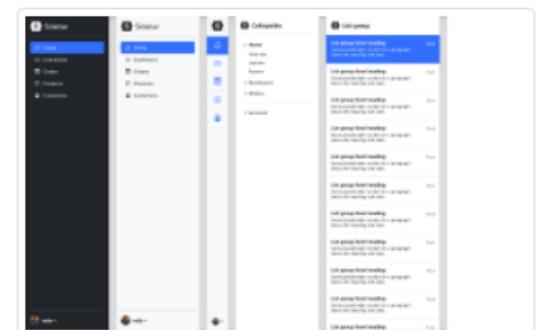
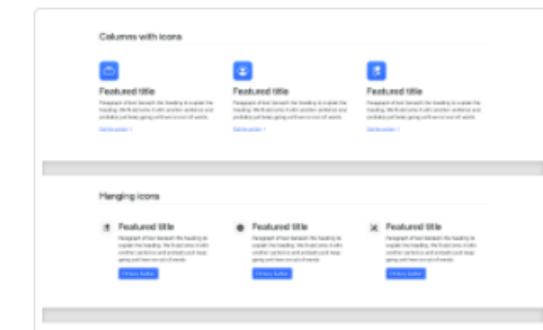
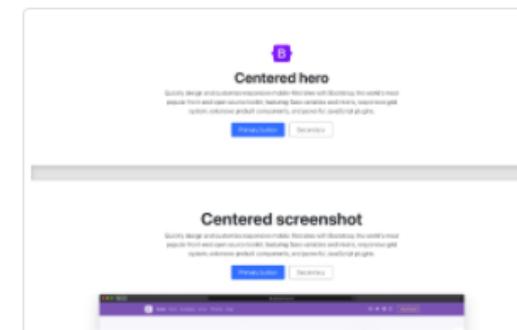
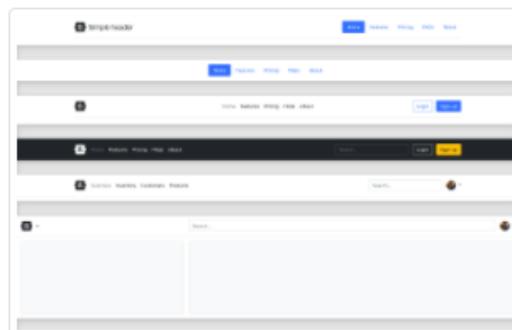
Quickly get a project started with any of our examples ranging from using parts of the framework to custom components and layouts.

[Download examples](#)

[Download source code](#)

## Snippets

Common patterns for building sites and apps that build on existing components and utilities with custom CSS and more.



# Examples

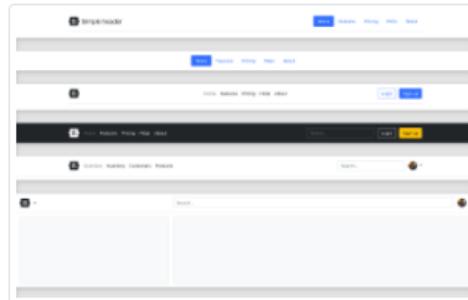
Quickly get a project started with any of our examples ranging from using parts of the framework to custom components and layouts.

[Download examples](#)

[Download source code](#)

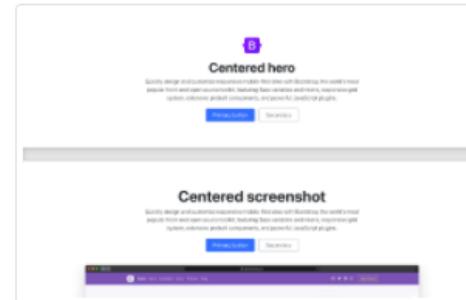
## Snippets

Common patterns for building sites and apps that build on existing components and utilities with custom CSS and more.



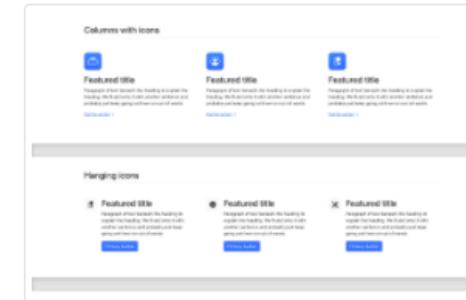
### Headers

Display your branding, navigation, search, and more with these header components



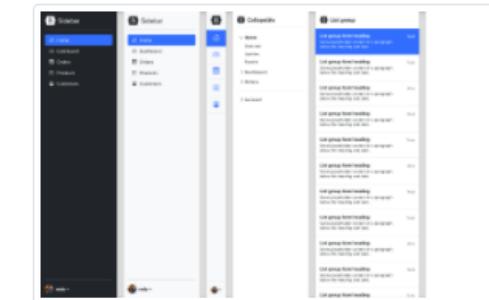
### Heroes

Set the stage on your homepage with heroes that feature clear calls to action.



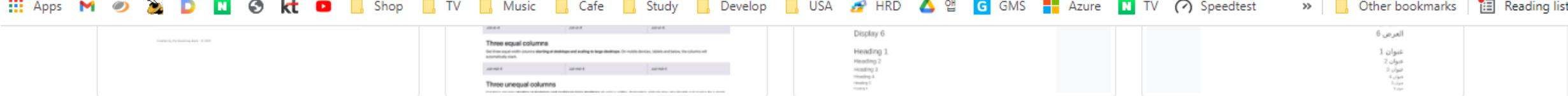
### Features

Explain the features, benefits, or other details in your marketing content.



### Sidebars

Common navigation patterns ideal for offcanvas or multi-column layouts.



## [Starter template](#)

Nothing but the basics: compiled CSS and JavaScript.

## [Grid](#)

Multiple examples of grid layouts with all four tiers, nesting, and more.

## [Cheatsheet](#)

Kitchen sink of Bootstrap components.

## [Cheatsheet RTL](#)

Kitchen sink of Bootstrap components, RTL.

# Navbars

Taking the default navbar component and showing how it can be moved, placed, and extended.



## [Navbars](#)

Demonstration of all responsive and container options for the navbar.

## [Navbar static](#)

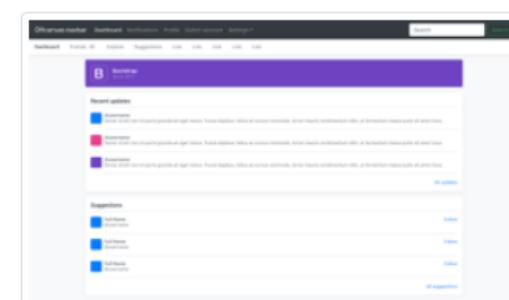
Single navbar example of a static top navbar along with some additional content.

## [Navbar fixed](#)

Single navbar example with a fixed top navbar along with some additional content.

## [Navbar bottom](#)

Single navbar example with a bottom navbar along with some additional content.



## Navbar example

This example is a quick exercise to illustrate how the top-aligned navbar works. As you scroll, this navbar remains in its original position and moves with the rest of the page.

[View navbar docs »](#)



# Thymeleaf

29 October 2018: Version 3.0.4 of the [Thymeleaf + Spring Security](#) integration package has been published, including support for Spring Security 5 in both Spring MVC and Spring WebFlux applications. See the [announcement](#).

29 October 2018: Thymeleaf 3.0.11 has been published. See the [announcement](#).

Thymeleaf is a modern server-side Java template engine for both web and standalone environments.

Thymeleaf's main goal is to bring elegant *natural templates* to your development workflow – displayed in browsers and also work as static prototypes, allowing for stronger collaboration.

## Tutorials

These are both learning and reference materials for Thymeleaf. If you want to know how to use Thymeleaf, then start here.

### Using Thymeleaf

All the Thymeleaf basics: from an introduction to Thymeleaf and how to use it, to a reference of all its processors/utilities. A must read for new and veteran Thymeleaf users.

- Thymeleaf 3.0: [Read online](#), [PDF](#), [EPUB](#), [MOBI](#)
- Thymeleaf 2.1: [Read online](#)
- Japanese translation (*non-official*): [Read online](#), [PDF](#), [EPUB](#), [MOBI](#) | [Read online \(v2.1\)](#)
- Example application: [Good Thymes Virtual Grocery](#)

### Thymeleaf + Spring

Docs and reference specific to Spring integration. Be sure to give [Using Thymeleaf](#) a read before venturing here.

B Introduction · Bootstrap v4.5 x | Hello, Spring Boot! x | Tutorial: Using Thymeleaf x | Suggested: Spring Boot으로 웹

← → C thymeleaf.org/doc/tutorials/3.0/usingthymeleaf.html#fragments

7.1 Simple conditionals: "if" and "unless"

7.2 Switch statements

8 Template Layout

8.1 Including template fragments

Defining and referencing fragments

Fragment specification syntax

Referencing fragments without th:fragment

Difference between th:insert and th:replace (and th:include)

8.2 Parameterizable fragment signatures

Fragment local variables without fragment arguments

A *default expression* is a special kind of conditional value we present in some languages like Groovy, lets you specify two expressions: the first one is used to null, but if it does then the second one is used.

Let's see it in action in our user profile page:

```
<div th:object="${session.user}">
...
<p>Age: <span th:text="*{age}?: '(no age specified)'>27</span>.</p>
</div>
```

As you can see, the operator is `?:`, and we use case) only if the result of evaluating `*{age}` is

```
<p>Age: <span th:text="*{age != null}?: '(no age specified)'>27</span>.</p>
```

As with conditional values, they can contain nested fragments.

th:replace

0/0

B Introduction · Bootstrap v4.5 x | Hello, Spring Boot! x | Tutorial: Using Thymeleaf x | +

← → C thymeleaf.org/doc/tutorials/3.0/usingthymeleaf.html#difference-between-thinsert-and-threplace-and-thinclude

7.1 Simple conditionals: "if" and "unless"

7.2 Switch statements

8 Template Layout

8.1 Including template fragments

Defining and referencing fragments

Fragment specification syntax

Referencing fragments without th:fragment

Difference between th:insert and th:replace (and th:include)

8.2 Parameterizable fragment signatures

Fragment local variables without fragment arguments

th:assert for in-template assertions

8.3 Flexible layouts: beyond mere fragment insertion

Using the empty fragment

- `th:insert` is the simplest: it will simply insert the specified fragment as the content of the host element.
- `th:replace` actually *replaces* its host tag with the specified fragment.
- `th:include` is similar to `th:insert`, but instead of inserting the fragment as the content of the host element, it inserts it as a child element.

So an HTML fragment like this:

```
<footer th:fragment="copy">
    &copy; 2011 The Good Thymes Virtual Grocery
</footer>
```

...included three times in host `<div>` tags, like this:

```
<body>
...
<div th:insert="footer :: copy"></div>
<div th:replace="footer :: copy"></div>
<div th:include="footer :: copy"></div>
```

7.1 Simple conditionals: "if" and "unless"

7.2 Switch statements

8 Template Layout

8.1 Including template fragments

Defining and referencing fragments

Fragment specification syntax

Referencing fragments without th:fragment

Difference between th:insert and  
th:replace (and th:include)

8.2 Parameterizable fragment signatures

Fragment local variables without fragment  
arguments

th:assert for in-template assertions

8.3 Flexible layouts: beyond mere fragment  
insertion

## Defining and referencing fragments

In our templates, we will often want to include parts from other templates.

In order to do this, Thymeleaf needs us to define these parts, "fragments", using the `th:fragment` attribute.

Say we want to add a standard copyright footer to all our grocery pages, so we create an `-INF/templates/footer.html` file containing this code:

```
<!DOCTYPE html>

<html xmlns:th="http://www.thymeleaf.org">

    <body>

        <div th:fragment="copy">
            © 2011 The Good Thymes Virtual Grocery
        </div>

    </body>

</html>
```

index.html common.html # starter-template.css

templates > fragments > common.html > html

```
1  <!DOCTYPE html>
2
3  <html xmlns:th="http://www.thymeleaf.org">
4
5      <body>
6
7          <div th:fragment="copy">
8              © 2011 The Good Thymes Virtual Grocery
9          </div>
10
11      </body>
12
13  </html>
```

index.html common.html # starter-template.css

```
templates > fragments > common.html > html > body > div
1  <!DOCTYPE html>
2  <html xmlns:th="http://www.thymeleaf.org">
3      <body>
4          <div th:fragment="menu">
5              |
6          </div>
7      </body>
8  </html>
```

index.html common.html # starter-template.css

```
templates > fragments > common.html > html > body > div
1  <!DOCTYPE html>
2  <html xmlns:th="http://www.thymeleaf.org">
3      <body>
4          <div th:fragment="menu">
5              |
6          </div>
7      </body>
8  </html>
```

index.html common.html # starter-template.css

templates > fragments > common.html > html > body > nav.navbar.navbar-expand-md.navbar-dark.bg-dark.fixed-top

```
1  <!DOCTYPE html>
2  <html xmlns:th="http://www.thymeleaf.org">
3  <body>
4      <nav class="navbar navbar-expand-md navbar-dark bg-dark fixed-top" th:fragment=>
5          <a class="navbar-brand" href="#">Spring Boot Tutorial</a>
6          <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarsExampleDefault"
7              aria-controls="navbarsExampleDefault" aria-expanded="false" aria-label="Toggle navigation">
8              <span class="navbar-toggler-icon"></span>
9          </button>
10
11         <div class="collapse navbar-collapse" id="navbarsExampleDefault">
12             <ul class="navbar-nav mr-auto">
13                 <li class="nav-item active">
14                     <a class="nav-link" href="#">홈 <span class="sr-only">(current)</span></a>
15                 </li>
16                 <li class="nav-item">
17                     <a class="nav-link" href="#">개시판</a>
18                 </li>
19                 <!-- <li class="nav-item">
20                     <a class="nav-link disabled" href="#" tabindex="-1" aria-disabled="true">Disabled</a>
21                 </li>
22                 <li class="nav-item dropdown">
23                     <a class="nav-link dropdown-toggle" href="#" id="dropdown01" data-
24                         aria-haspopup="true" aria-expanded="false">Dropdown</a>
25                     <div class="dropdown-menu" aria-labelledby="dropdown01">
26                         <a class="dropdown-item" href="#">Action</a>
27                         <a class="dropdown-item" href="#">Another action</a>
28                         <a class="dropdown-item" href="#">Something else here</a>
29                     </div>
30                 </li> -->
31         </ul>
32     </div>
33 
```

index.html common.html # starter-template.css

templates > fragments > common.html > html > body > nav.navbar.navbar-expand-md.navbar-dark.bg-dark.fixed-top

```
1  <!DOCTYPE html>
2  <html xmlns:th="http://www.thymeleaf.org">
3  <body>
4      <nav class="navbar navbar-expand-md navbar-dark bg-dark fixed-top" th:fragment="menu">
5          <a class="navbar-brand" href="#">Spring Boot Tutorial</a>
6          <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarsExampleDefault"
7              aria-controls="navbarsExampleDefault" aria-expanded="false" aria-label="Toggle navigation">
8              <span class="navbar-toggler-icon"></span>
9          </button>
10
11         <div class="collapse navbar-collapse" id="navbarsExampleDefault">
12             <ul class="navbar-nav mr-auto">
13                 <li class="nav-item active">
14                     <a class="nav-link" href="#">홈 <span class="sr-only">(current)</span></a>
15                 </li>
16                 <li class="nav-item">
17                     <a class="nav-link" href="#">개시판</a>
18                 </li>
19                 <!-- <li class="nav-item">
20                     <a class="nav-link disabled" href="#" tabindex="-1" aria-disabled="true">Disabled</a>
21                 </li>
22                 <li class="nav-item dropdown">
23                     <a class="nav-link dropdown-toggle" href="#" id="dropdown01" data-toggle="dropdown"
24                         aria-haspopup="true" aria-expanded="false">Dropdown</a>
25                     <div class="dropdown-menu" aria-labelledby="dropdown01">
26                         <a class="dropdown-item" href="#">Action</a>
27                         <a class="dropdown-item" href="#">Another action</a>
28                         <a class="dropdown-item" href="#">Something else here</a>
29                     </div>
30                 </li> -->
31         </ul>
32     </div>
33 
```

← → ⌂ thymeleaf.org/doc/tutorials/3.0/usingthymeleaf.html#difference-between-thinsert-and-threplace-and-

7.1 Simple conditionals: "if" and "unless"

7.2 Switch statements

8 Template Layout

8.1 Including template fragments

Defining and referencing fragments

Fragment specification syntax

Referencing fragments without th:fragment

Difference between th:insert and th:replace (and th:include)

8.2 Parameterizable fragment signatures

Fragment local variables without fragment arguments

th:assert for in-template assertions

8.3 Flexible layouts: beyond mere fragment insertion

Using the empty fragment

Using the no-operation token

```
<footer th:fragment="copy">
    &copy; 2011 The Good Thymes Virtual Grocery
</footer>
```

...included three times in host <div> tags, like this:

```
<body>
    ...
    <div th:insert="footer :: copy"></div>
    <div th:replace="footer :: copy"></div>
    <div th:include="footer :: copy"></div>
</body>
```

...will result in:

```
<body>
    ...
    <div>
        <footer>
            &copy; 2011 The Good Thymes Virtual Grocery
        </footer>
    </div>
    ...
</body>
```

common.html # starter-template.css

index.html > html > body > nav.navbar.navbar-expand-md.navbar-dark.bg-dark.fixed-top

Required meta tags -->  
charset="utf-8"  
name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no"

Bootstrap CSS -->  
rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/css/bootstrap.min.css"  
integrity="sha384-9aIt2nRpC12Uk9gS9baD141NQApFmC26EwAOH8WgZl5MYxXffc+NcPb1dKGj7Sk" crossorigin="anonymous"  
href="starter-template.css" rel="stylesheet">

a>Hello, Spring Boot!</title>

class="navbar navbar-expand-md navbar-dark bg-dark fixed-top" th:replace="fragments/common :: menu"[]

a class="navbar-brand" href="#">Spring Boot Tutorial</a>

button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarsExampleDefault" aria-controls="navbarsExampleDefault" aria-expanded="false" aria-label="Toggle navigation">

<span class="navbar-toggler-icon"></span>

</button>

<div class="collapse navbar-collapse" id="navbarsExampleDefault">

22
23
24
25
26
27
28
29
30
31
32
33
34

templates &gt; index.html &gt; html &gt; body &gt; nav.navbar.navbar-expand-md.navbar-dark.bg-dark.fixed-top

```
13
14     <title>Hello, Spring Boot!</title>
15 </head>
16
17 <body>
18     <nav class="navbar navbar-expand-md navbar-dark bg-dark fixed-top" th:replace="fragments/common :: menu">
19         <!-- <a class="navbar-brand" href="#">Spring Boot Tutorial</a>
20         <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#naybarsExampleDefault"
21             aria-controls="naybarsExampleDefault" aria-expanded="false" aria-label="Toggle navigation">
22             <span class="navbar-toggler-icon"></span>
23         </button>
24
25         <div class="collapse navbar-collapse" id="naybarsExampleDefault">
26             <ul class="navbar-nav mr-auto">
27                 <li class="nav-item active">
28                     <a class="nav-link" href="#">홈 <span class="sr-only">(current)</span></a>
29                 </li>
30                 <li class="nav-item">
31                     <a class="nav-link" href="#">개시판</a>
32                 </li>
33             </ul>
34         </div> -->
35     </nav>
36
37     <main role="main" class="container">
38
39         <div class="starter-template">
40             <h1>Spring Boot Tutorial</h1>
41             <p class="lead">Spring Boot을 이용해 웹 페이지 제작<br>
42                 Spring Security, JPA를 이용해 보안 설정과 데이터 다루기 </p>
```

The screenshot shows a code editor interface with the following details:

- File Tabs:** index.html, list.html, common.html X, # starter-template.css
- Breadcrumbs:** templates > fragments > common.html > html > head
- Code Content:** A snippet of Thymeleaf template code for the head section of a web page.

```
1  <!DOCTYPE html>
2  <html xmlns:th="http://www.thymeleaf.org">
3  |     <head th:fragment="head">
4  |         <!-- Required meta tags -->
5  |         <meta charset="utf-8">
6  |         <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
7
8         <!-- Bootstrap CSS -->
9         <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/css/bootstrap.min.css"
10            integrity="sha384-9aIt2nRpC12Uk9gS9baDl411NQApFmC26EwAOH8WgZ15MYxFFc+NcPb1dKGj7Sk" crossorigin="an
11         <link href="starter-template.css" rel="stylesheet">
12
13         <title>Hello, Spring Boot!</title>
14     </head>
15
16     <body>
```

A red rectangular box highlights the entire content of the head section, starting from the opening `<head>` tag and ending at the closing `</head>` tag.

7.1 Simple conditionals: "if" and "unless"

7.2 Switch statements

## 8 Template Layout

8.1 Including template fragments

Defining and referencing fragments

Fragment specification syntax

Referencing fragments without th:fragment

Difference between th:insert and  
th:replace (and th:include)

## 8.2 Parameterizable fragment signatures

Fragment local variables without fragment  
arguments

th:assert for in-template assertions

8.3 Flexible layouts: beyond mere fragment  
insertion

Using the empty fragment

&lt;/body&gt;

## 8.2 Parameterizable fragment signatures

In order to create a more *function-like* mechanism for template fragments, fragments defined with `th:fragment` can specify a set of parameters:

```
<div th:fragment="frag (onevar,twovar)">
    <p th:text="${onevar} + ' - ' + ${twovar}">...</p>
</div>
```

This requires the use of one of these two syntaxes to call the fragment from `th:insert` or `th:replace`:

```
<div th:replace="::frag (${value1},${value2})">...</div>
<div th:replace="::frag (onevar=${value1},twovar=${value2})">...</div>
```

Note that order is not important in the last option:

```
<div th:replace="::frag (twovar=${value2},onevar=${value1})">...</div>
```

The screenshot shows a code editor interface with several tabs at the top: 'index.html', 'list.html', 'common.html X', and '# starter-template.css'. The current file is 'common.html'. The code itself is a Thymeleaf template:

```
1  <!DOCTYPE html>
2  <html xmlns:th="http://www.thymeleaf.org">
3  <head th:fragment="head(title)">
4      <!-- Required meta tags -->
5      <meta charset="utf-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
7
8      <!-- Bootstrap CSS -->
9      <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/css/bootstrap.min.css"
10         integrity="sha384-9aIt2nRpC12Uk9gS9baDl411NQApFmC26EwAOH8WgZ15MYYxFfc+NcPb1dKGj7Sk" crossorigin="an
11      <link href="starter-template.css" rel="stylesheet">
12
13      <title th:text="${title}">Hello, Spring Boot!</title>
14  </head>
15  <body>
16      <nav class="navbar navbar-expand-md navbar-dark bg-dark fixed-top" th:fragment="menu">
17          <a class="navbar-brand" href="#">Spring Boot Tutorial</a>
18          <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarsExampleDef
19              aria-controls="navbarsExampleDefault" aria-expanded="false" aria-label="Toggle navigation">
20                  <span class="navbar-toggler-icon"></span>
21          </button>
22
23          <div class="collapse navbar-collapse" id="navbarsExampleDefault">
24              <ul class="navbar-nav mr-auto">
25                  <li class="nav-item active">
26                      <a class="nav-link" href="#">首頁 <span class="sr-only">(current)</span></a>
27                  </li>
28                  <li class="nav-item">
29                      <a class="nav-link" href="#">개시판</a>
30                  </li>
```

Two specific sections of the code are highlighted with red boxes: the head fragment declaration at line 3 and the title element at line 13.

```
index.html      list.html  X  common.html  # starter-template.css
templates > board > list.html > html > head
1  <!doctype html>
2  <html xmlns:th="http://www.thymeleaf.org">
3
4  <head th:replace="fragments/common :: head('게시판')">
5      
6      <meta charset="utf-8">
7      <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
8
9      
10     <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/css/bootstrap.min.css"
11         integrity="sha384-9aIt2nRpC12Uk9gS9baD1411NQApFmC26EwAOH8WgZl5MYYxFfc+NcPb1dKGj7Sk" crossorigin="ar
12     <link href="starter-template.css" th:href="@{/starter-template.css}" rel="stylesheet">
13
14     <title>게시판</title>
15 </head>
16
17 <body>
18     <nav class="navbar navbar-expand-md navbar-dark bg-dark fixed-top" th:replace="fragments/common :: menu">
19         <!-- <a class="navbar-brand" href="#">Spring Boot Tutorial</a>
20         <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarsExampleDefault"
21             aria-controls="navbarsExampleDefault" aria-expanded="false" aria-label="Toggle navigation">
22             <span class="navbar-toggler-icon"></span>
23         </button>
24
25         <div class="collapse navbar-collapse" id="navbarsExampleDefault">
26             <ul class="navbar-nav mr-auto">
27                 <li class="nav-item active">
28                     <a class="nav-link" href="#"> 게시판 <span class="sr-only">(current)</span></a>
29                 </li>
30                 <li class="nav-item">
```

```
index.html X listhtml common.html # starter-template.css

templates > index.html > html > head
1  <!doctype html>
2  <html xmlns:th="http://www.thymeleaf.org">
3
4  <head th:replace="fragments/common :: head['Hello, Spring Boot!']">
5  |   <!-- Required meta tags -->
6  |   <meta charset="utf-8">
7  |   <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
8
9  |   <!-- Bootstrap CSS -->
10 |   <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/css/bootstrap.min.css"
11 |       integrity="sha384-9aIt2nRpC12Uk9gS9baD141NQApFmC26EwAOH8WgZl5MYxFfc+NcPb1dKGj7Sk" crossorigin="ar
12 |   <link href="starter-template.css" rel="stylesheet">
13
14     <title>Hello, Spring Boot!</title>
15 </head>
16
17 <body>
18     <nav class="navbar navbar-expand-md navbar-dark bg-dark fixed-top" th:replace="fragments/common :: menu">
19     |   <!-- <a class="navbar-brand" href="#">Spring Boot Tutorial</a>
20     |   <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarsExampleDefault"
21     |       aria-controls="navbarsExampleDefault" aria-expanded="false" aria-label="Toggle navigation">
22     |       <span class="navbar-toggler-icon"></span>
23     |   </button>
24
25     |   <div class="collapse navbar-collapse" id="navbarsExampleDefault">
26     |       <ul class="navbar-nav mr-auto">
27     |           <li class="nav-item active">
28     |               <a class="nav-link" href="#"> (current)</span></a>
29     |           </li>
30           <li class="nav-item">
```

[index.html](#)[list.html](#)[common.html X](#)[# starter-template.css](#)

...

```
vbar-expand-md.navbar-dark.bg-dark.fixed-top > div#navbarsExampleDefault.collapse.navbar-collapse > ul.navbar-nav.mr-auto > li.nav-item > a.nav-link
17     <a class="navbar-brand" href="#">Spring Boot Tutorial</a>
18     <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarsExampleDef"
19         aria-controls="navbarsExampleDefault" aria-expanded="false" aria-label="Toggle navigation">
20         <span class="navbar-toggler-icon"></span>
21     </button>

22
23     <div class="collapse navbar-collapse" id="navbarsExampleDefault">
24         <ul class="navbar-nav mr-auto">
25             <li class="nav-item active">
26                 <a class="nav-link" href="#">(current)<span class="sr-only">(current)</span></a>
27             </li>
28             <li class="nav-item">
29                 <a class="nav-link" href="#">개시판</a>
30             </li>
31             <!-- <li class="nav-item">
32                 <a class="nav-link disabled" href="#" tabindex="-1" aria-disabled="true">Disabled</a>
33             </li>
34             <li class="nav-item dropdown">
35                 <a class="nav-link dropdown-toggle" href="#" id="dropdown01" data-toggle="dropdown"
36                     aria-haspopup="true" aria-expanded="false">Dropdown</a>
37                 <div class="dropdown-menu" aria-labelledby="dropdown01">
38                     <a class="dropdown-item" href="#">Action</a>
39                     <a class="dropdown-item" href="#">Another action</a>
40                     <a class="dropdown-item" href="#">Something else here</a>
41                 </div>
42             </li> -->
43         </ul>
44         <!-- <form class="form-inline my-2 my-lg-0">
45             <input class="form-control mr-sm-2" type="text" placeholder="Search" aria-label="Search">
46             <button class="btn btn-secondary my-2 my-sm-0" type="submit">Search</button>
```

← → ⌂ thymeleaf.org/doc/tutorials/3.0/usingthymeleaf.html#difference-between-thinsert-and-threplace-and-thinclude  
7.1 Simple conditionals: "if" and "unless"

7.2 Switch statements

8 Template Layout

8.1 Including template fragments

Defining and referencing fragments

Fragment specification syntax

Referencing fragments without th:fragment

Difference between th:insert and  
th:replace (and th:include)

8.2 Parameterizable fragment signatures

Fragment local variables without fragment  
arguments

th:assert for in-template assertions

8.3 Flexible layouts: beyond mere fragment  
insertion

Using the empty fragment

Using the no-operation token

For example, you might want to store the name of a CSS class to be added (not set, just added) to one of your buttons in a context variable, because the specific CSS class to be used would depend on something that the user did before:

```
<input type="button" value="Do it!" class="btn" th:attrappend="class=${' ' + cssStyle}" />
```

If you process this template with the `cssStyle` variable set to "warning", you will get:

```
<input type="button" value="Do it!" class="btn warning" />
```

There are also two specific *appending attributes* in the Standard Dialect: the `th:classappend` and `th:styleappend` attributes, which are used for adding a CSS class or a fragment of `style` to an element without overwriting the existing ones:

```
<tr th:each="prod : ${prods}" class="row" th:classappend="${prodStat.odd} ? 'odd'">
```

(Don't worry about that `th:each` attribute. It is an *iterating attribute* and we will talk about it later.)

## 5.5 Fixed-value boolean attributes

HTML has the concept of *boolean attributes*, attributes that have no value and the presence of one means that

```
index.html      list.html      common.html X # starter-template.css
```

.navbar.navbar-expand-md.navbar-dark.bg-dark.fixed-top > div#navbarsExampleDefault.collapse.navbar-collapse > ul.navbar-nav.mr-auto > li.r

```
13     <title th:text="${title}">Hello, Spring Boot!</title>
14   </head>
15   <body>
16     <nav class="navbar navbar-expand-md navbar-dark bg-dark fixed-top" th:fragment="menu">
17       <a class="navbar-brand" href="#">Spring Boot Tutorial</a>
18       <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarsExampleDef"
19           aria-controls="navbarsExampleDefault" aria-expanded="false" aria-label="Toggle navigation">
20           <span class="navbar-toggler-icon"></span>
21       </button>
22
23       <div class="collapse navbar-collapse" id="navbarsExampleDefault">
24         <ul class="navbar-nav mr-auto">
25           <li class="nav-item active" th:classappend="${prodStat.odd} ? 'odd'">
26             <a class="nav-link" href="#">홈 <span class="sr-only">(current)</span></a>
27           </li>
28           <li class="nav-item">
29             <a class="nav-link" href="#">게시판</a>
30           </li>
31           <li class="nav-item">
```

nav.navbar.navbar-expand-md.navbar-dark.bg-dark.fixed-top > div#navbarsExampleDefault.collapse.navbar-collapse > ul.navbar-nav.mr-auto

```
13     <title th:text="${title}">Hello, Spring Boot!</title>
14   </head>
15   <body>
16     <nav class="navbar navbar-expand-md navbar-dark bg-dark fixed-top" th:fragment="menu(menu)">
17       <a class="navbar-brand" href="#">Spring Boot Tutorial</a>
18       <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarsExampleDef"
19           aria-controls="navbarsExampleDefault" aria-expanded="false" aria-label="Toggle navigation">
20           <span class="navbar-toggler-icon"></span>
21       </button>
22
23       <div class="collapse navbar-collapse" id="navbarsExampleDefault">
24         <ul class="navbar-nav mr-auto">
25           <li class="nav-item active" th:classappend="${menu} == 'home' ? 'active'">
26             <a class="nav-link" href="#">홈 <span class="sr-only">(current)</span></a>
27           </li>
28           <li class="nav-item" th:classappend="${menu} == 'board' ? 'active'">
29             <a class="nav-link" href="#">게시판</a>
30           </li>
```

```
index.html      list.html      common.html X # starter-template.css
avbar-dark.bg-dark.fixed-top > div#navbarsExampleDefault.navbar-collapse.navbar-collapse > ul.navbar-nav.mr-auto > li.nav-item > a.nav-link >
13   :h:text="${title}">Hello, Spring Boot!</title>
14
15
16   :ss="navbar navbar-expand-md navbar-dark bg-dark fixed-top" th:fragment="menu(menu)">
17     :lass="navbar-brand" href="#">Spring Boot Tutorial</a>
18     :ton class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarsExampleDefault"
19       aria-controls="navbarsExampleDefault" aria-expanded="false" aria-label="Toggle navigation">
20       <span class="navbar-toggler-icon"></span>
21     :ton>
22
23   / class="collapse navbar-collapse" id="navbarsExampleDefault">
24     <ul class="navbar-nav mr-auto">
25       <li class="nav-item" th:classappend="${menu} == 'home'? 'active'">
26         <a class="nav-link" href="#">홈 <span class="sr-only" th:if="${menu} == 'home'">(current)</span></a>
27       </li>
28       <li class="nav-item" th:classappend="${menu} == 'board'? 'active'">
29         <a class="nav-link" href="#">게시판 <span class="sr-only" th:if="${menu} == 'board'">(current)</span></a>
30       </li>
--
```

```
index.html      list.html      common.html X  # starter-template.css
nav.navbar.navbar-expand-md.navbar-dark.bg-dark.fixed-top > div#navbarsExampleDefault.collapse.navbar-collapse > ul.navbar-nav.mr-auto >
13   :h:text="${title}">Hello, Spring Boot!</title>
14
15
16   iss="navbar navbar-expand-md navbar-dark bg-dark fixed-top" th:fragment="menu(menu)">
17     :lass="navbar-brand" href="#">Spring Boot Tutorial</a>
18     :ton class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarsExampleDefault"
19       aria-controls="navbarsExampleDefault" aria-expanded="false" aria-label="Toggle navigation">
20         <span class="navbar-toggler-icon"></span>
21       :tton>
22
23   / class="collapse navbar-collapse" id="navbarsExampleDefault">
24     <ul class="navbar-nav mr-auto">
25       <li class="nav-item" th:classappend="${menu} == 'home'? 'active'">
26         <a class="nav-link" href="#" th:href="@{/}">[Redacted] <span class="sr-only" th:if="${menu} == 'home'">(current)</span>
27       </li>
28       <li class="nav-item" th:classappend="${menu} == 'board'? 'active'">
29         <a class="nav-link" href="#" th:href="@{/board/list}">[Redacted] <span class="sr-only" th:if="${menu} == 'board'">(current)</span>
30       </li>
31       <!-- <li class="nav-item">
```

## MariaDB Foundation



# MariaDB Foundation

## MariaDB is free and open source software

The MariaDB database server is published as free and open source software under the General Public License version 2. You can download and use it as much as you want free of charge. *All use of the binaries from mariadb.org is at your own risk as stated in the GPLv2.* While we do our best to make the world's best database software, the MariaDB Foundation does not provide any guarantees

## Downloads

 Source, Binaries, and Packa

### MariaDB

Thank you, your download will begin shortly. If it fails to start, click [here](#).  
If the mirror is down or returns 'file not found', click [here](#) to try a differ

### Donate to the MariaDB Foundation

The MariaDB Foundation welcomes donations large and small. Ever development of MariaDB. In case you wonder how much to donate a <https://mariadb.org/donate/> for more information.

You will be redirected to the [MariaDB Knowledge Base](#) once your



User settings



## Default instance properties

MariaDB 10.4 (x64) database configuration

MariaDB  
Server 

Modify password for database user 'root'

New root password:

\*\*\*\*\*

Enter new root password

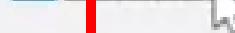
Confirm:

\*\*\*\*\*

Retype the password

Enable access from remote machines for 'root' user

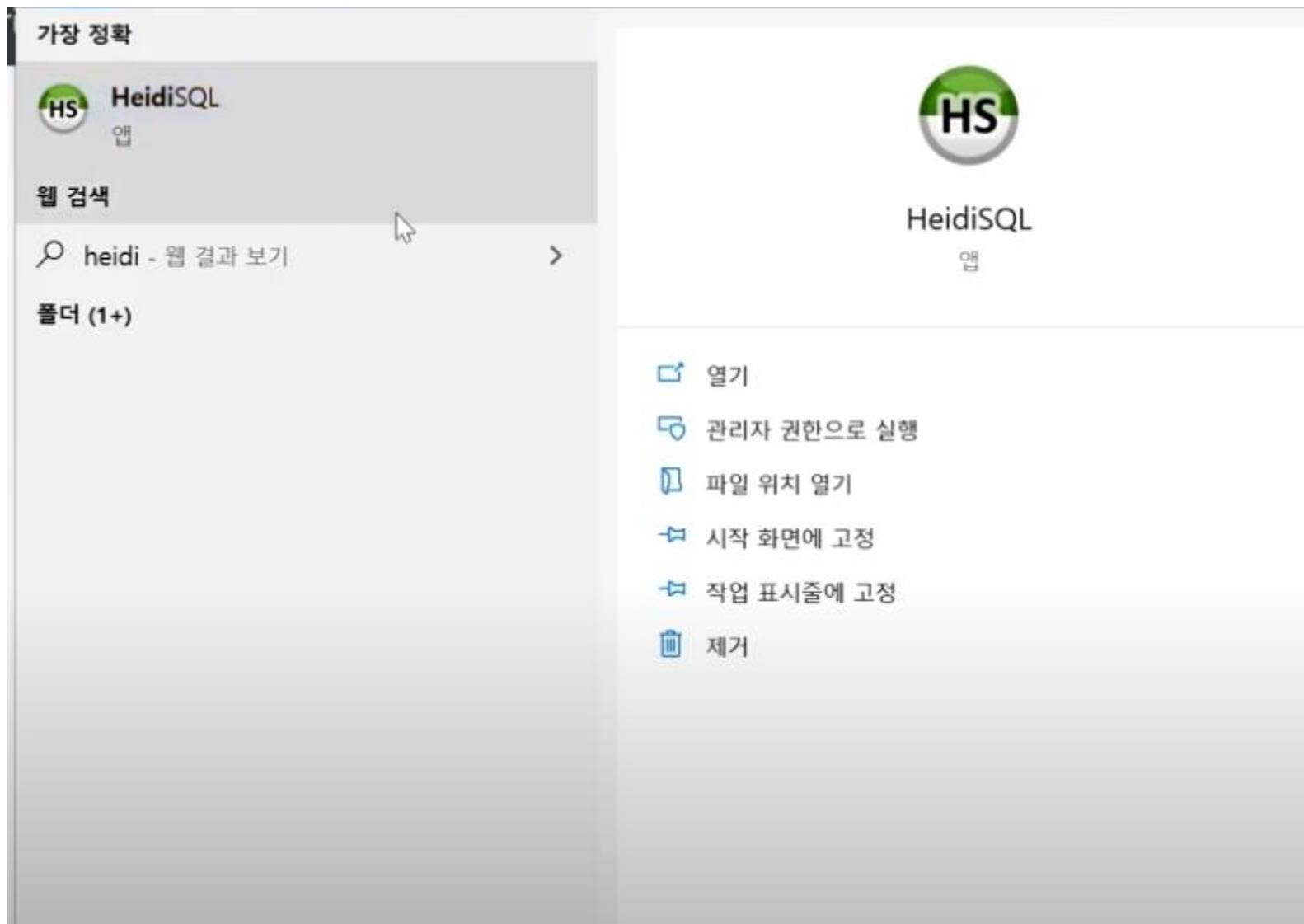
Use UTF8 as default server's character set



Back

Next

Cancel





## 세션 관리자

? X

Filter ...

세션 이름 ^	호스트
LocalDB *	127...

< >

+ 신규 | 저장 | 삭제

설정 고급 통계

네트워크 유형: MySQL (TCP/IP)

Library: libmariadb.dll

호스트명 / IP: 127.0.0.1

자격 증명 프롬프트

Windows 인증 사용

사용자: root

암호: ●●●●●●●●●|

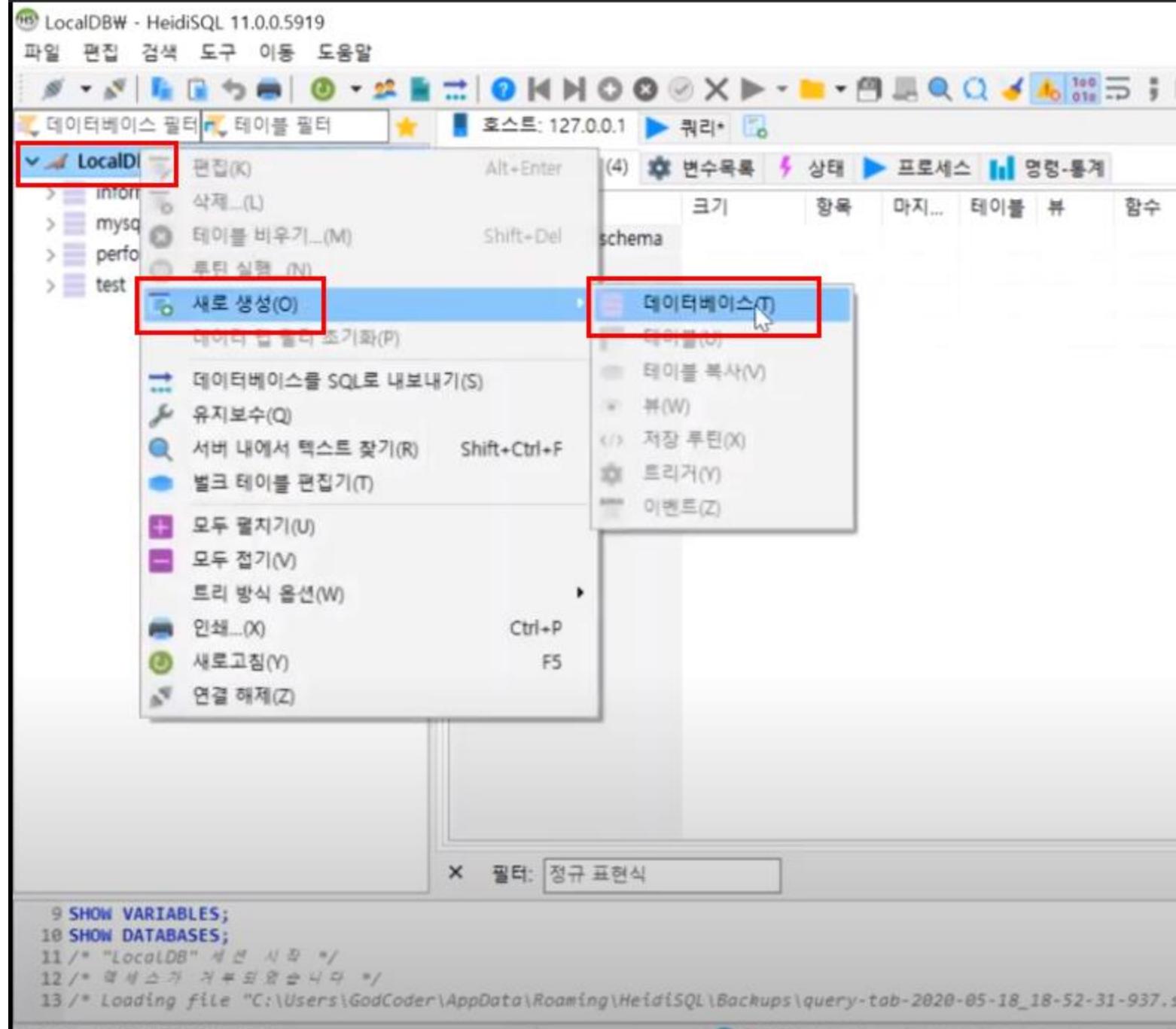
포트: 3306

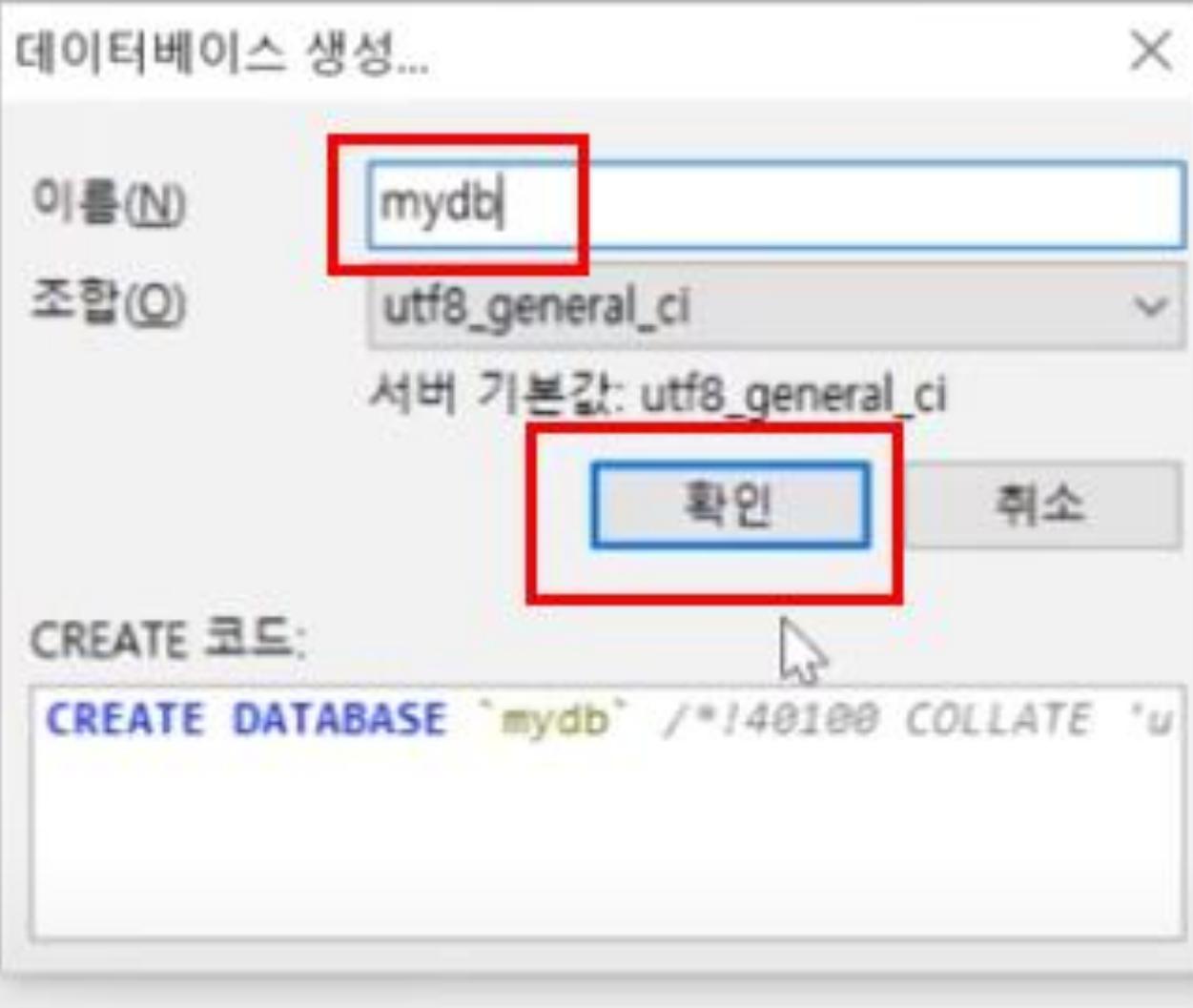
압축된 클라이언트/서버 프로토콜

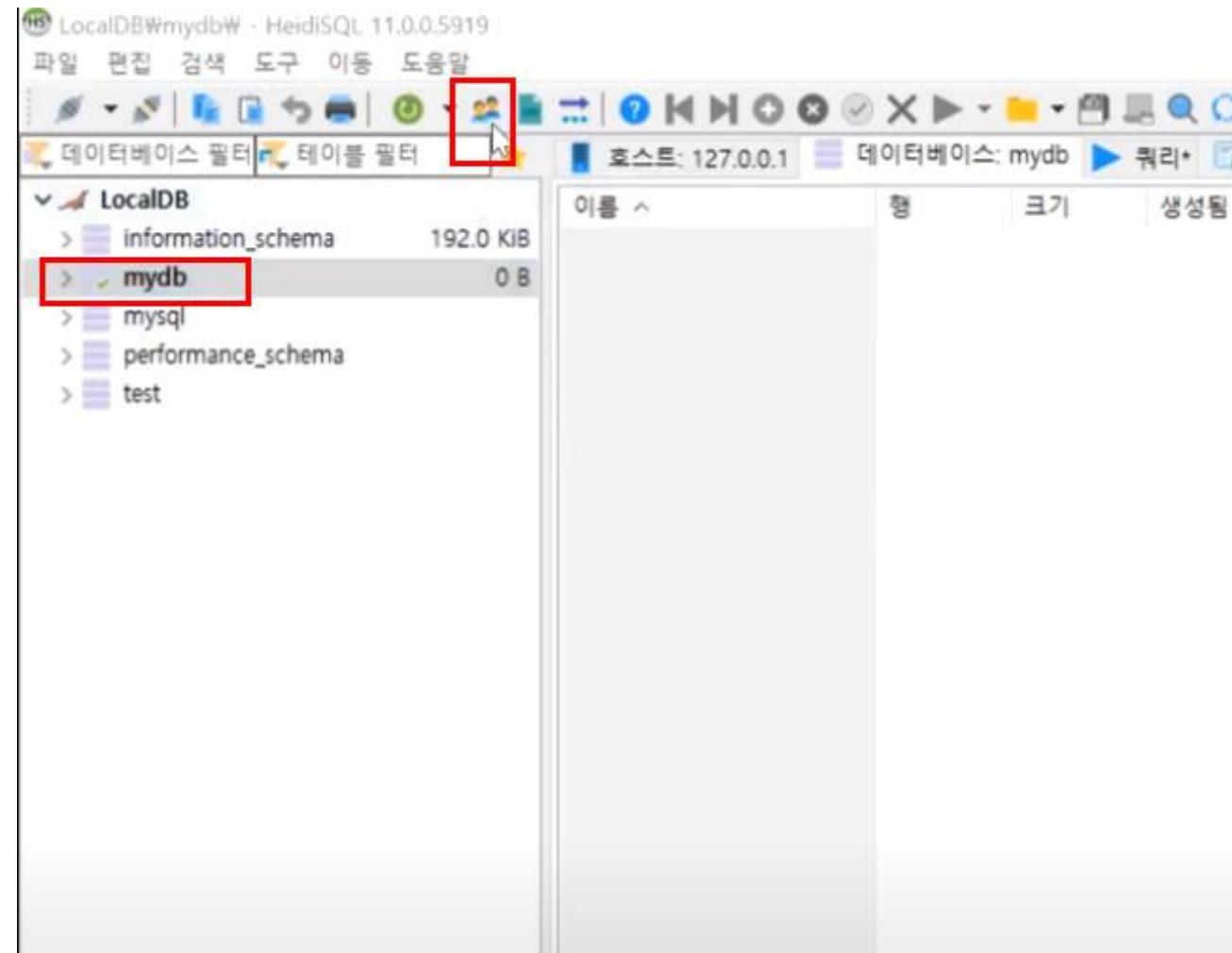
데이터베이스: 세미콜론으로 구분

코멘트:

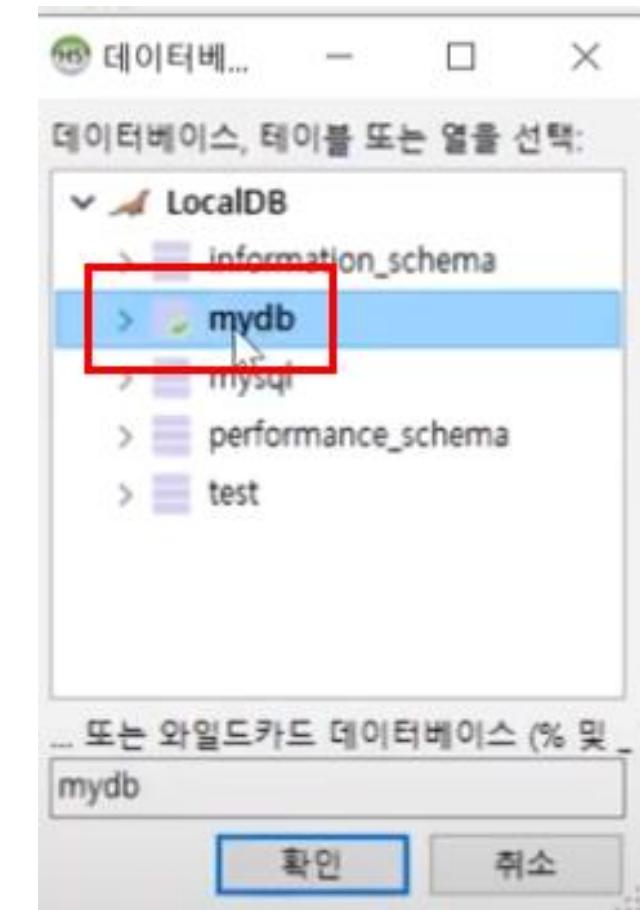
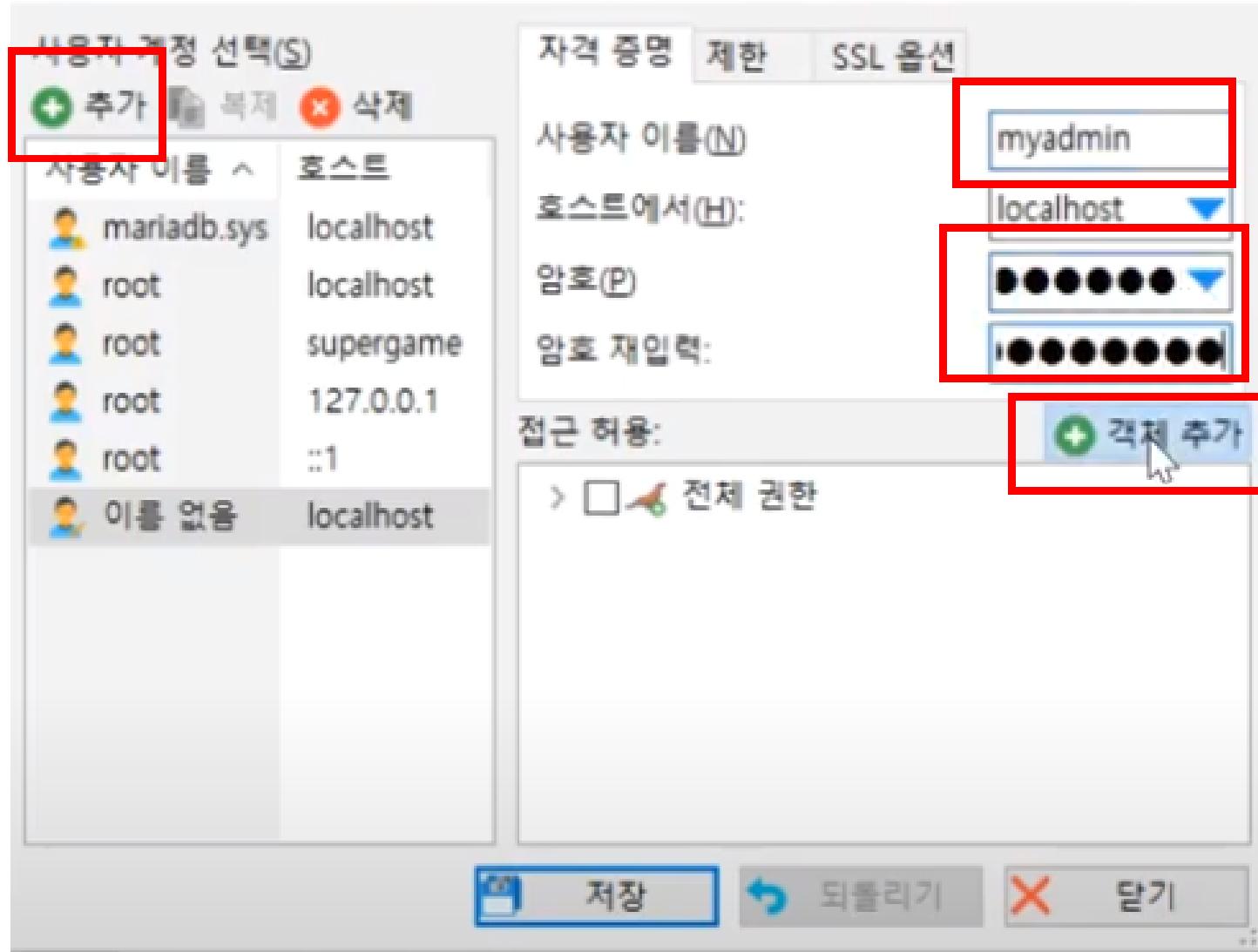
열기 취소 더 보기 |▼







## 16 사용자 관리자 - 이름 없음



## ⑯ 사용자 관리자 - 이름 없음

사용자 계정 선택(S)

+ 추가 ━ 삭제 ━ 삭제

사용자 이름 ^	호스트
mariadb.sys	localhost
root	localhost
root	supergame
root	127.0.0.1
root	::1
이름 없음	localhost

자격 증명 제한 SSL 옵션

사용자 이름(N) myadmin  
호스트에서(H) localhost  
암호(P)   
암호 재입력:   
접근 허용:

> 전체 권한

데이터베이스: mydb

EXECUTE  
 SELECT  
 SHOW VIEW  
 ALTER  
 ALTER ROUTINE

저장 되돌리기 닫기

## ⑯ 사용자 관리자 - 이름 없음

사용자 계정 선택(S)

+ 추가 ━ 삭제 ━ 삭제

사용자 이름 ^	호스트
mariadb.sys	localhost
root	localhost
root	supergame
root	127.0.0.1
root	::1
이름 없음	localhost

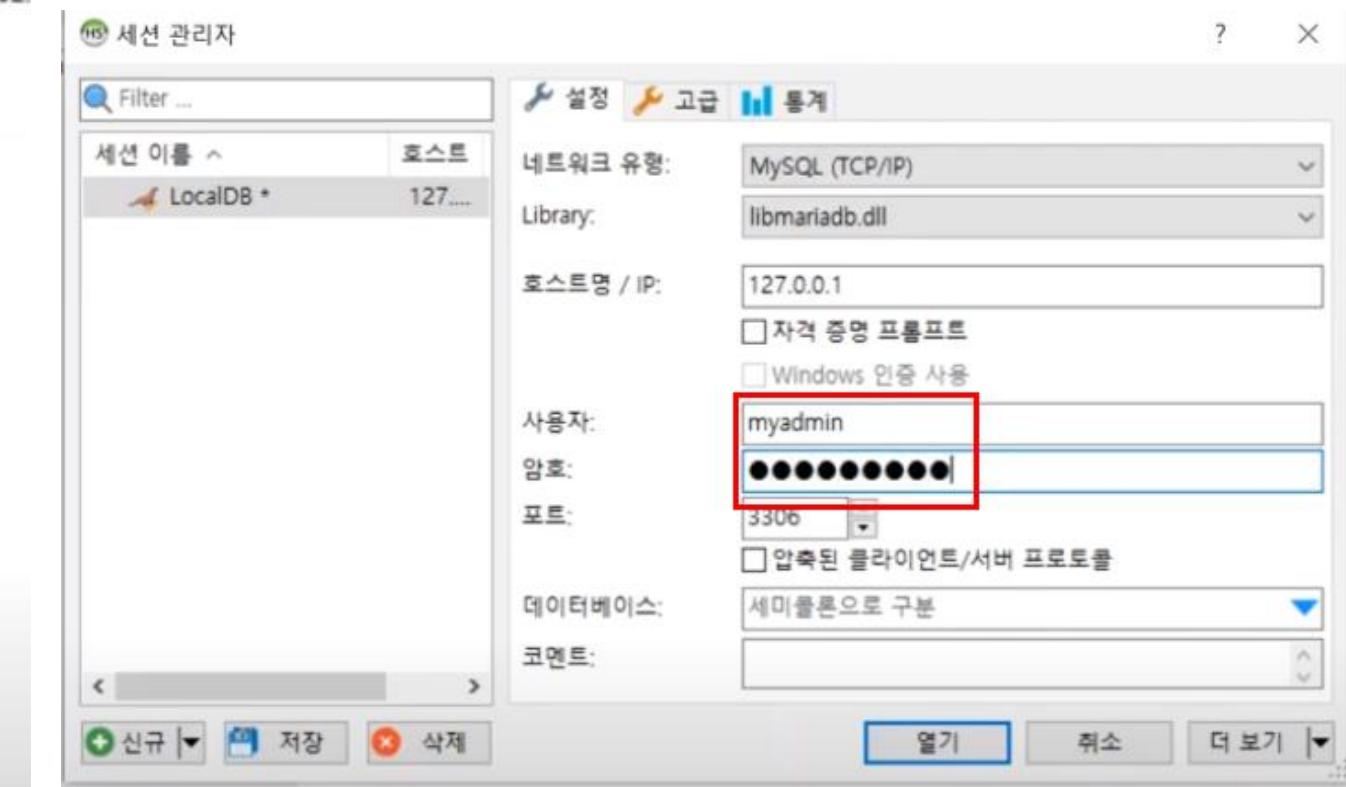
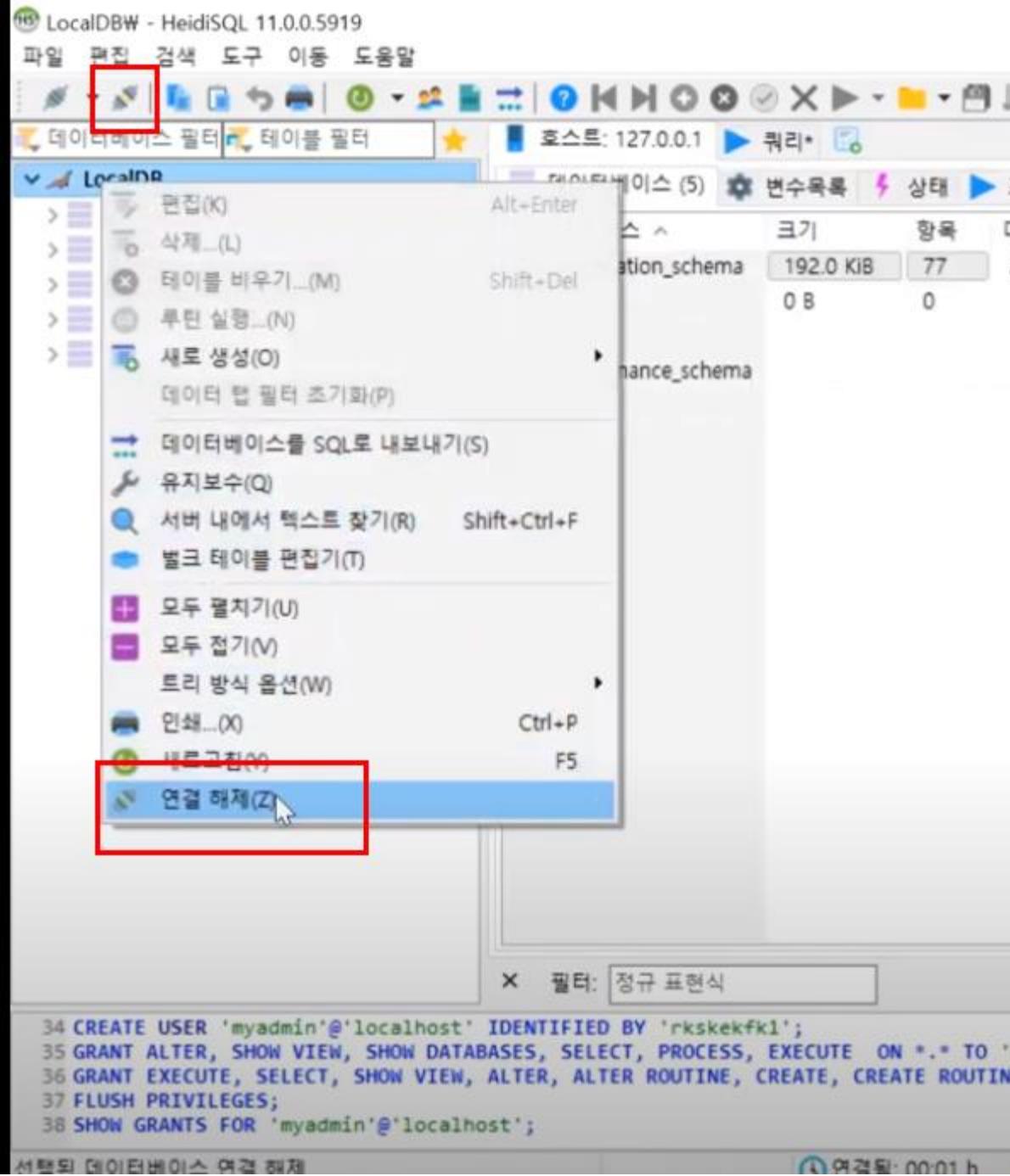
자격 증명 제한 SSL 옵션

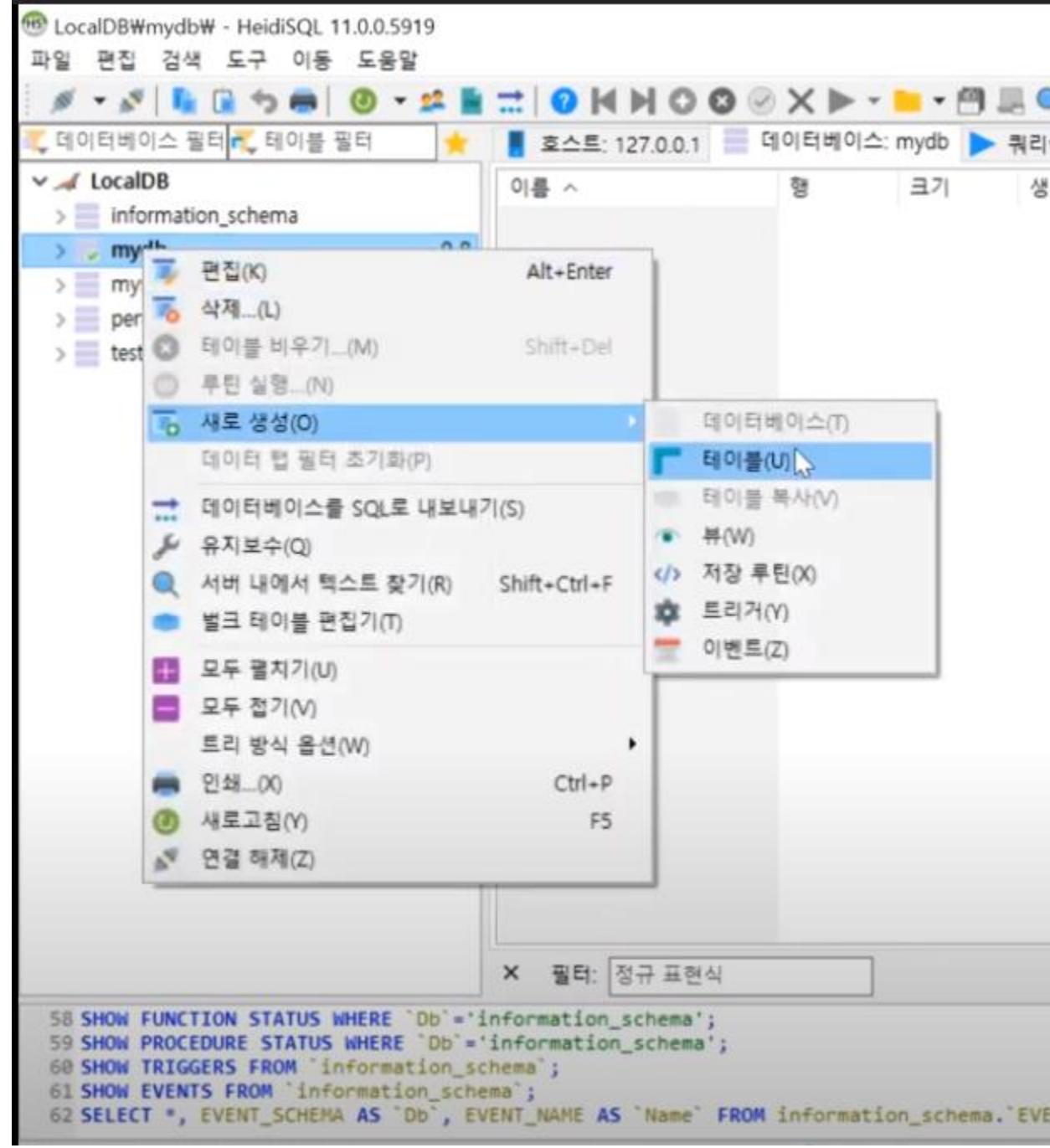
사용자 이름(N) myadmin  
호스트에서(H) localhost  
암호(P)   
암호 재입력:   
접근 허용:

전체 권한

EXECUTE  
 PROCESS  
 SELECT  
 SHOW DATABASES  
 SHOW VIEW  
 ALTER

저장 되돌리기 닫기





파일 편집 검색 도구 이동 도움말

데이터베이스 필터 테이블 필터 ★

호스트: 127.0.0.1 데이터베이스: mydb 테이블: [U]

LocalDB

- information\_schema 192.0 KiB
- mydb** 0 B
- mysql
- performance\_schema
- test

기본 옵션 인덱스 외래 키 분할 </>

이름: **board**

코멘트: 게시판

열: **추가** 제거 위로 아래로

#	이름	데이터 유형	길이
			0

열 추가

The screenshot shows the MySQL Workbench interface. On the left, a tree view displays the LocalDB schema with several databases listed. The 'mydb' database is selected and highlighted with a red box. In the main workspace, a table creation dialog is open. The '이름:' field contains the value 'board', which is also highlighted with a red box. Below the table definition, a toolbar has a green '+' icon labeled '추가' (Add), which is also highlighted with a red box. The bottom part of the screen shows a table structure with columns for '#', '이름', '데이터 유형', and '길이', with one row currently present.

LocalDB

- information\_schema 192.0 KiB
- mvldb 16.0 KiB
- board** 16.0 KiB
- mysql
- performance\_schema
- test

기본 옵션 인덱스 외래 키 분할 CREATE 코드 ALTER 코드

추가 제거 청소 위로 아래로

이름: PRIMARY KEY id

#	이름	데이터 유형	길이/설정	부호 ...	NULL 허용	0으로 재설정	기본값
1	<b>id</b>	<b>BIGINT</b>	20	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<b>AUTO_INCREMENT</b>
2	title	VARCHAR	50	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	"
3	content	TEXT	65535	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	"

도움말 되돌리기 저장



jpa



All

Images

News

Videos

Books

More

Settings

Tools

About 24,300,000 results (0.46 seconds)

[spring.io](#) › [projects](#) › [spring-data-jpa](#) ▾

## Spring Data JPA

This module deals with enhanced support for **JPA** based data access layers. It makes it easier to build Spring-powered applications that use data access ...

[docs.spring.io](#) › [jpa](#) › [docs](#) › [current](#) › [reference](#) › [html](#) ▾

## Spring Data JPA - Reference Documentation

May 12, 2020 - Spring Data JPA 1.10 added the following features: Support for Projections in repository query methods. Support for Query by Example. The ...

[Project Metadata](#) · [Dependencies](#) · [Working with Spring Data](#) ... · [JPA Repositories](#)

hibernate



All

Images

Videos

Books

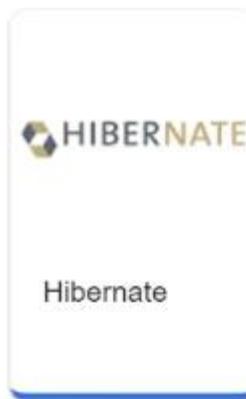
News

More

Settings

Tools

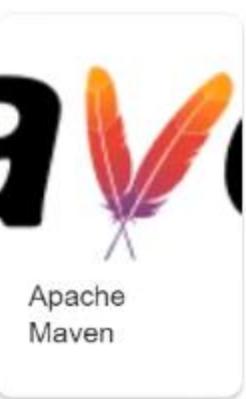
## Related to Hibernate and Java Persistence API



Hibernate



Spring  
Framework



Apache  
Maven



Java  
Persistence  
API

hibernate.org ▾

### Hibernate. Everything data. - Hibernate

More than an ORM, discover the **Hibernate** galaxy. **Hibernate** ORM. Domain model persistence for relational databases. More ...

You visited this page on 5/17/20.

The screenshot shows a Java code editor with three tabs: MyhomeApplication.java, BoardController.java, and application.properties. The BoardController.java tab is active and highlighted with a red box at the top. The code within the tab is also highlighted with a red box around the annotations @Controller and @RequestMapping. The code is as follows:

```
1 package com.godcoder.myhome.controller;
2
3 import org.springframework.stereotype.Controller;
4 import org.springframework.web.bind.annotation.GetMapping;
5 import org.springframework.web.bind.annotation.RequestMapping;
6
7 @Controller
8 @RequestMapping("/board")
9 public class BoardController {
10
11     @GetMapping("/list")
12     public String list() {
13         return "board/list";
14     }
15 }
```

A status bar at the bottom of the editor shows the text "BoardController > list()".



jpa maven

X

All

Videos

Images

News

Maps

More

Setl

About 5,110,000 results (0.37 seconds)

[mvnrepository.com](#) › tags › jpa ▾

## [jpa - Maven Repository](#)

Clean-room definition of **JPA APIs** intended for use in developing Hibernate **JPA** implemen

See README.md for details. Last Release on Jan 23, 2018 ...

You visited this page on 5/18/20.

[mvnrepository.com](#) › artifact › spring-data-jpa ▾

## [org.springframework.data](#) » [spring-data-jpa](#) - Maven Repository

Spring Data module for **JPA** repositories. License, Apache 2.0. Tags, dataspringpersistenc

Used By, 512 artifacts ...

Used By: 482 artifacts Tags: dataspringpersistencjpa

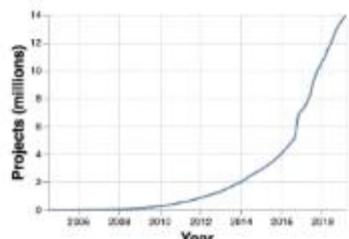
[mvnrepository.com](#) › artifact › hibernate-jpa-2 ▾

## [org.hibernate](#).javavx.persistence - Maven Repository

← → Cmvnrepository.com/tags/jpa

Search for groups, artifacts, categories

## Indexed Artifacts (17.1M)



## Popular Categories

- Aspect Oriented
- Actor Frameworks
- Application Metrics
- Build Tools
- Bytecode Libraries
- Command Line Parsers
- Cache Implementations
- Cloud Computing
- Code Analyzers
- Collections
- Configuration Libraries

## jpa

Sort: popular | newest1. **Hibernate EntityManager Relocation**  
org.hibernate » hibernate-entitymanagerHibernate ORM 6.0.0.Alpha5 release. See <http://hibernate.org/orm/releasenotes/6.0.0.Alpha5>

Last Release on Apr 24, 2020

2. **Spring Boot Starter Data JPA**  
org.springframework.boot » spring-boot-starter-data-jpa

Starter for using Spring Data JPA with Hibernate

Last Release on May 15, 2020

3. **JPA 2.0 API**  
org.hibernate.javax.persistence » hibernate-jpa-2.0-api

Hibernate definition of the Java Persistence 2.0 (JSR 317) API.

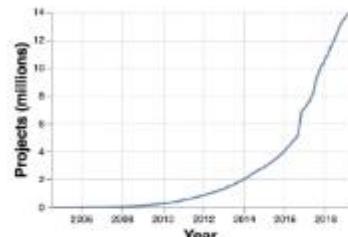
Last Release on Jun 15, 2011



Search for groups, artifacts, categories

Search

## Indexed Artifacts (17.1M)



## Popular Categories

- Aspect Oriented
- Actor Frameworks
- Application Metrics
- Build Tools
- Bytecode Libraries
- Command Line Parsers
- Cache Implementations
- Cloud Computing
- Code Analyzers
- Collections
- Configuration Libraries
- Core Utilities
- Date and Time Utilities

## Spring Boot Starter Data JPA » 2.3.0.RELEASE



Starter for using Spring Data JPA with Hibernate

License	Apache 2.0
Organization	Pivotal Software, Inc.
HomePage	<a href="https://spring.io/projects/spring-boot">https://spring.io/projects/spring-boot</a>
Date	(May 15, 2020)
Files	<a href="#">jar (4 KB)</a> View All
Repositories	Central
Used By	1,107 artifacts

Maven

Gradle

SBT

Ivy

Grape

Leiningen

Buildr

```
<!-- https://mvnrepository.com/artifact/org.springframework.boot/spring-boot-starter-data-jpa -->
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-data-jpa</artifactId>
    <version>2.3.0.RELEASE</version>
</dependency>
```

 Include comment with link to declaration

Copied to clipboard!

← → C google.com/search?q=mariadb+maven&oq=mariadb+maven&aqs=chrome..69i57j69i59j35i39j0l3

**mariadb maven**

All News Videos Maps Shopping More Settings

About 1,310,000 results (0.39 seconds)

[mvnrepository.com](#) › artifact › mariadb-java-client ▾

[org.mariadb.jdbc » mariadb-java-client - Maven Repository](#)

... Mapping · PDF Libraries · Top Categories · Home » org.mariadb.jdbc » mariadb-java-clien  
MariaDB Java Client. JDBC driver for MariaDB and MySQL ...

Used By: 371 artifacts License: LGPL 2.1

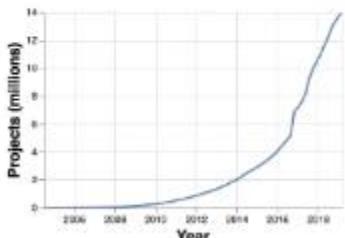
You visited this page on 5/18/20.

<b>1.1.7</b> MariaDB Java Client » 1.1.7. JDBC driver for MariaDB and ...	<b>Clojars (1)</b> MariaDB Java Client. JDBC driver for MariaDB and MySQL ...
<b>2.4.3</b> MariaDB Java Client » 2.4.3. JDBC driver for MariaDB and ...	<b>1.2.0</b> MariaDB Java Client » 1.2.0. JDBC driver for MariaDB and ...



Search for groups, artifacts, categories

Indexed Artifacts (17.1M)



## Popular Categories

- Aspect Oriented
- Actor Frameworks
- Application Metrics
- Build Tools
- Bytecode Libraries
- Command Line Parsers
- Cache Implementations
- Cloud Computing
- Code Analyzers
- Collections
- Configuration Libraries
- Core Utilities

## MariaDB Java Client » 2.6.0



JDBC driver for MariaDB and MySQL

License	LGPL 2.1
Categories	MySQL Drivers
Organization	mariadb.com
HomePage	<a href="https://mariadb.com/kb/en/mariadb/about-mariadb-connection-drivers/">https://mariadb.com/kb/en/mariadb/about-mariadb-connection-drivers/</a>
Date	(Mar 19, 2020)
Files	<a href="#">jar (605 KB)</a> View All
Repositories	Central
Used By	402 artifacts



Maven

Gradle

SBT

Ivy

Grape

Leiningen

Buildr

```
<!-- https://mvnrepository.com/artifact/org.mariadb.jdbc/mariadb-java-client -->
<dependency>
    <groupId>org.mariadb.jdbc</groupId>
    <artifactId>mariadb-java-client</artifactId>
    <version>2.6.0</version>
</dependency>
```

The screenshot shows a Java IDE interface with several tabs at the top: MyhomeApplication.java, BoardController.java, pom.xml (which is selected and highlighted with a red box), and application.properties. The main content area displays the pom.xml file's XML code. A red box highlights the entire `<dependency>` block for the MariaDB JDBC driver, starting from line 34 and ending at line 67. The code within this block is as follows:

```
<dependency>
    <groupId>org.mariadb.jdbc</groupId>
    <artifactId>mariadb-java-client</artifactId>
    <version>2.6.0</version>
</dependency>
```

About 569,000 results (0.41 seconds)

[springframework.guru](#) > configuring-spring-boot-for-...

## Configuring Spring Boot for MariaDB - Spring Framework Guru

Feb 6, 2017 - So, when we configure MariaDB for use, Spring Boot won't set up the H2

database anymore. The following properties are required to configure ...

You visited this page on 5/18/20.

[www.leafcats.com](#) > ... ▾ [Translate this page](#)

## Spring boot mariadb 연동 - 복세편살 개발라이프

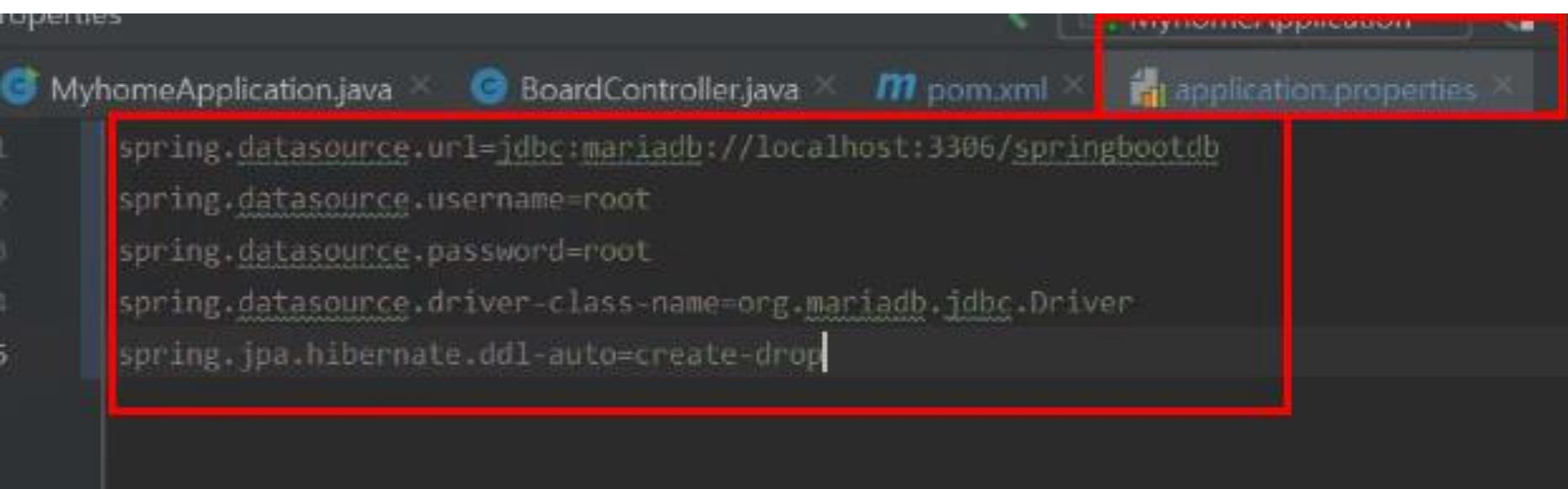
Nov 14, 2018 - Hibernate와 Mybatis 사이에서 한참을 고민하고 있는 중이다. 둘의 차이점과 장단  
점은 추후 포스팅 하기로 하고, 일단 Spring boot에 mariadb를 ...

[stackoverflow.com](#) > [questions](#) > [spring-boot](#)

Configure Spring Boot to use MariaDB as the database

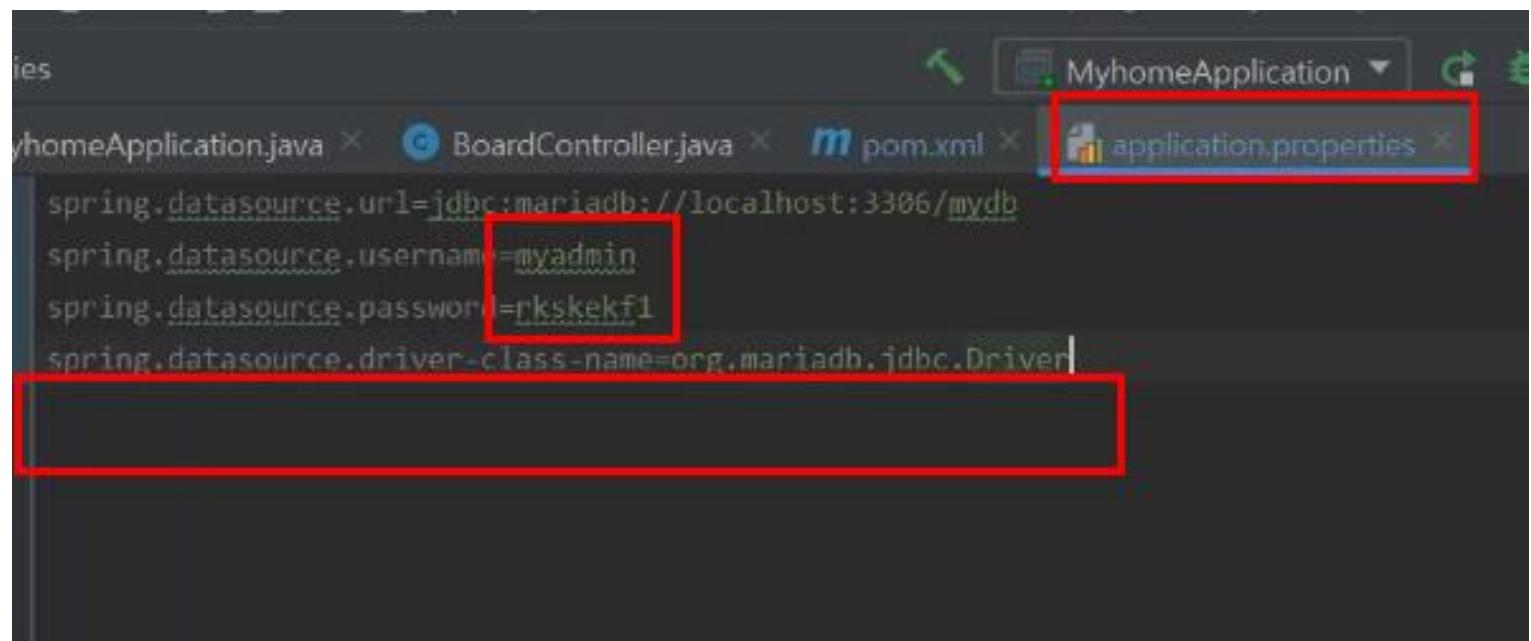
```
spring.datasource.url=jdbc:mariadb://localhost:3306/springbootdb
spring.datasource.username=root
spring.datasource.password=root
spring.datasource.driver-class-name=org.mariadb.jdbc.Driver
spring.jpa.hibernate.ddl-auto=create-drop
```





application.properties

```
spring.datasource.url=jdbc:mariadb://localhost:3306/springbootdb
spring.datasource.username=root
spring.datasource.password=root
spring.datasource.driver-class-name=org.mariadb.jdbc.Driver
spring.jpa.hibernate.ddl-auto=create-drop
```



application.properties

```
spring.datasource.url=jdbc:mariadb://localhost:3306/mydb
spring.datasource.username=myadmin
spring.datasource.password=rkskekf1
spring.datasource.driver-class-name=org.mariadb.jdbc.Driver
```

The screenshot shows a Java IDE interface with three tabs at the top: 'MyhomeApplication.java', 'Board.java', and 'BoardController.java'. The 'Board.java' tab is currently active. Below the tabs, there is a code editor window containing the following Java code:

```
1 package com.godcoder.myhome.model;
2
3 import lombok.Data;
4
5 @Data
6 public class Board {
7     private Long id;
8     private String title;
9     private String content;
10 }
11
```

To the right of the code editor, the 'File' menu is open, displaying various options like 'New', 'Open...', 'Settings...', 'Project Structure...', etc. The 'Settings...' option is highlighted with a red box. The status bar at the bottom right shows the file name 'Board' and the line number '14048 --- [ restartedMain] o.s.s.concurrent.ThreadPool'.

```
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help myhome [Board]
New > godcoder > myhome > model > Board
Open... Open Recent Close Project
Settings... Ctrl+Alt+S Project Structure... Ctrl+Alt+Shift+S Other Settings >
Import Settings... Export Settings... Settings Repository... Export Project to Eclipse...
Save All Ctrl+S Reload All from Disk Ctrl+Alt+Y Invalidate Caches / Restart...
Export to HTML... Print... Add to Favorites >
File Encoding Remove BOM Associate with File Type...
14048 --- [ restartedMain] o.s.s.concurrent.ThreadPool
```

The screenshot shows the IntelliJ IDEA interface with the Plugins section selected in the left sidebar (highlighted with a red box). A search bar at the top contains the query "lombok". The results list includes several plugins:

- Scala
- TeamCity
- Rust
- AWS Toolkit
- Gradianto
- File Watchers

The "lombok" search result is displayed in a larger panel on the right, showing its details:

Marketplace

Plugins

Search Results (14)

Sort By: Relevance

**Lombok**

± 8.5M ⭐ 4.3 Michail Plushnikov

**Install**

Json2Pojo with Lombok

± 35.7K ⭐ 4.5 Leon

**Install**

Hrisey Plugin

± 16.7K Maciej Górski

**Install**

Jpa Support

± 62.2K ⭐ 4.8 https://github.com/carter-ya/ide...

**Install**

Mateu MDD

± 1.6K Miguel Pérez

**Install**

The "Lombok" plugin entry has an "Install" button highlighted with a red box.

The screenshot shows the IntelliJ IDEA interface with the following details:

- Project Structure:** The left sidebar displays the project structure under "myhome".
  - src/main/java:** Contains packages `com.godcoder.myhome`, `com.godcoder.myhome.controller`, `com.godcoder.myhome.model`, and `com.godcoder.myhome.repository`.
    - repository:** Contains the file `BoardRepository`, which is highlighted with a red box.
  - resources:** Contains `static`, `templates`, and `application.properties`.
- Code Editor:** The right panel shows the code for `BoardRepository.java`. The code is as follows:

```
package com.godcoder.myhome.repository;

import com.godcoder.myhome.model.Board;
import org.springframework.data.jpa.repository.JpaRepository;

public interface BoardRepository extends JpaRepository<Board, Long> {}
```

A red box highlights the entire code block.
- Toolbars and Status Bar:** Standard IntelliJ IDEA toolbars and status bar are visible at the top and bottom of the interface.

```
MyhomeApplication.java X Board.java X BoardRepository.java X BoardController.java X
```

```
10 @Controller
11 @RequestMapping("/board")
12 public class BoardController {
13
14     @Autowired
15     private BoardRepository boardRepository;
16
17     @GetMapping("/list")
18     public String list(Model model) {
19
20         return "board/list";
21     }
22 }
```

```
MyhomeApplication.java X Board.java X BoardRepository.java X BoardController.java X
```

```
3 @Controller
4 @RequestMapping("/board")
5 public class BoardController {
6
7     @Autowired
8     private BoardRepository boardRepository;
9
10    @GetMapping("/list")
11    public String list(Model model) {
12        List<Board> boards = boardRepository.findAll();
13        model.addAttribute("boards", boards);
14        return "board/list";
15    }
16 }
```



# Thymeleaf

29 October 2018: Version 3.0.4 of the **Thymeleaf + Spring Security** integration package has been published, including support for Spring Security 5 in both Spring MVC and Spring WebFlux applications. See the [announcement](#).

29 October 2018: Thymeleaf 3.0.11 has been published. See the [announcement](#).

Thymeleaf is a modern server-side Java template engine for both web and standalone environments.

Thymeleaf's main goal is to bring elegant, natural templates to your development workflow - HTML that can be

```
        class=" ${prodStat.odd}? 'odd'">
        </td>
        </td>
        <td th:text="${prod.inStock}? ${true} : ${false}">yes</td>
        <td>
            <span th:text="#{lists.size(prod.comments)}">2</span> comment/s
            <a href="@{/product/comments(prodId=${prod.id})}">
                th:href="@{/product/comments(prodId=${prod.id})}"
                th:if="#{not #lists.isEmpty(prod.comments)}">view</a>
            </td>
        </td>
    </tr>
</table>
```

: attribute:

list.html - resources - Visual Studio Code

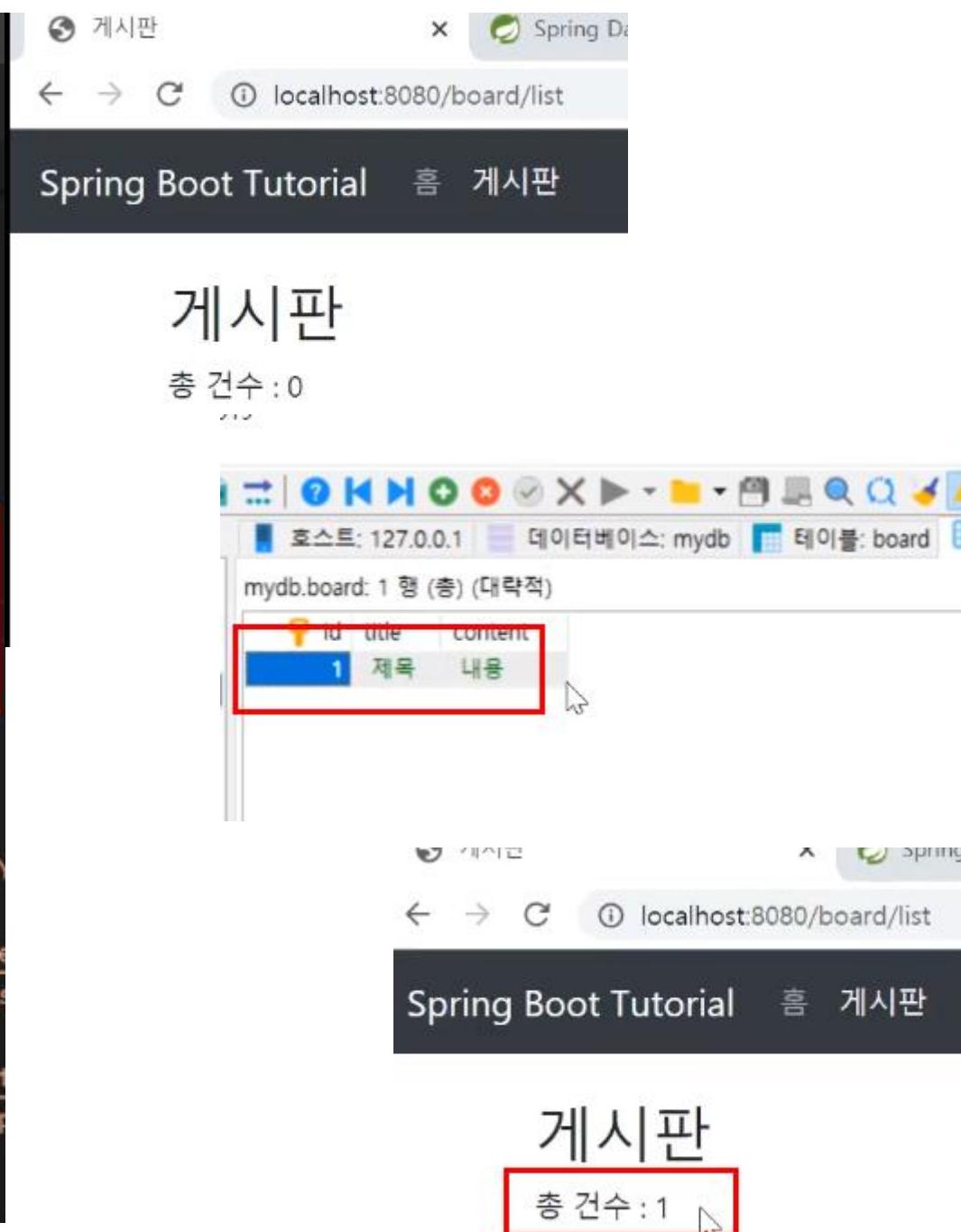
index.html list.html

templates > board > list.html > html > body > div.container > div > span

```
9         </li>
10        <li class="nav-item">
11            <a class="nav-link" href="#">게시판</a>
12        </li>
13    </ul>
14  </div> -->
15</nav>

16
17<div class="container">
18    <h2>게시판</h2>
19    <div>총 건수 : <span th:text="#{lists.size(boards)}"></span></div>
20</div>

21
22<!-- Optional JavaScript -->
23<!-- jQuery first, then Popper.js, then Bootstrap JS -->
24<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"
25       integrity="sha384-DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbV
26       crossorigin="anonymous"></script>
27<script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.js"
28       integrity="sha384-Q6E9RHvbIYZFJoft+2mJbHaEWldlvI9IOYy5n3zV9zzTtmI3Uk
29       crossorigin="anonymous"></script>
30<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/js/bootstrap.min.js"
31       integrity="sha384-OgVRvuATP1z7jHLkuOU7Xw704+h835Lr+6QL9UvYjZE3Ip
32       crossorigin="anonymous"></script>
33
34</body>
```

















게시판 | Forms · Bootstrap v4.5 | Spring | Guides | Getting Started | Handling | Tutorial: Using Thymeleaf

thymeleaf.org/doc/tutorials/3.0/usingthymeleaf.html

1 Introducing Thymeleaf

- 1.1 What is Thymeleaf?
- 1.2 What kind of templates can Thymeleaf process?
- 1.3 Dialects: The Standard Dialect

2 The Good Thymes Virtual Grocery

- 2.1 A website for a grocery
- 2.2 Creating and configuring the Template Engine

The Template Resolver

The Template Engine

3 Using Texts

- 3.1 A multi-language welcome
- Using th:text and externalizing text
- Contexts
- Executing the template engine

 Thymeleaf

# Tutorial: Using Thymeleaf

Document version: 20181029 - 29 October 2018  
Project version: 3.0.11.RELEASE  
Project web site: <https://www.thymeleaf.org>

[thymeleaf.org/doc/tutorials/3.0/usingthymeleaf.html](https://www.thymeleaf.org/doc/tutorials/3.0/usingthymeleaf.html)

## 1 Introducing Thymeleaf

- 1.1 What is Thymeleaf?
- 1.2 What kind of templates can Thymeleaf process?
- 1.3 Dialects: The Standard Dialect

## 2 The Good Thymes Virtual Grocery

- 2.1 A website for a grocery
- 2.2 Creating and configuring the Template Engine
  - The Template Resolver
  - The Template Engine

## 3 Using Texts

- 3.1 A multi-language welcome
  - Using th:text and externalizing text
  - Contexts
  - Executing the template engine

The real processing of these expressions and their conversion to the URLs that will be output is done by implementations of the `org.thymeleaf.linkbuilder.ILinkBuilder` interface that are registered into the `ITemplateEngine` object being used.

By default, a single implementation of this interface is registered of the class `org.thymeleaf.linkbuilder.StandardLinkBuilder`, which is enough for both offline (non-web) and also web scenarios based on the Servlet API. Other scenarios (like integration with non-ServletAPI web frameworks) might need specific implementations of the link builder interface.

Let's use this new syntax. Meet the `th:href` attribute:

```
<!-- Will produce 'http://localhost:8080/gtvg/order/details?orderId=3' (plus rewriting) -->
<a href="details.html" th:href="@{http://localhost:8080/gtvg/order/details(orderId=${o.id})}">view</a>

<!-- Will produce '/gtvg/order/details?orderId=3' (plus rewriting) -->
<a href="details.html" th:href="@{/order/details(orderId=${o.id})}">view</a>

<!-- Will produce '/gtvg/order/3/details' (plus rewriting) -->
<a href="details.html" th:href="@{/order/{orderId}/details(orderId=${o.id})}">view</a>
```

Some things to note here:

- `th:href` is a modifier attribute: once processed, it will compute the link URL to be used and set that value to the `href` attribute of the `<a>` tag.
- We are allowed to use expressions for URL parameters (as you can see in `orderId=${o.id}`). The required URL-parameter-encoding operations will also be automatically performed.

list.html

```
templates > board > list.html > html > body > div.container > table.table-striped > tbody > tr > td > a  
33     </ul>  
34   </div> -->  
35 </nav>  
  
37 <div class="container">  
38   <h2>게시판</h2>  
39   <div>총 건수 : <span th:text="#{lists.size(boards)}"></span></div>  
40   <table class="table table-striped">  
41     <thead>  
42       <tr>  
43         <th scope="col">번호</th>  
44         <th scope="col">제목</th>  
45         <th scope="col">작성자</th>  
46       </tr>  
47     </thead>  
48     <tbody>  
49       <tr th:each="board : ${boards}">  
50         <td th:text="${board.id}">Mark</td>  
51         <td><a th:text="${board.title}" href="@{/order/details(orderId=${o.id})}">Otto</a></td>  
52         <td>총길동</td>  
53       </tr>  
54     </tbody>  
55   <div class="text-right">  
56     <a type="button" class="btn btn-primary" th:href="@{/board/form}">쓰기</a>  
57   </div>  
58 </div>  
  
59 <!-- Optional JavaScript -->  
60 <!-- jQuery first, then Popper.js, then Bootstrap JS -->  
61 <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"
```

```
body > div.container > table.table-striped > tbody > tr > td > a  
text="#{lists.size(boards)}"></span></div>  
e-striped">  
  
번호</th>  
제목</th>  
작성자</th>
```

```
48   <tbody>  
49     <tr th:each="board : ${boards}">  
50       <td th:text="${board.id}">Mark</td>  
51       <td><a th:text="${board.title}" th:href="@{/board/form(id=${board.id})}">Otto</a></td>  
52       <td>총길동</td>  
53     </tr>  
54   </tbody>  
55   <div class="text-right">  
56     <a type="button" class="btn btn-primary" th:href="@{/board/form}">쓰기</a>  
57   </div>
```

```
BoardController.java × Board.java × BoardRepository.java ×
19     @GetMapping("/list")
20     public String list(Model model) {
21         List<Board> boards = boardRepository.findAll();
22         model.addAttribute("boards", boards);
23         return "board/list";
24     }
25
26     @GetMapping("/form")
27     public String form(Model model, @RequestParam(required = false) Long id) {
28         model.addAttribute("board", new Board());
29         return "board/form";
30     }
31
32     @PostMapping("/form")
33     public String greetingSubmit(@ModelAttribute Board board) {
```

BoardController > form()

```
BoardController.java × Board.java × BoardRepository.java ×
22         model.addAttribute("boards", boards);
23         return "board/list";
24     }
25
26     @GetMapping("/form")
27     public String form(Model model, @RequestParam(required = false) Long id) {
28         if(id == null) {
29             model.addAttribute("board", new Board());
30         } else {
31             Board board = boardRepository.findById(id).orElse(null);
32             model.addAttribute("board", board);
33         }
34         return "board/form";
35     }
```

게시판

Forms · Bootstrap v4.5 | Spring | Guides | Getting Started | Handling | Tutorial: Using Thymeleaf

localhost:8080/board/form?id=1

Spring Boot Tutorial | 게시판

# 게시판

제목

제목1

내용

내용!!!

취소

확인

list.html

form.html X

templates > board > form.html > html > body > div.container > form > input

```
13
14     <title>게시판</title>
15 </head>
16
17 <body>
18     <nav class="navbar navbar-expand-md navbar-dark bg-dark fixed-top" th:replace="fragments/common :: menu"
19     </nav>
20
21     <div class="container">
22         <h2>게시판</h2>
23         <form action="#" th:action="@{/board/form}" th:object="${board}" method="post">
24             <input type="hidden" th:field="*{id}"> [Red Box]
25             <div class="form-group">
26                 <label for="title">제목</label>
27                 <input type="text" class="form-control" id="title" th:field="*{title}">
28             </div>
29             <div class="form-group">
30                 <label for="content">내용</label>
31                 <textarea class="form-control" id="content" rows="3" th:field="*{content}"></textarea>
32             </div>
33             <div class="text-right">
34                 <a type="button" class="btn btn-primary" th:href="@{/board/list}">취소</a>
35                 <button type="submit" class="btn btn-primary">확인</button>
36             </div>
37         </form>
38     </div>
```

list.html X

form.html

templates > board > list.html > html > body > div.container > div.text-right > a.btn.btn-primary

```
37     <div class="container">
38         <h2>게시판</h2>
39         <div>총 건수 : <span th:text="#lists.size(boards)"></span></div>
40         <table class="table table-striped">
41             <thead>
42                 <tr>
43                     <th scope="col">번호</th>
44                     <th scope="col">제목</th>
45                     <th scope="col">작성자</th>
46                 </tr>
47             </thead>
48             <tbody>
49                 <tr th:each="board : ${boards}">
50                     <td th:text="${board.id}">Mark</td>
51                     <td th:text="${board.title}">Otto</td>
52                     <td>홍길동</td>
53                 </tr>
54             </tbody>
55             <div class="text-right">
56                 <a type="button" class="btn btn-primary" th:href="@{/board/form}">쓰기</a>
57             </div>
58         </div>
59     
```

BoardController.java

Board.java

BoardRepository.java

```
31         model.addAttribute("board", new Board());
32
33     }
34
35     @PostMapping("/form")
36     public String greetingSubmit(@ModelAttribute Board board) {
37         boardRepository.save(board);
38         return "redirect:/board/list";
39     }
40
41 }
42 }
```

list.html

form.html X

??

templates > board > form.html > html > body > div.container > form > div.form-group > textarea#content.form-control

```
13
14      <title>게시판</title>
15  </head>
16
17  <body>
18      <nav class="navbar navbar-expand-md navbar-dark bg-dark fixed-top" th:replace="fragments/common :: menu">
19          </nav>
20
21      <div class="container">
22          <h2>게시판</h2>
23          <form action="#" th:action="@{/board/form}" th:object="${board}" method="post">
24              <div class="form-group">
25                  <label for="title">제목</label>
26                  <input type="text" class="form-control" id="title" th:field="*{title}">
27              </div>
28              <div class="form-group">
29                  <label for="content">내용</label>
30                  <textarea class="form-control" id="content" rows="3" th:field="*{content}"></textarea>
31              </div>
32              <div class="text-right">
33                  <a type="button" class="btn btn-primary" th:href="@{/board/list}">취소</a>
34                  <button type="submit" class="btn btn-primary">확인</button>
35              </div>
36          </form>
37      </div>
38
39      <!-- Optional JavaScript -->
40      <!-- jQuery first, then Popper.js, then Bootstrap JS -->
```







Why Spring ▾Learn ▾Projects ▾TrainOverviewQuickstartGuidesBlog

# Spring makes things simple.

WHY SPRINGQUICKSTART

# Getting Started Guides

Designed to be completed in 15-30 minutes, these guides provide quick, hands-on instructions for building the "Hello World" of any development task with Spring. In most cases, the only prerequisites are a JDK and a text editor.



## [Authenticating a User with LDAP](#)

Learn how to secure an application with LDAP.



## [Accessing Vault](#)

Learn how to use Spring Vault to load secrets from HashiCorp Vault



## [Securing a Web Application](#)

Learn how to protect your web application with Spring Security.



src/main/java/com/example/securingweb/WebSecurityConfig.java ) ensures that only authenticated users can see the secret greeting:

```
package com.example.securingweb;  
  
import org.springframework.context.annotation.Bean;  
import org.springframework.context.annotation.Configuration;  
import org.springframework.security.config.annotation.web.builders.HttpSecurity;  
import org.springframework.security.config.annotation.web.configuration.EnableWebSecurity;  
import  
org.springframework.security.config.annotation.web.configuration.WebSecurityConfigurerAdapter;  
import org.springframework.security.core.userdetails.User;  
import org.springframework.security.core.userdetails.UserDetails;  
import org.springframework.security.core.userdetails.UserDetailsService;  
import org.springframework.security.provisioning.InMemoryUserDetailsManager;  
  
@Configuration  
@EnableWebSecurity  
public class WebSecurityConfig extends WebSecurityConfigurerAdapter {  
    @Override  
    protected void configure(HttpSecurity http) throws Exception {  
        http  
            .authorizeRequests()  
                .antMatchers("/", "/home").permitAll()  
                .anyRequest().authenticated()  
                .and()  
            .formLogin()  
                .loginPage("/login")  
                .permitAll()  
                .and()  
            .logout()  
                .permitAll();  
    }  
}
```

**COPY**

Project



BoardController.java

1: Project

myhome C:\Users\godcoder.SUPERGAME\Doc

- .idea
- .mvn
- src
  - main
    - java
      - com.godcoder.myhome
        - config
        - controller
        - model
        - repository
        - validator
        - MyhomeApplication
      - resources
      - test
    - target
  - .gitignore
  - mvnw
  - mvnw.cmd
  - myhome.iml
  - pom.xml

External Libraries

Scratches and Consoles

```
1 package com.godcoder.myhome.controller;
2
3 import ...
4
5 @Controller
6 @RequestMapping("/board")
7 public class BoardController {
8
9     @Autowired
10    private ...
11
12    @GetMapping("/board")
13    public ...
14
15    @GetMapping("/board/list")
16    public ...
17
18    @GetMapping("/board/write")
19    public ...
20
21    @PostMapping("/board/write")
22    public ...
23
24    @GetMapping("/board/detail")
25    public ...
26
27    @PutMapping("/board/detail")
28    public ...
29
30    @DeleteMapping("/board/detail")
31    public ...
32
33    // ...
34
35    Page<Board> boards = boardRepository.findByTitleContaining("...");
36
37    int startPage = Math.max(1, boards.getPageable().get...
38
39    int endPage = Math.min(boards.getTotalPages(), boards...
40
41    model.addAttribute("startPage", startPage);
42
43    model.addAttribute("endPage", endPage);
44
45    model.addAttribute("boards", boards);
```

New Java Class

WebSecurityConfig

Class

Interface

Enum

Annotation

JavaFXApplication

1: Project

myhome C:\Users\godcoder.SUPERGAME\Doc

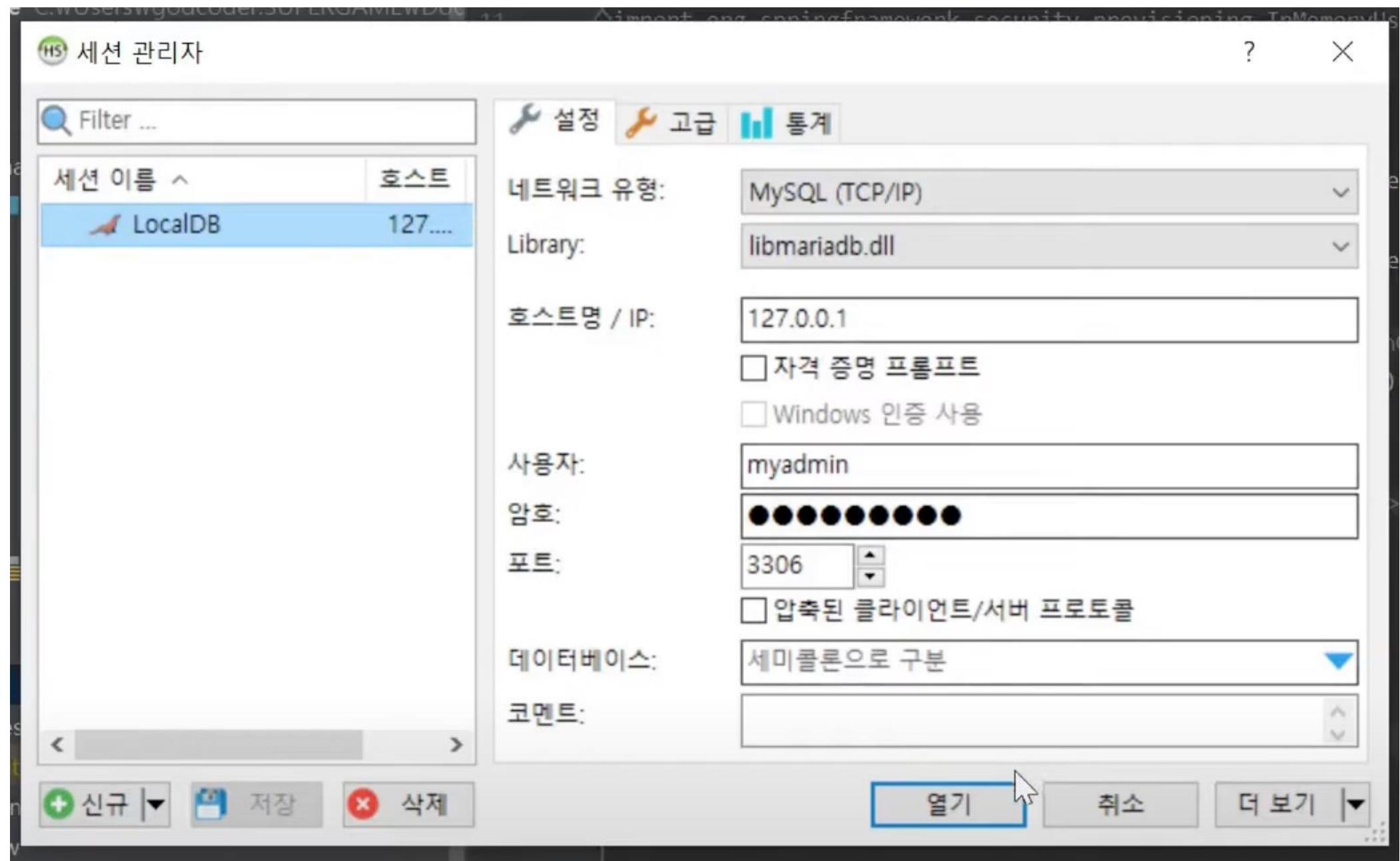
- Project
- .idea
- .mvn
- src
  - main
    - java
      - com.godcoder.myhome
        - config
          - WebSecurityConfig
  - resources
  - test
  - target
    - .gitignore
    - mvnw
    - mvnw.cmd
    - myhome.iml
    - pom.xml
- External Libraries
- Scratches and Consoles

2: Structure

3: Favorites

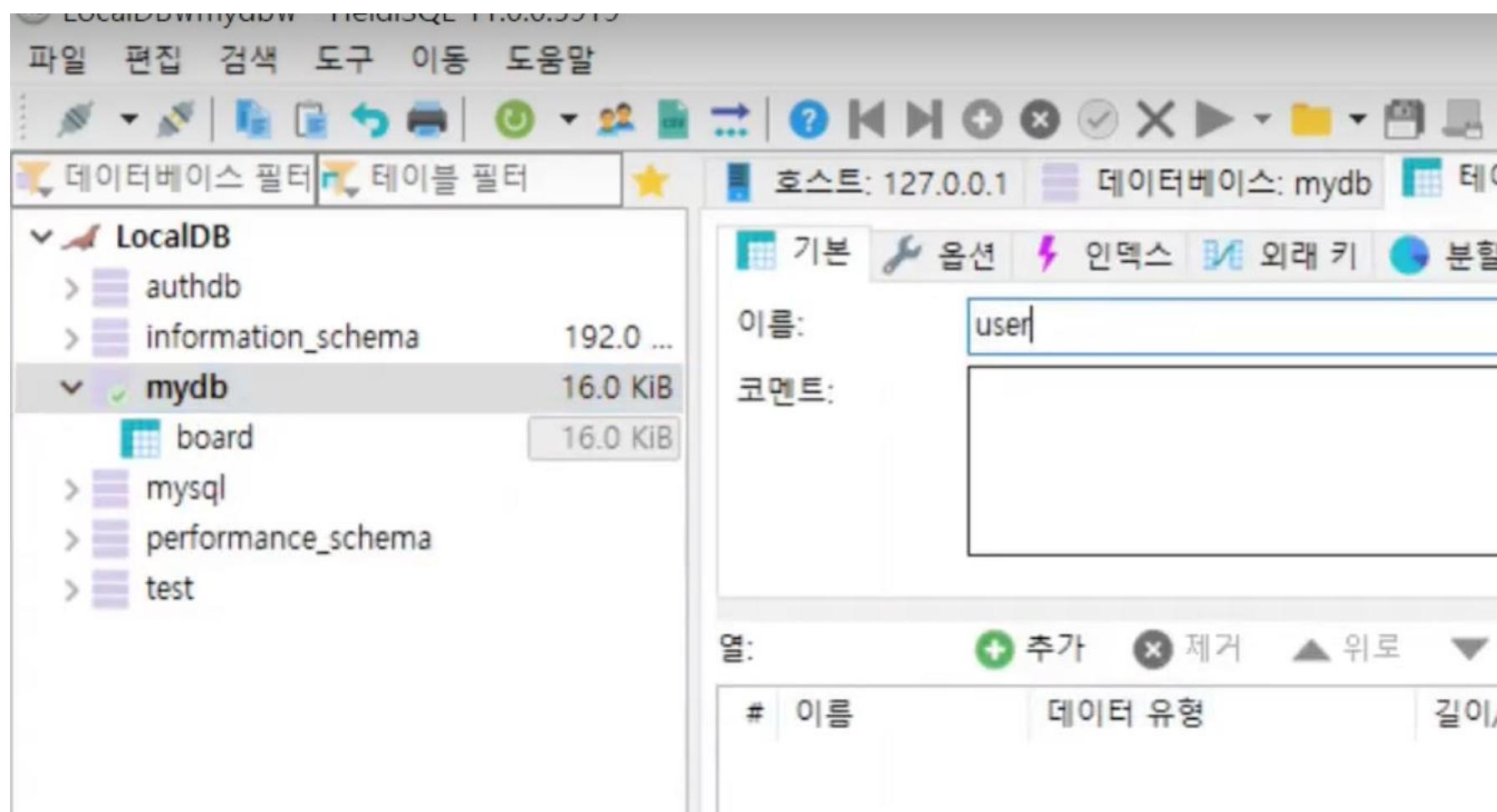
BoardController.java > WebSecurityConfig.java

```
1 package com.godcoder.myhome.config;
2
3 import org.springframework.context.annotation.Bean;
4 import org.springframework.context.annotation.Configuration;
5 import org.springframework.security.config.annotation.web.builders.HttpSecurity;
6 import org.springframework.security.config.annotation.web.configuration.EnableWebSecurity;
7 import org.springframework.security.config.annotation.web.configuration.WebSecurityConfigurerAdapter;
8 import org.springframework.security.core.userdetails.User;
9 import org.springframework.security.core.userdetails.UserDetails;
10 import org.springframework.security.core.userdetails.UserDetailsService;
11 import org.springframework.security.provisioning.InMemoryUserDetailsManager;
12
13 @Configuration
14 @EnableWebSecurity
15 public class WebSecurityConfig extends WebSecurityConfigurerAdapter {
16     @Override
17     protected void configure(HttpSecurity http) throws Exception {
18         http
19             .authorizeRequests() ExpressionUrlAuthorizationConfigurer<H>.ExpressionInterceptUrlRegistry
20             .antMatchers( ...antPatterns: "/", "/home").permitAll()
21             .anyRequest().authenticated()
22             .and() HttpSecurity
23             .formLogin() FormLoginConfigurer<HttpSecurity>
24             .loginPage("/login")
25             .permitAll()
26             .and() HttpSecurity
```



The screenshot shows the HeidiSQL interface with the following details:

- Toolbar:** Includes icons for file operations (New, Open, Save, Print, Find, Copy, Paste, Undo, Redo, etc.), search, and connection management.
- Host and Database:** Host: 127.0.0.1, Database: mydb.
- Table List:** Shows a table named "board" with 4 rows and 16.0 KiB size.
- Database Context Menu:** A context menu is open for the "mydb" database, listing options like 편집(K), 삭제...(L), 테이블 비우기...(M), 루틴 실행...(N), and several other database-related commands.
- Table Context Menu:** A context menu is open for the "board" table, listing options like 새로 생성(O), 데이터 탭 필터 초기화(P), and various table management commands.
- Submenu for Table Creation:** A submenu under "새로 생성(O)" is open, showing 테이블(U) selected, along with other options: 데이터베이스(T), 테이블 복사(V), 뷰(W), 저장 루틴(X), 트리거(Y), and 이벤트(Z).
- Bottom Bar:** Includes a filter field with "정규 표현식" (Regular Expression) and a help icon.



파일 편집 검색 도구 이동 도움말

데이터베이스 필터 테이블 필터 호스트: 127.0.0.1 데이터베이스: mydb 테이블: user 데이터 쿼리

LocalDB

- authdb
- information\_schema
- mydb
  - board 16.0 KiB
  - user 32.0 KiB
- mysql
- performance\_schema
- test

기본 옵션 인덱스 외래 키 분할 CREATE 코드 ALTER 코드

+ 추가 × 제거 ✎ 청소 ▲ 위로 ▾ 아래로

이름

- ▶ PRIMARY KEY
- ▶ username

열: + 추가 × 제거 ▲ 위로 ▾ 아래로

#	이름	데이터 유형	길이/설정	부호 없음	NULL 허용	0으...	기본값	코멘트
1	id	BIGINT	20	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	AUTO_INCREMENT	
2	username	VARCHAR	50	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	기본값 없음	
3	password	VARCHAR	100	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	기본값 없음	
4	enabled	BIT	1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	기본값 없음	



jdbcauthentication



All

Maps

News

Videos

Shopping

More

Settings

Tools

About 29,500 results (0.33 seconds)

[www.baeldung.com](http://www.baeldung.com/spring-security-jdbc-authentication) › spring-security-jdbc-authentication ▾

## [Spring Security: Exploring JDBC Authentication | Baeldung](http://www.baeldung.com/spring-security-jdbc-authentication)

Feb 22, 2020 - Explore the capabilities offered by Spring to perform **JDBC Authentication** using an existing DataSource configuration.

You've visited this page 4 times. Last visit: 7/28/20



[stackoverflow.com](https://stackoverflow.com/questions/6250373/jdbcauthentication-instead-of-inmemoryauthentication) › questions › jdbcauthentication-inst... ▾

## [jdbcAuthentication\(\) instead of inMemoryAuthentication ...](https://stackoverflow.com/questions/6250373/jdbcauthentication-instead-of-inmemoryauthentication)

Aug 4, 2010 - my guess the mistake is in your code with `inMemoryAuthentication().dataSource()`



# Spring Security: Exploring JDBC Authentication

Last modified: February 22, 2020

by Ger Roza

Persistence

Spring Security



Authentication configuration offers the possibility of **customizing the queries used to retrieve user details in the authentication process.**

## 4.3. Customizing the Search Queries [🔗](#)

Adapting the queries is quite easy. We simply have to provide our own SQL statements when configuring the *AuthenticationManagerBuilder*.

```
1 @Autowired
2 public void configureGlobal(AuthenticationManagerBuilder auth)
3     throws Exception {
4     auth.jdbcAuthentication()
5         .dataSource(dataSource)
6         .usersByUsernameQuery("select email,password(enabled "
7             + "from bael_users "
8             + "where email = ?")
9         .authoritiesByUsernameQuery("select email,authority "
10            + "from authorities "
11            + "where email = ?");
12 }
```

```
BoardController.java × WebSecurityConfig.java × application.properties ×
protected void configure(HttpSecurity http) throws Exception {
    http
        .authorizeRequests() ExpressionUrlAuthorizationConfigurer<HttpSecurity>.ExpressionInterceptUrlRegistry
            .antMatchers( ...antPatterns: "/", "/home").permitAll()
            .anyRequest().authenticated()
            .and() HttpSecurity
        .formLogin() FormLoginConfigurer<HttpSecurity>
            .loginPage("/login")
            .permitAll()
            .and() HttpSecurity
        .logout() LogoutConfigurer<HttpSecurity>
            .permitAll();
}
}

@Autowired
public void configureGlobal(AuthenticationManagerBuilder auth)
    throws Exception {
    auth.jdbcAuthentication()
        .dataSource(dataSource)
        .usersByUsernameQuery("select email,password(enabled "
            + "from bael_users "
            + "where email = ?")
        .authoritiesByUsernameQuery("select email,authority "
            + "from authorities "
            + "where email = ?");
}
```



## 2.2. Configuring JDBC Authentication

We'll use [Spring Security's \*AuthenticationManagerBuilder\* configuration helper](#) to configure JDBC Authentication:

```
1  @Autowired
2  private DataSource dataSource;
3
4  @Autowired
5  public void configureGlobal(AuthenticationManagerBuilder auth)
6      throws Exception {
7      auth.jdbcAuthentication()
8          .dataSource(dataSource)
9          .withDefaultSchema()
10         .withUser(User.withUsername("user")
11             .password(passwordEncoder().encode("pass"))
12             .roles("USER")));
13 }
14
15 @Bean
16 public PasswordEncoder passwordEncoder() {
17     return new BCryptPasswordEncoder();
18 }
```

```
BoardController.java X WebSecurityConfig.java X application.properties X
8 import org.springframework.security.config.annotation.web.configuration.EnableWebSecurity;
9 import org.springframework.security.config.annotation.web.configuration.WebSecurityConfigurerAdapter;
10 import org.springframework.security.core.userdetails.User;
11 import org.springframework.security.core.userdetails.UserDetails;
12 import org.springframework.security.core.userdetails.UserDetailsService;
13 import org.springframework.security.provisioning.InMemoryUserDetailsManager;
14
15 import javax.sql.DataSource;
16
17 @Configuration
18 @EnableWebSecurity
19 public class WebSecurityConfig extends WebSecurityConfigurerAdapter {
20
21     @Autowired
22     private DataSource dataSource;
23
24     @Override
25     protected void configure(HttpSecurity http) throws Exception {
26         http
27             .authorizeRequests() ExpressionUrlAuthorizationConfigurer<HttpSecurity>.ExpressionInterceptUrlRegistry
28                 .antMatchers( ...antPatterns: "/", "/home").permitAll()
29                 .anyRequest().authenticated()
30                 .and() HttpSecurity
31             .formLogin() FormLoginConfigurer<HttpSecurity>
32                 .loginPage("/login")
33                 .permitAll()
```

A screenshot of a code editor showing the `application.properties` file. The file contains the following configuration:

```
1 spring.datasource.url=jdbc:mariadb://localhost:3306/mydb
2 spring.datasource.username=myadmin
3 spring.datasource.password=rkskekfk1
4 spring.datasource.driver-class-name=org.mariadb.jdbc.Driver
```



## 2.2. Configuring JDBC Authentication

We'll use Spring Security's `AuthenticationManagerBuilder` configuration helper to configure JDBC Authentication:

```
1 @Autowired
2 private DataSource dataSource;
3
4 @Autowired
5 public void configureGlobal(AuthenticationManagerBuilder auth)
6     throws Exception {
7     auth.jdbcAuthentication()
8         .dataSource(dataSource)
9         .withDefaultSchema()
10        .withUser(User.withUsername("user")
11            .password(passwordEncoder().encode("pass"))
12            .roles("USER"));
13 }
14
15 @Bean
16 public PasswordEncoder passwordEncoder() {
17     return new BCryptPasswordEncoder();
18 }
```

```
BoardController.java × WebSecurityConfig.java × application.properties ×
34                                     .and() HttpSecurity
35                                     .logout() LogoutConfigurer<HttpSecurity>
36                                     .permitAll();
37
38
39 @Autowired
40     public void configureGlobal(AuthenticationManagerBuilder auth)
41             throws Exception {
42         auth.jdbcAuthentication()
43             .dataSource(dataSource)
44             .usersByUsernameQuery("select email,password(enabled "
45                         + "from bael_users "
46                         + "where email = ?")
47             .authoritiesByUsernameQuery("select email,authority "
48                         + "from authorities "
49                         + "where email = ?");
50
51
52     @Bean
53     public PasswordEncoder passwordEncoder() {
54         return new BCryptPasswordEncoder();
55     }
56
57 }
```

```
BoardController.java × WebSecurityConfig.java × application.properties ×
36             .and() HttpSecurity
37                 .logout() LogoutConfigurer<HttpSecurity>
38                     .permitAll();
39     }
40
41     @Autowired
42     @
43         public void configureGlobal(AuthenticationManagerBuilder auth)
44             throws Exception {
45         auth.jdbcAuthentication()
46             .dataSource(dataSource)
47             .passwordEncoder(passwordEncoder())
48             .usersByUsernameQuery("select email,password(enabled "
49                         + "from bael_users "
50                         + "where email = ?")
51             .authoritiesByUsernameQuery("select email,authority "
52                         + "from authorities "
53                         + "where email = ?");
54
55     @Bean
56     public PasswordEncoder passwordEncoder() {
57         return new BCryptPasswordEncoder();
58     }
59
60 }
```

```
BoardController.java × WebSecurityConfig.java × application.properties ×
32                                         .anyRequest().authenticated()
33                                         .and() HttpSecurity
34                                         .formLogin() FormLoginConfigurer<HttpSecurity>
35                                         .loginPage("/login")
36                                         .permitAll()
37                                         .and() HttpSecurity
38                                         .logout() LogoutConfigurer<HttpSecurity>
39                                         .permitAll();
40
41 @Autowired
42 public void configureGlobal(AuthenticationManagerBuilder auth)
43     throws Exception {
44     auth.jdbcAuthentication()
45         .dataSource(dataSource)
46         .passwordEncoder(passwordEncoder())
47         .usersByUsernameQuery("select email,password(enabled "
48             + "from bael_users "
49             + "where email = ?")
50         .authoritiesByUsernameQuery("select email,authority "
51             + "from authorities "
52             + "where email = ?");
53 }
54 | Authentication 로그인
55 Authroization 권한
```

```
@Autowired  
public void configureGlobal(AuthenticationManagerBuilder auth)  
    throws Exception {  
    auth.jdbcAuthentication()  
        .dataSource(dataSource)  
        .passwordEncoder(passwordEncoder())  
        .usersByUsernameQuery("select username, password, enabled "  
            + "from user "  
            + "where username = ?")
```

```
.authoritiesByUsernameQuery("select email,authority "
    + "from authorities "
    + "where email = ?");
```

The screenshot shows the MySQL Workbench interface. At the top, there's a toolbar with various icons. Below it, the connection information is displayed: 호스트: 127.0.0.1 and 데이터베이스: mydb. The main pane shows a tree view of databases under LocalDB, with 'mydb' selected. To the right, there's a table showing two tables: 'board' and 'user'. A context menu is open over the 'mydb' entry in the tree view. The menu items are:

- 편집(K) - Alt+Enter
- 삭제...(L)
- 테이블 비우기...(M) - Shift+Del
- 루틴 실행...(N)
- 새로 생성(O)**
- 데이터 탭 필터 초기화(P)
- 데이터베이스를 SQL로 내보내기(S)
- 유지보수(Q)
- 서버 내에서 텍스트 찾기(R) - Shift+Ctrl+F
- 벌크 테이블 편집기(T)
- 모두 펼치기(U)
- 모두 접기(V)
- 트리 방식 옵션(W)
- 인쇄...(X) - Ctrl+P
- 새로고침(Y) - F5
- 연결 해제(Z)

A secondary context menu is also open, listing options: 데이터베이스(T), 테이블(U) (which is highlighted in blue), 테이블 복사(V), 뷰(W), 저장 루틴(X), 트리거(Y), and 이벤트(Z).

파일 편집 검색 도구 이동 도움말

데이터베이스 필터 테이블 필터 ☆ 호스트: 127.0.0.1 데이터베이스: mydb 테이블: role 데이터 쿼리

LocalDB

- authdb
- information\_schema 192.0 KiB
- mydb 64.0 KiB
  - board 16.0 KiB
  - role 16.0 KiB
  - user 32.0 KiB
- mysql
- performance\_schema
- test

기본 옵션 인덱스 외래 키 분할 CREATE 코드 ALTER 코드

추가 이름 > PRIMARY KEY

제거 청소 위로 아래로

열: 추가 제거 위로 아래로

#	이름	데이터 유형	길이/설정	부호 없음	NULL 허용	0으...	기본값
1	id	BIGINT	20	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	AUTO_INCREMENT
2	name	VARCHAR	50	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	기본값 없음

@OneToOne

ex) user - user\_detail

@OneToMany

ex) user - board

@ManyToOne

ex) board - user

@ManyToMany

ex) user - role|

The screenshot shows the MySQL Workbench interface. At the top, there's a toolbar with various icons. Below it, the title bar displays "데이터베이스 필터" (Database Filter), "테이블 필터" (Table Filter), a yellow star icon, "호스트: 127.0.0.1", "데이터베이스: mydb", and a play button icon.

The main pane on the left shows a tree view of databases under "LocalDB". The "myd" database is selected, and its context menu is open. The menu items include:

- 편집(K) (Edit)
- 삭제...(L) (Delete)
- 테이블 비우기...(M) (Empty Table)
- 루틴 실행...(N) (Execute Routine)
- 새로 생성(O)** (Create New) - This item is highlighted with a blue selection bar.
- 데이터 탭 필터 초기화(P) (Reset Data Tab Filter)
- 데이터베이스를 SQL로 내보내기(S) (Export Database to SQL)
- 유지보수(Q) (Maintenance)
- 서버 내에서 텍스트 찾기(R) (Find Text in Server) (Shift+Ctrl+F)
- 벌크 테이블 편집기(T) (Bulk Table Editor)
- 모두 펼치기(U) (Expand All)
- 모두 접기(V) (Collapse All)
- 트리 방식 옵션(W) (Tree View Options)
- 인쇄...(X) (Print)
- 새로고침(Y) (Refresh)
- 연결 해제(Z) (Disconnect)

To the right of the menu, a table lists tables in the "mydb" database:

이름	행	크기	생성됨
board	4	16.0 KiB	2020-01-01
role	0	16.0 KiB	2020-01-01
	0	32.0 KiB	2020-01-01

A secondary context menu is also open on the right side of the screen, listing options for creating new objects:

- 데이터베이스(T) (Database)
- 테이블(U)** (Table) - This item is highlighted with a blue selection bar.
- 테이블 복사(V) (Copy Table)
- 뷰(W) (View)
- 저장 루틴(X) (Save Routine)
- 트리거(Y) (Trigger)
- 이벤트(Z) (Event)

호스트: 127.0.0.1 데이터베이스: mydb

기본 옵션 인덱스 외래 키

이름: user\_role

코멘트:

열: 추가 제거 위로 아래로

#	이름	데이터 유형
1	user_id	BIGINT
2	role_id	BIGINT

추가

제거

청소

위로

아래로

이름

PRIMARY KEY

user\_id

role\_id

열: 추가 제거 위로 아래로

#	이름	데이터 유형	길이/설정	부호 없음	NULL 허용	0으...	기본값
1	user_id	BIGINT		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	기본값 없음
2	role_id	BIGINT		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	기본값 없음

호스트: 127.0.0.1 데이터베이스: mydb 테이블: user\_role 데이터 쿼리

기본 옵션 인덱스 외래 키 분할 CREATE 코드 ALTER 코드

추가 제거 청소 위로 아래로

이름 > PRIMARY KEY

#	이름	데이터 유형	길이/설정	부호 없음	NULL 허용	0으...	기본값
1	user_id	BIGINT	20	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	기본값 없음
2	role_id	BIGINT	20	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	기본값 없음

The screenshot shows the MySQL Workbench interface with the following details:

- Toolbar:** Includes standard database management icons like back, forward, search, and refresh.
- Header:** Displays the host (127.0.0.1), database (mydb), and table (user\_role) names.
- Tool Buttons:** Basic (기본), Option (옵션), Index (인덱스), Foreign Key (외래 키), Split (분할), CREATE Code (CREATE 코드), and ALTER Code (ALTER 코드).
- Key Management:** Buttons for Adding (추가), Removing (제거), and Cleaning (청소) keys. A key named "FK1" is listed.
- Table Structure View:** Shows the columns of the user\_role table:

#	이름	데이터 유형	길이/설정	부호	없음	NULL 허용	0으...	기본값
1	user_id	BIGINT	20	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	기본값 없음
2	role_id	BIGINT	20	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	기본값 없음

호스트: 127.0.0.1 데이터베이스: mydb 테이블: user\_role 데이터 쿼리

기본 옵션 인덱스 외래 키 분할 CREATE 코드 ALTER 코드

추가 키 이름 제거 청소

키 이름: FK1

열 참조 테이블

열: 추가 제거 위로 아래로

#	이름	데이터 유형	길이/설정	부호 없음	NULL 허용	본값
1	user_id	BIGINT	20	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> 기본값 없음
2	role_id	BIGINT	20	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> 기본값 없음

열: 확인 취소

호스트: 127.0.0.1 데이터베이스: mydb 테이블: user\_role 데이터 쿼리

기본 옵션 인덱스 외래 키 분할 CREATE 코드 ALTER 코드

+ 추가 키 이름  
- 제거 FK\_user\_role\_user  
X 청소

열	참조 테이블	외래 열
user_id	user	<input checked="" type="checkbox"/> id <input type="checkbox"/> usernar <input type="checkbox"/> passwo <input type="checkbox"/> enabled

열: + 추가 - 제거 ▲ 위로 ▼ 아래로

#	이름	데이터 유형	길이/설정	부호 없음	NULL 허용	0으...	기본값
1	user_id	BIGINT	20	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	기본값 없음
2	role_id	BIGINT	20	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	기본값 없음

확인 취소

Donate

호스트: 127.0.0.1 | 데이터베이스: mydb | 테이블: user\_role | 데이터 | 쿼리 |

기본 옵션 인덱스 외래 키 분할 CREATE 코드 ALTER 코드

작업	키 이름	열	참조 테이블	외래 열	UPDATE ...	DELETE 될 ...
+ 추가		user_id	user	id	RESTRICT	RESTRICT
- 제거	FK_user_role_user	role_id	role	id	RESTRICT	RESTRICT
- 청소	FK_user_role_role					

열: + 추가 - 제거 ▲ 위로 ▾ 아래로

#	이름	데이터 유형	길이/설정	부호	없음	NULL 허용	0으...	기본값	코멘트	조합
1	user_id	BIGINT	20	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	기본값 없음		
2	role_id	BIGINT	20	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	기본값 없음		

```
+ " where username = ? )  
authoritiesByUsernameQuery("select username, name "  
+ "from user_role ur inner join user u on ur.user_id = u.id "  
+ "inner join role r on ur.role_id = r.id "  
+ "where email = ?");  
}
```

BoardController.java

WebSecurityConfig.java

Board.java

application.properties

```
17 import javax.sql.DataSource;
18
19 @Configuration
20 @EnableWebSecurity
21 public class WebSecurityConfig extends WebSecurityConfigurerAdapter {
22
23     @Autowired
24     private DataSource dataSource;
25
26     @Override
27     protected void configure(HttpSecurity http) throws Exception {
28         http
29             .authorizeRequests() ExpressionUrlAuthorizationConfigurer<HttpSecurity>.ExpressionInterceptUrlRegistry
30                 .antMatchers( ...antPatterns: "/", "/home").permitAll()
31                     .anyRequest().authenticated()
32                     .and() HttpSecurity
33             .formLogin() FormLoginConfigurer<HttpSecurity>
34                 .loginPage("/login")
35                 .permitAll()
36                 .and() HttpSecurity
37             .logout() LogoutConfigurer<HttpSecurity>
38                 .permitAll();
39     }
}
```

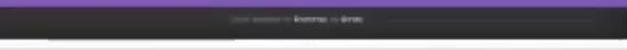
```
BoardController.java ✘ WebSecurityConfig.java ✘ Board.java ✘ application.properties ✘
16
17     import javax.sql.DataSource;
18
19     @Configuration
20     @EnableWebSecurity
21     public class WebSecurityConfig extends WebSecurityConfigurerAdapter {
22
23         @Autowired
24         private DataSource dataSource;
25
26         @Override
27         protected void configure(HttpSecurity http) throws Exception {
28             http
29                 .authorizeRequests() ExpressionUrlAuthorizationConfigurer<HttpSecurity>.ExpressionInterceptUrlRegistry
30                     .antMatchers("/*").permitAll()
31                     .anyRequest().authenticated()
32                     .and() HttpSecurity
33                     .formLogin() FormLoginConfigurer<HttpSecurity>
34                         .loginPage("/login")
35                         .permitAll()
36                         .and() HttpSecurity
37                     .logout() LogoutConfigurer<HttpSecurity>
38                         .permitAll();
39     }
```

```
BoardController.java × WebSecurityConfig.java × Board.java × application.properties ×
22
23     @Autowired
24     private DataSource dataSource;
25
26     @Override
27     protected void configure(HttpSecurity http) throws Exception {
28         http
29             .authorizeRequests() ExpressionUrlAuthorizationConfigurer<HttpSecurity>.ExpressionInterceptUrlRegistry
30             .antMatchers( ...antPatterns: "/", "/css/**").permitAll()
31             .anyRequest().authenticated()
32             .and() HttpSecurity
33             .formLogin() FormLoginConfigurer<HttpSecurity>
34                 .loginPage("/login")
35                 .permitAll()
36                 .and() HttpSecurity
WebSecurityConfig > configure()
```

The screenshot shows the Visual Studio Code interface with the following details:

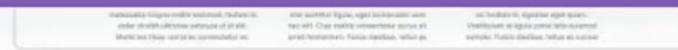
- EXPLORER** pane on the left:
  - OPEN EDITORS**: index.html (templates), # starter-template.css (static\css)
  - RESOURCES**: static\css (selected), templates, board, fragments
  - File counts: 5 files in static\css, 3 files in templates, 2 files in board, and 2 files in fragments.
- EDITOR** pane on the right:
  - File: index.html
  - Editor tab: # starter-template.css
  - Content:

```
static > css > # starter-template.css > body
body {
    padding-top: 5rem;
}
.starter-template {
    padding: 3rem 1.5rem;
    text-align: center;
}
```



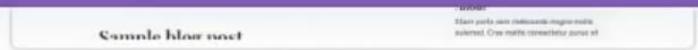
## Cover

A one-page template for building simple and beautiful home pages.



## Carousel

Customize the navbar and carousel, then add some new components.



## Blog

Magazine like blog template with header, navigation, featured content.



## Sign-in

Custom form layout and design for a simple sign in form.



## Sticky footer

Attach a footer to the bottom of the viewport when page content is short.



## Sticky footer navbar

Attach a footer to the bottom of the viewport with a fixed top navbar.

getbootstrap.com/docs/4.5/examples/sign-in/



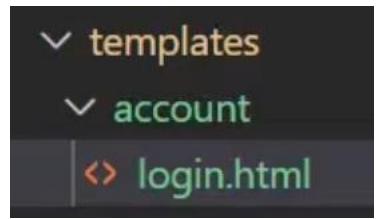
## Please sign in

Email address

Password

Remember me

Sign in



```
25
26
27 <style>
28   .bd-placeholder-img {
29     font-size: 1.125rem;
30     text-anchor: middle;
31     -webkit-user-select: none;
32     -moz-user-select: none;
33     -ms-user-select: none;
34     user-select: none;
35   }
36
37   @media (min-width: 768px) {
38     .bd-placeholder-img-lg {
39       font-size: 3.5rem;
40     }
41   }
42 </style>
43 <!-- Custom styles for this template -->
44 <link href="signin.css" rel="stylesheet">
45 </head>
46 <body class="text-center">
47   <form class="form-signin">
48     
49     <h1 class="h3 mb-3 font-weight-normal">Please sign in</h1>
50     <label for="inputEmail" class="sr-only">Email address</label>
51     <input type="email" id="inputEmail" class="form-control" placeholder="Email address" required autofocus>
52     <label for="inputPassword" class="sr-only">Password</label>
53     <input type="password" id="inputPassword" class="form-control" placeholder="Password" required>
54     <div class="checkbox mb-3">
55       <label>
56         <input type="checkbox" value="remember-me" checked> Remember me
57       </label>
58     </div>
59     <button class="btn btn-lg btn-primary btn-block" type="submit">Sign in</button>
60     <p class="mt-5 mb-3 text-muted">&copy; 2017–2020</p>
61   </form>
62 </body>
63 </html>
```

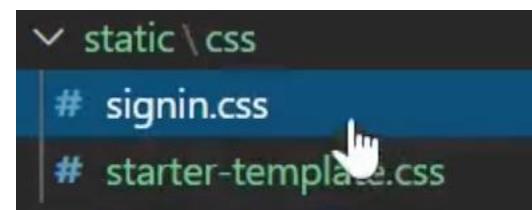
```
<body>
  <form class="form-signin">
    
    <h1 class="h3 mb-3 font-weight-normal">Please sign in</h1>
    <label for="inputEmail" class="sr-only">Email address</label>
    <input type="email" id="inputEmail" class="form-control" placeholder="Email address" required auto>
    <label for="inputPassword" class="sr-only">Password</label>
    <input type="password" id="inputPassword" class="form-control" placeholder="Password" required>
    <div class="checkbox mb-3">
      <label>
        <input type="checkbox" value="remember-me"> Remember me
      </label>
    </div>
    <button class="btn btn-lg btn-primary btn-block" type="submit">Sign in</button>
    <p class="mt-5 mb-3 text-muted">&copy; 2017-2020</p>
  </form>

  <!-- Optional JavaScript -->
  <!-- jQuery first, then Popper.js, then Bootstrap JS -->
```

```
<body class="text-center">
  <form class="form-signin">
    
      <label for="inputEmail" class="sr-only'>
```

```
.font-size: 3.5rem;
}

}
</style>
<!-- Custom styles for this template --&gt;
&lt;link href="signin.css" rel="stylesheet"&gt;
&lt;/head&gt;
&lt;body class="text-center"&gt;
  &lt;form class="form-signin"&gt;
    &lt;img class="mb-4" src="/docs/4.5/assets/brand/bootstrap-solid.svg" alt=""
    &lt;h1 class="h3 mb-3 font-weight-normal"&gt;Please sign in&lt;/h1&gt;
    &lt;label for="inputEmail" class="sr-only"&gt;Email address&lt;/label&gt;
    &lt;input type="email" id="inputEmail" class="form-control" placeholder="Email address" required="" aria-describedby="inputGroupPrepend"&gt;
    &lt;div class="checkbox"&gt;
      &lt;label for="checkbox"&gt;Remember me&lt;/label&gt;
    &lt;/div&gt;
    &lt;button class="btn btn-lg btn-primary" type="submit"&gt;Sign in&lt;/button&gt;
  &lt;/form&gt;
&lt;/body&gt;
&lt;/html&gt;</pre>
```



```
<!-- Bootstrap CSS -->
<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css" integrity="sha384-JcKs1u2htHzjyPrXlk/3GqZnLpnGJUlxJWnVZlFJZsIuVHJF4rEw46ZBQF9qYH" crossorigin="anonymous">
<link href="@{[/css/signin.css]}" rel="stylesheet">

<title>Login</title>
</head>
```

```
index.html common.html login.html # signin.css X
static > css > # signin.css > .form-signin input[type="password"]

16 .form-signin {
17   width: 100%;
18   max-width: 330px;
19   padding: 15px;
20   margin: auto;
21 }
22 .form-signin .checkbox {
23   font-weight: 400;
24 }
25 .form-signin .form-control {
26   position: relative;
27   box-sizing: border-box;
28   height: auto;
29   padding: 10px;
30   font-size: 16px;
31 }
32 .form-signin .form-control:focus {
33   z-index: 2;
34 }
35 .form-signin input[type="email"] {
36   margin-bottom: -1px;
37   border-bottom-right-radius: 0;
38   border-bottom-left-radius: 0;
39 }
40 .form-signin input[type="password"] {
41   margin-bottom: 10px;
42   border-top-left-radius: 0;
43   border-top-right-radius: 0;
44 }
```

```
BoardController.java × WebSecurityConfig.java × Board.java × application.properties ×
25
26     @Override
27     protected void configure(HttpSecurity http) throws Exception {
28         http
29             .authorizeRequests() ExpressionUrlAuthorizationConfigurer<HttpSecurity>.ExpressionInterceptUrlRegistry
30                 .antMatchers( ...antPatterns: "/", "/css/**").permitAll()
31                 .anyRequest().authenticated()
32                 .and() HttpSecurity
33             .formLogin() FormLoginConfigurer<HttpSecurity>
34                 .loginPage("/account/login")
35                 .permitAll()
36                 .and() HttpSecurity
37             .logout() LogoutConfigurer<HttpSecurity>
38                 .permitAll();
39
WebSecurityConfig > configure()
```

myhome > src > main > java > com > godcoder > myhome > controller > AccountController

Project

1: Project

.idea  
.mvn  
src  
main  
java  
com.godcoder.myhome  
config  
controller  
AccountController  
BoardApiController  
BoardController  
HomeController  
model  
Board  
repository  
validator

BoardController.java WebSecurityConfig.java AccountController.java

```
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RequestMapping;

@Controller
@RequestMapping("/account")
public class AccountController {

    @GetMapping("/login")
    public String login() {
        return "account/login";
    }
}
```

AccountController

```
y class="text-center">
<form class="form-signin">
  
  <h1 class="h3 mb-3 font-weight-normal">Please sign in</h1>
  <label for="username" class="sr-only">Username</label>
  <input type="text" id="username" class="form-control" placeholder="Username" required autofocus>
  <label for="inputPassword" class="sr-only">Password</label>
  <input type="password" id="inputPassword" class="form-control" placeholder="Password" required>
  <div class="checkbox mb-3">
    <label>
```

```
<label for="username" class="sr-only">Username</label>
<input type="text" id="username" class="form-control" placeholder="Username"
<label for="inputPassword" class="sr-only">Password</label>
<input type="password" id="inputPassword" class="form-control" placeholder="Password"
<!-- <div class="checkbox mb-3">
    <label>
        <input type="checkbox" value="remember-me"> Remember me
    </label>
</div> -->
<button class="btn btn-lg btn-primary btn-block" type="submit">Sign in</button>
<p class="mt-5 mb-3 text-muted">&copy; 2017-2020</p>
</form>
```



Please sign in

Username

Password

Sign in

© 2017-2020

given a user name of `user`, a password of `password`, and a role of `USER`.

Now you need to create the login page. There is already a view controller for the `login` view, so you need only to create the login view itself, as the following listing (from `src/main/resources/templates/login.html`) shows:

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml" xmlns:th="https://www.thymeleaf.org"
      xmlns:sec="https://www.thymeleaf.org/thymeleaf-extras-springsecurity3">
    <head>
        <title>Spring Security Example </title>
    </head>
    <body>
        <div th:if="${param.error}">
            Invalid username and password.
        </div>
        <div th:if="${param.logout}">
            You have been logged out.
        </div>
        <form th:action="@{/login}" method="post">
            <div><label> User Name : <input type="text" name="username"/> </label></div>
            <div><label> Password: <input type="password" name="password"/> </label></div>
            <div><input type="submit" value="Sign In"/></div>
        </form>
    </body>
</html>
```

The screenshot shows the `login.html` file in a code editor. A blue rectangular highlight covers the entire content of the file. Inside this highlight, several specific lines of code are highlighted with blue boxes:

- `<div th:if="${param.error}">` has a blue box around it.
- `Invalid username and password.` is followed by a blue box.
- `<div th:if="${param.logout}">` has a blue box around it.
- `You have been logged out.` is followed by a blue box.
- `<form th:action="@{/login}" method="post">` has a blue box around it.
- `<div><label> User Name : <input type="text" name="username"/> </label></div>` has a blue box around it.
- `<div><label> Password: <input type="password" name="password"/> </label></div>` has a blue box around it.
- `<div><input type="submit" value="Sign In"/></div>` has a blue box around it.

A small gray arrow points from the bottom right towards the highlighted area. The status bar at the bottom of the editor shows the path: `File / login.html`.

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml" xmlns:th="https://www.thymeleaf.org"
      xmlns:sec="https://www.thymeleaf.org/thymeleaf-extras-springsecurity3">
    <head>
        <title>Spring Security Example </title>
    </head>
    <body>
        <div th:if="${param.error}">
            Invalid username and password.
        </div>
        <div th:if="${param.logout}">
            You have been logged out.
        </div>
        <form th:action="@{/login}" method="post">
            <div><label> User Name : <input type="text" name="username"/> </label></div>
            <div><label> Password: <input type="password" name="password"/> </label></div>
            <div><input type="submit" value="Sign In"/></div>
        </form>
    </body>
</html>
```

This Thymeleaf template presents a form that captures a username and password and posts them to `/login`. As configured, Spring Security provides a filter that intercepts that request and authenticates the user. If the user fails to authenticate, the page is redirected to `/login?error` and

```
index.html common.html login.html X
templates > account > login.html > html > body.text-center > form.form-signin
13
14     <title>Login</title>
15 </head>
16
17 <body class="text-center">
18     <div th:if="${param.error}">
19         Invalid username and password.
20     </div>
21     <div th:if="${param.logout}">
22         You have been logged out.
23     </div>
24     <form class="form-signin">
25         Please sign in</h1>
27         <label for="username" class="sr-only">Username</label>
28         <input type="text" id="username" class="form-control" place
```

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml" xmlns:th="https://www.thymeleaf.org"
      xmlns:sec="https://www.thymeleaf.org/thymeleaf-extras-springsecurity3">
    <head>
        <title>Spring Security Example </title>
    </head>
    <body>
        <div th:if="${param.error}">
            Invalid username and password.
        </div>
        <div th:if="${param.logout}">
            You have been logged out.
        </div>
        <form th:action="@{/login}" method="post">
            <div><label> User Name : <input type="text" name="username"/> </label></div>
            <div><label> Password: <input type="password" name="password"/> </label></div>
            <div><input type="submit" value="Sign In"/></div>
        </form>
    </body>
</html>
```

COPY

index.html

common.html

login.html ●

templates > account > login.html > html > body.text-center > form.form-signin

```
13
14      <title>Login</title>
15  </head>
16
17  <body class="text-center">
18      <div th:if="${param.error}">
19          Invalid username and password.
20      </div>
21      <div th:if="${param.logout}">
22          You have been logged out.
23      </div>
24      <form class="form-signin" th:action="@{/login}" method="post">
25          Please sign in</h1>
```

```
th:action="@{/account/login}" method="post">
- "https://getbootstrap.com/docs/4.5/assets/bnr"
```

```
</div>
<form class="form-signin" th:action="@{/account/login}" method="post">
    Please sign in</h1>
    <label for="username" class="sr-only">Username</label>
    <input type="text" id="username" name="username" class="form-control" placeholder="Username" required>
    <label for="inputPassword" class="sr-only">Password</label>
    <input type="password" id="inputPassword" class="form-control" placeholder="Password" required>
    <!-- <div class="checkbox mb-3">
        <form class="form-signin" th:action="@{/account/login}" method="post">
            Please sign in</h1>
            <label for="username" class="sr-only">Username</label>
            <input type="text" id="username" name="username" class="form-control" placeholder="Username" required>
            <label for="inputPassword" class="sr-only">Password</label>
            <input type="password" id="inputPassword" name="password" class="form-control" placeholder="Password" required>
        <!-- <div class="checkbox mb-3">
```

[←](#) [→](#) [⟳](#)

localhost:8080/account/login



Please sign in

A text input field containing the text 'sdf'.  
A password input field containing three dots (...).[Sign in](#)



## Please sign in

Username

Password

Sign in

Invalid username and password.

File Edit Selection View Go Run Terminal Help login.html - resources - visual studio

EXPLORER ... index.html common.html login.html X

OPEN EDITORS index.html templates common.html templates\fra... M  
X login.html templates\account U

RESOURCES 10 static\css signin.css starter-template.css

templates account login.html board

```
13
14     <title>Login</title>
15     </head>
16
17     <body class="text-center">
18         <div th:if="${param.error}">
19             Invalid username and password.
20         </div>
21         <div th:if="${param.logout}">
22             You have been logged out.
23         </div>
24         <form class="form-signin" th:action="@{/login}">
25             
```

alert



Getting started

Introduction

Download

Contents

Browsers & devices

JavaScript

Theming

Build tools

Webpack

Accessibility

Layout

Content

# Introduction

Get started with Bootstrap, the world's most popular framework for building responsive, mobile-first sites, with BootstrapCDN and a template starter page.



Limited time offer: Get 10  
free Adobe Stock images.

ads via Carbon

## Quick start

A simple secondary alert—check it out!

A simple success alert—check it out!

A simple danger alert—check it out!

A simple warning alert—check it out!

A simple info alert—check it out!

A simple light alert—check it out!

A simple dark alert—check it out!

```
<div class="alert alert-primary" role="alert">  
  A simple primary alert—check it out!  
</div>  
  
<div class="alert alert-secondary" role="alert">  
  A simple secondary alert—check it out!  
</div>  
  
<div class="alert alert-success" role="alert">  
  A simple success alert—check it out!  
</div>  
  
<div class="alert alert-danger" role="alert">  
  A simple danger alert—check it out!   
</div>  
  
<div class="alert alert-warning" role="alert">  
  A simple warning alert—check it out!  
</div>  
  
<div class="alert alert-info" role="alert">  
  A simple info alert—check it out!  
</div>  
  
<div class="alert alert-light" role="alert">  
  A simple light alert—check it out!  
</div>  
  
<div class="alert alert-dark" role="alert">  
  A simple dark alert—check it out!  
</div>
```

ates > account > login.html > html > body.text-center > div.alert.alert-danger

```
<title>Login</title>
</head>

<body class="text-center">
    <div th:if="${param.error}" class="alert alert-danger" role="alert">
        Invalid username and password.
    </div>
    <div th:if="${param.logout}">
        You have been logged out.
    </div>
```

```
<title>Login</title>
</head>

<body class="text-center">
  <form class="form-signin" th:action="@{/account/login}" method="post">
    Please sign in</h1>
    <div th:if="${param.error}" class="alert alert-danger" role="alert">
      .... Invalid username and password.
    </div>
```

```
✓ account
```

```
↳ login.html
```

```
↳ login copy.html
```

```
✓ account
```

```
↳ login.html
```

```
↳ register.html
```

```
<body class="text-center">
```

```
    <form class="form-signin" th:action="@{/account/register}"
```

```
        
```

```
<body class="text-center">
    <form class="form-signin" th:action="@{/account/login}" method="post">
        
        <h1 class="h3 mb-3 font-weight-normal">회원가입</h1>
        <div th:if="${param.error}" class="alert alert-danger" role="alert">
            Invalid username and password.
        </div>
        <h1 class="h3 mb-3 font-weight-normal">회원가입</h1>
        <div th:if="${param.error}" class="alert alert-danger" role="alert">
            Invalid username and password.
        </div>
        <div th:if="${param.logout}" class="alert alert-primary" role="alert">
            You have been logged out.
        </div>
        <label for="username" class="sr-only">Username</label>
```

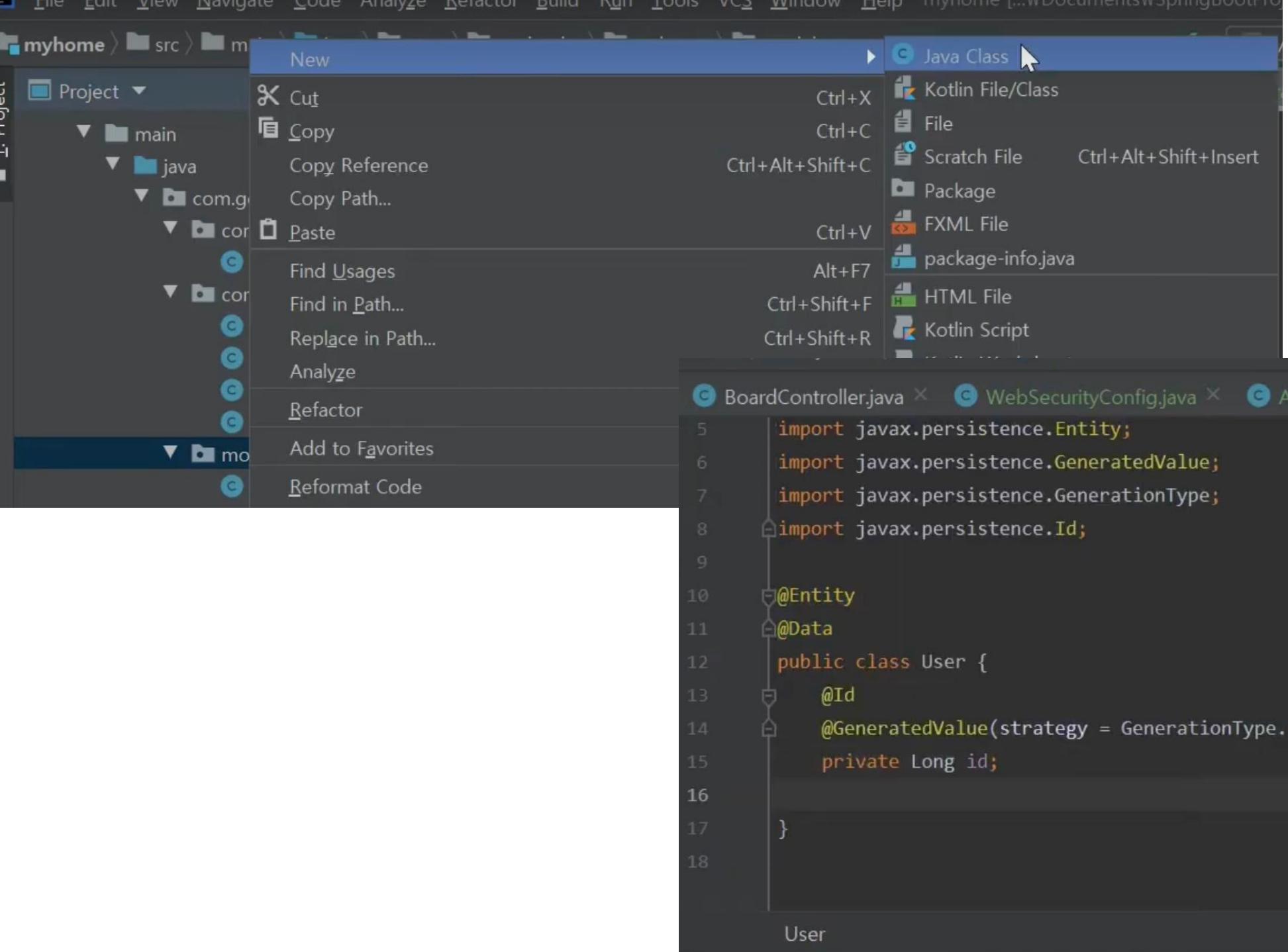
```
    <label>
```

```
        <input type="checkbox" value="remember-me"> Remember me
```

```
    </label>
```

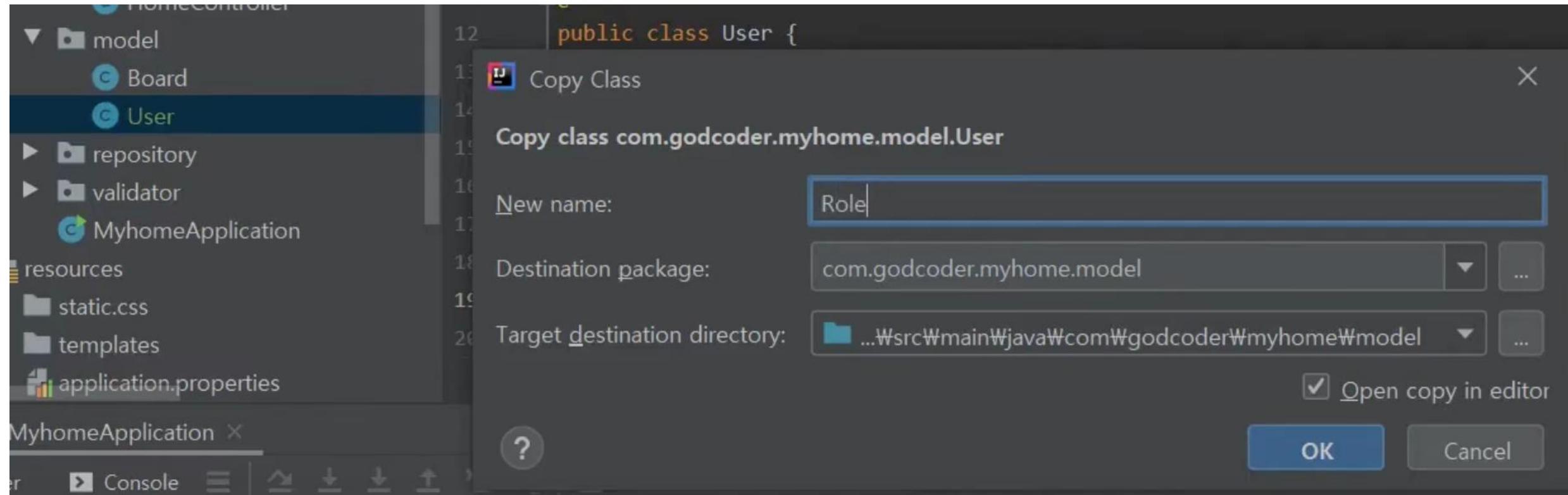
```
</div> -->
```

```
    <button class="btn btn-lg btn-primary btn-block" type="submit">회원가입</button>
```



```
BoardController.java × WebSecurityConfig.java × AccountController.java × User.java ×
6 import javax.persistence.GeneratedValue;
7 import javax.persistence.GenerationType;
8 import javax.persistence.Id;
9
10 @Entity
11 @Data
12 public class User {
13     @Id
14     @GeneratedValue(strategy = GenerationType.IDENTITY)
15     private Long id;
16
17     private String username;
18     private String password;
19     private Boolean enabled;
20 }
```

User > enabled



The image shows a Java IDE interface with two windows displaying the same code. The top window has tabs for several files: CardController.java, WebSecurityConfig.java, AccountController.java, User.java, and Role.java. The bottom window also has tabs for these files, with Role.java currently selected. Both windows show the following Java code:

```
package com.godcoder.myhome.model;

import ...

@Entity
@Data
public class Role {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

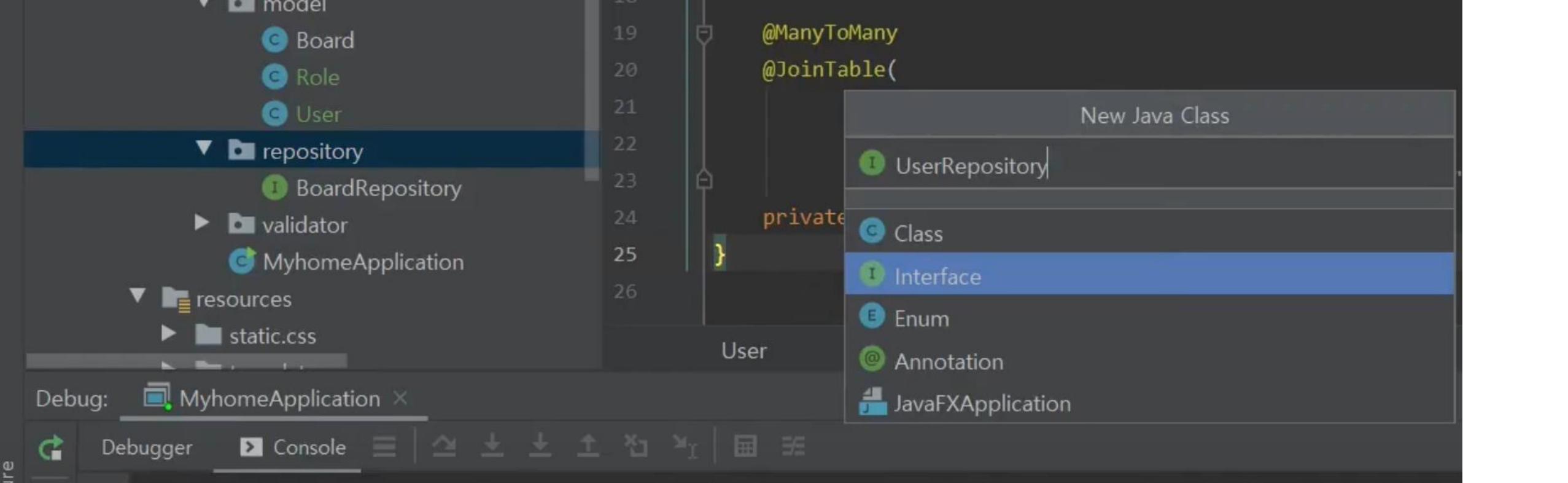
    private String name;
}
```

The code defines a class named `Role` with an `@Entity` annotation. It includes `@Data` for generating boilerplate code. The class has a primary key `id` annotated with `@Id` and `@GeneratedValue(strategy = GenerationType.IDENTITY)`. It also has a string field `name`.

```
CardController.java ×  C WebSecurityConfig.java ×  C AccountController.java ×  C User.java ×  
private Long id;  
  
private String username;  
private String password;  
private Boolean enabled;  
  
@ManyToMany  
@JoinTable(  
    name = "user_role",  
    joinColumns = @JoinColumn(name = "user_id"),  
    inverseJoinColumns = @JoinColumn(name = "role_id"))  
private List<Role> roles;  
}
```

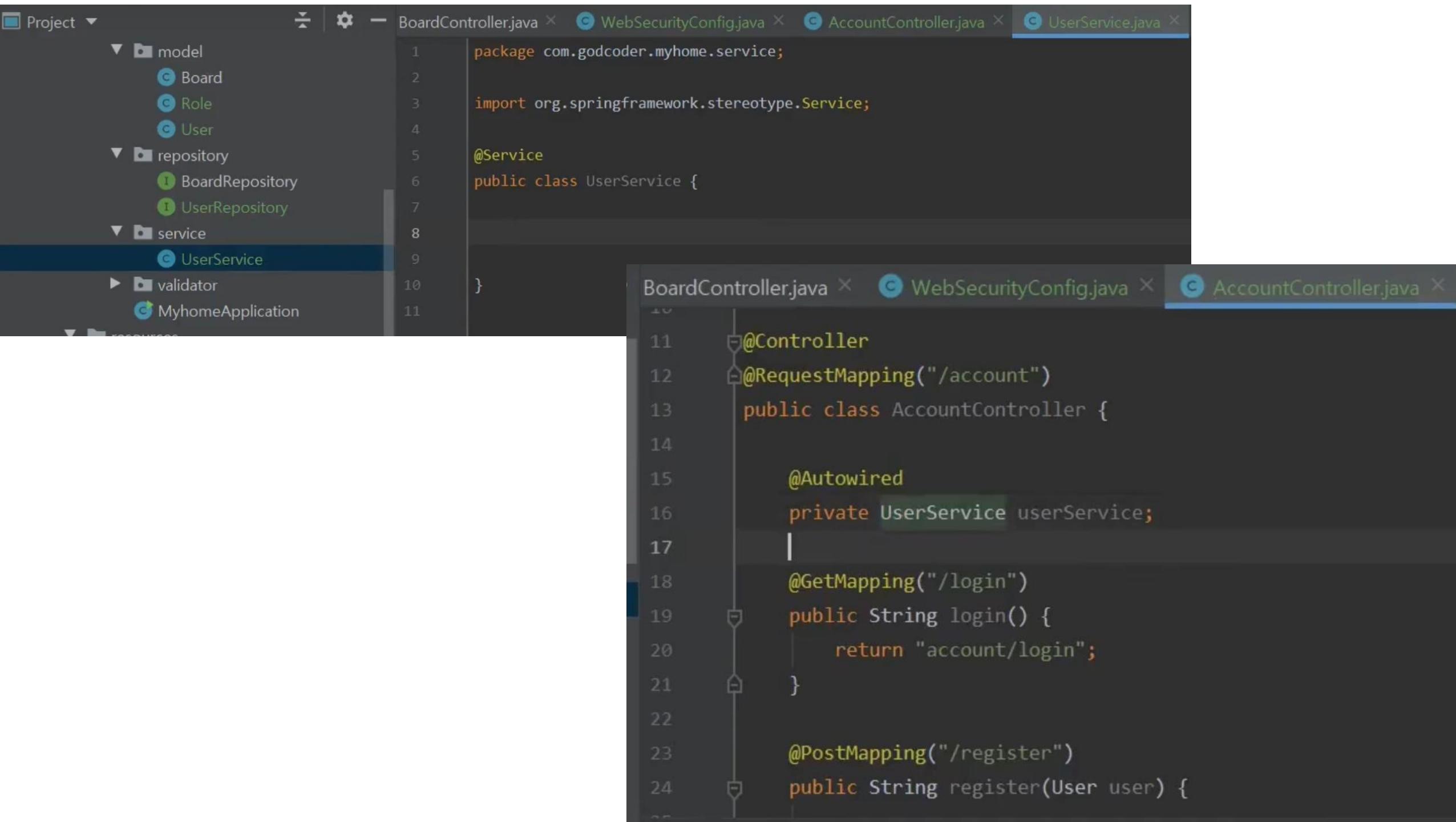
The screenshot shows the MySQL Workbench interface. On the left, the LocalDB tree displays various databases and their sizes: authdb (192.0 KiB), information\_schema (192.0 KiB), mydb (96.0 KiB), mysql (16.0 KiB), performance\_schema (32.0 KiB), and test (32.0 KiB). The mydb database is expanded, showing its tables: board, role, user, and user\_role. The role table is currently selected. On the right, the details for the mydb.role table are shown. The table has one row with id 1 and name ROLE\_USER.

	id	name
	1	ROLE_USER



```
controller.java ×  WebSecurityConfig.java ×  AccountController.java ×  User.java ×  UserRepository.java ×
package com.godcoder.myhome.repository;

import com.godcoder.myhome.model.Board;
import org.springframework.data.jpa.repository.JpaRepository;
public interface UserRepository extends JpaRepository<Board, Long> {
}
```



BoardController.java × WebSecurityConfig.java × AccountController.java ×

```
17  
18     @GetMapping("/login")  
19     public String login() {  
20         return "account/login";  
21     }  
22  
23     @PostMapping("/register")  
24     public String register(User user) {  
25         userService.save(user);  
26         return "account/login";  
27     }  
28 }  
29
```

BoardController.java × WebSecurityConfig.java × AccountController.java × UserService.java × User.java > UserRepository.java >

```
1 package com.godcoder.myhome.repository;  
2  
3 import com.godcoder.myhome.model.User; // Red box highlights this line  
4 import org.springframework.data.jpa.repository.JpaRepository;  
5  
6 public interface UserRepository extends JpaRepository<User, Long> {  
7 }
```

```
ntroller.java ×  ⚒ WebSecurityConfig.java ×  ⚒ AccountController.java ×  ⚒ UserService.java ×
import org.springframework.stereotype.Service;

@Service
public class UserService {

    @Autowired
    private UserRepository userRepository;

    @Autowired
    private PasswordEncoder passwordEncoder;
```

```
roller.java ✘  C WebSecurityConfig.java ✘  C AccountController.java ✘  C UserService.java ✘
```

```
@Autowired  
private UserRepository userRepository;  
  
@Autowired  
private PasswordEncoder passwordEncoder;  
  
public User save(User user) {  
    String encodedPassword = passwordEncoder.encode(user.getPassword());  
    user.setPassword(encodedPassword);  
  
    return userRepository.save(user);  
}  
}
```

```
itroller.java ✘  C WebSecurityConfig.java ✘  C AccountController.java ✘  C UserService.java ✘
```

```
@Autowired  
private UserRepository userRepository;  
  
@Autowired  
private PasswordEncoder passwordEncoder;  
  
public User save(User user) {  
    String encodedPassword = passwordEncoder.encode(user.getPassword());  
    user.setPassword(encodedPassword);  
    user.setEnabled(true);  
  
    return userRepository.save(user);  
}
```

× C AccountController.java × C UserService.java × C User.java × I UserRepository

```
@Autowired  
private PasswordEncoder passwordEncoder;  
  
public User save(User user) {  
    String encodedPassword = passwordEncoder.encode(user.getPassword());  
    user.setPassword(encodedPassword);  
    user.setEnabled(true);  
    Role role = new Role();  
    role.setId(11);  
    user.getRoles().add(role);  
    return userRepository.save(user);  
}  
}  
UserService > save()
```

myhome > src > main > java > com > godcoder > myhome > controller > AccountController

The screenshot shows a Java project structure on the left and a code editor on the right. The project structure includes .mvn, src, main, java, com.godcoder.myhome, config, WebSecurityConfig, controller (containing AccountController, BoardApiController, BoardController, HomeController), and model (containing Board, Role). The code editor displays AccountController.java with the following content:

```
21     return "account/login";
22 }
23
24 @GetMapping("/register")
25 public String register() {
26
27     return "account/register";
28 }
29
30 @PostMapping("/register")
31 public String register(User user) {
32
33     userService.save(user);
34
35 }
```

```
WebSecurityConfig.java × AccountController.java × HomeController.java × UserService.java × User.java × ⓘ
```

```
25
26     @Override
27     protected void configure(HttpSecurity http) throws Exception {
28         http
29             .authorizeRequests()
30                 .antMatchers("/*", "/account/register", "/css/**").permitAll()
31                 .anyRequest().authenticated()
32             .and()
33         .formLogin() FormLoginConfigurer<HttpSecurity>
34             .loginPage("/account/login")
35             .permitAll()
36             .and()
37         .logout() LogoutConfigurer<HttpSecurity>
38             .permitAll();
39 }
```

The code snippet shows a configuration for an `HttpSecurity` object. It starts by calling `.authorizeRequests()`, which then uses `.antMatchers()` to define paths that should be accessible to everyone (permitted all). Following this, `.anyRequest().authenticated()` is called, which means that any other request must be authenticated before being processed. This pattern of first permitting all and then requiring authentication is highlighted with a red box around the `.anyRequest().authenticated()` line.



## 회원가입

Username

Password

회원가입



# 회원가입

gildong

.....

### 회원가입

데이터베이스 필터 테이블 필터 ★

호스트: 127.0.0.1 데이터베이스: mydb 테이블: user 데이터 쿼리

mydb.user: 1 행 (총) (대략적)

id	username	password	enabled
3	gildong	\$2a\$10\$A88sn7M8PUZ57b2tM/gRWOeXvrPOsBXkV6k7/EGIssO1OrMBfQu8m	1

LocalDB

- authdb
- information\_schema
- mydb
  - board
  - role
  - user
  - user\_role
- mysql
- performance\_schema
- test

데이터베이스 콜드 데이터 데일을 콜드 ★

호스트: 127.0.0.1 데이터베이스: mydb 테이블: user\_role

user_id	role_id
3	1

LocalDB

- authdb
- information\_schema
- mydb
  - board
  - role
  - user
  - user\_role

```
WebSecurityConfig.java ✘ AccountController.java ✘ HomeController.java ✘ UserService.java ✘
42 @Configuration
43 public void configureGlobal(AuthenticationManagerBuilder auth)
44     throws Exception {
45     auth.jdbcAuthentication()
46         .dataSource(dataSource)
47         .passwordEncoder(passwordEncoder())
48         .usersByUsernameQuery("select username, password, enabled "
49             + "from user "
50             + "where username = ?")
51         .authoritiesByUsernameQuery("select u.username, r.name "
52             + "from user_role ur inner join user u on ur.user_id = u.id "
53             + "inner join role r on ur.role_id = r.id "
54             + "where u.username = ?");
55 }
56
@Bean
```

WebSecurityConfig > configureGlobal()



Search

Search

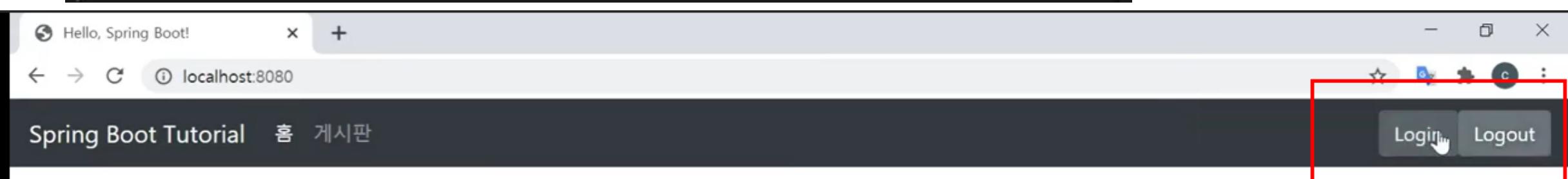
# Srping Boot Tutorial

Spring Boot을 이용해 웹 페이지 제작

```
<ul class="nav justify-content-end">
    <li class="nav-item dropdown">
        <a class="nav-link dropdown-toggle" href="#" id="dropdown01" data-toggle="dropdown"
           aria-haspopup="true" aria-expanded="false">Dropdown</a>
        <div class="dropdown-menu" aria-labelledby="dropdown01">
            <a class="dropdown-item" href="#">Action</a>
            <a class="dropdown-item" href="#">Another action</a>
            <a class="dropdown-item" href="#">Something else here</a>
        </div>
    </li> -->
</ul>
<form class="form-inline my-2 my-lg-0">
    <input class="form-control mr-sm-2" type="text" placeholder="Search" aria-label="Search">
    <button class="btn btn-secondary my-2 my-sm-0" type="submit">Search</button>
</form>
</div>
</nav>
</body>
</html>
```

```
</ui>
<form class="form-inline my-2 my-lg-0">
    <!-- <input class="form-control mr-sm-2" type="text" placeholder="Search" aria-label="Search" -->
    <button class="btn btn-secondary my-2 my-sm-0" type="submit">Search</button>
</form>
<div>
```

```
</ui>
<form class="form-inline my-2 my-lg-0">
    <!-- <input class="form-control mr-sm-2" type="text" placeholder="Search" aria-label="Search" -->
    <button class="btn btn-secondary my-2 my-sm-0" type="submit">Login</button>
</form>
<form class="form-inline my-2 my-lg-0">
    <!-- <input class="form-control mr-sm-2" type="text" placeholder="Search" aria-label="Search" -->
    <button class="btn btn-secondary my-2 my-sm-0" type="submit">Logout</button>
</form>
</div>
<v>
>
```

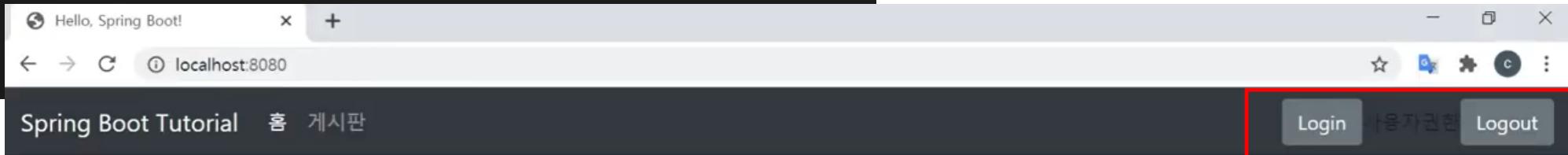


# Srpwing Boot Tutorial

Spring Boot을 이용해 웹 페이지 제작

```
<form class="form-inline my-2 my-lg-0">
    <!-- <input class="form-control mr-sm-2" type="text" placeholder="Search" aria-label="Search" -->
    <button class="btn btn-secondary my-2 my-sm-0" type="submit">Login</button>
</form>
<form class="form-inline my-2 my-lg-0">
    <!-- <input class="form-control mr-sm-2" type="text" placeholder="Search" aria-label="Search" -->
    <span>사용자</span>
    <span>권한</span>
    <button class="btn btn-secondary my-2 my-sm-0" type="submit">Logout</button>
</form>
</div>
</nav>
</body>
</html>
```

```
<a class="btn btn-secondary my-2 my-sm-0" th:href="@{/account/login}">Login</a>
<form class="form-inline my-2 my-lg-0">
    <!-- <input class="form-control mr-sm-2" type="text" placeholder="Search" aria-label="Search" -->
    <span>사용자</span>
    <span>권한</span>
    <button class="btn btn-secondary my-2 my-sm-0" type="submit">Logout</button>
</form>
</div>
```



# Srpwing Boot Tutorial

Spring Boot을 이용해 웹 페이지 제작  
Spring Security, JPA를 이용해 보안 설정과 데이터 다루기

```
<a class="btn btn-secondary my-2 my-sm-0" th:href="@{/account/login}">Login</a>
<form class="form-inline my-2 my-lg-0">
    <!-- <input class="form-control mr-sm-2" type="text" placeholder="Search" aria-label="Search" -->
    <span class="text-white">사용자</span>
    <span class="text-white mx-2">권한</span>
    <button class="btn btn-secondary my-2 my-sm-0" type="submit">Logout</button>
</form>
</div>
</v>
>
```



Hello, Spring Boot! x Thymeleaf x +

thymeleaf.org

Home Download Docs Ecosystem FAQ Forum Twitter GitHub

 Thymeleaf

29 October 2018: Version 3.0.4 of the *Thymeleaf + Spring Security* integration package has been published, including support for Spring Security 5 in both Spring MVC and Spring WebFlux applications. See the announcement.

thymeleaf.org/documentation.html

Home Download Docs Ecosystem FAQ Forum Twitter GitHub

 Thymeleaf

### Thymeleaf 3 Quick Links

- Thymeleaf 3.0 announcement and more info
- Thymeleaf 3.0 migration guide

- Standard URL Syntax
- Say Hello! Extending Thymeleaf in 5 minutes
- Say Hello Again! Extending Thymeleaf even more in another 5 minutes
- From HTML to HTML (via HTML)

## With Spring

- Bringing Thymeleaf and Natural Templates to the Spring PetClinic (Thymeleaf 2.1)
- [Spring MVC + Thymeleaf: how to access data from templates](#) thymeleaf.org/doc/articles/springsecurity.html
- [Thymeleaf + Spring Security integration basics](#)
- [Sending email in Spring with Thymeleaf](#)

## 4. Spring Security Dialect

In Spring MVC environments, the [Spring Security integration module](#) works as a replacement of the [Spring security taglib](#).

We use this dialect in the example in order to print the logged user credentials and to show different content to different roles.

The `sec:authorize` attribute renders its content when the attribute expression is evaluated to `true`:

```
<div sec:authorize="isAuthenticated()>
    This content is only shown to authenticated users.
</div>
<div sec:authorize="hasRole('ROLE_ADMIN')>
    This content is only shown to administrators.
</div>
<div sec:authorize="hasRole('ROLE_USER')>
    This content is only shown to users.
</div>
```

The `sec:authentication` attribute is used to print logged user name and roles:

# Thymeleaf - Spring Security integration modules

[Please make sure to select the branch corresponding to the version of Thymeleaf you are using]

## Status

This is a *Thymeleaf Extras* module, not a part of the Thymeleaf core (and as such following its own versioning schema), but fully supported by the Thymeleaf team.

This repository contains 3 projects:

- thymeleaf-extras-springsecurity3 for integration with Spring Security 3.x
- thymeleaf-extras-springsecurity4 for integration with Spring Security 4.x
- thymeleaf-extras-springsecurity5 for integration with Spring Security 5.x

Current versions:

- Version 3.0.4.RELEASE - for Thymeleaf 3.0 (requires Thymeleaf 3.0.10+)
- Version 2.1.3.RELEASE - for Thymeleaf 2.1 (requires Thymeleaf 2.1.2+)

← → C https://google.com/search?q=thymeleaf-extras-springsecurity5+maven&oq=thymeleaf-extras-springsecu

Google

thymeleaf-extras-springsecurity5 maven

All Images Videos News Shopping More Settings

About 4,980 results (0.29 seconds)

mvnrepository.com > artifact > thymeleaf-extras-spring...

**Thymeleaf Extras Springsecurity5 - Maven Repository**

Oct 12, 2018 - Modern server-side Java template engine for both web and standalone environments. License, Apache 2.0. Tags, security. Used By, 13 ...

Used By: 9 artifacts

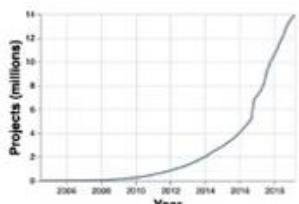
You visited this page on 7/28/20.

**3.0.4.RELEASE**  
... thymeleaf-extras-springsecurity5  
3.0.4.RELEASE

**3.0.3.RELEASE**  
... thymeleaf-extras-springsecurity5  
3.0.3.RELEASE



Indexed Artifacts (17.6M)



## Popular Categories

- Aspect Oriented
- Actor Frameworks
- Application Metrics
- Build Tools
- Bytecode Libraries
- Command Line Parsers
- Cache Implementations

Search for groups, artifacts, categories

Search

Home » org.thymeleaf.extras » thymeleaf-extras-springsecurity5



## Thymeleaf Extras Springsecurity5

Modern server-side Java template engine for both web and standalone environments

License

Apache 2.0

Tags

security

Used By

13 artifacts

<https://mvnrepository.com/artifact/org.thymeleaf.extras/thymeleaf-extras-springsecurity5/3.0.4.RELEASE>

## Thymeleaf Extras Springsecurity5 » 3.0.4.RELEASE

Modern server-side Java template engine for both web and standalone environments

Central (2) ICM (1)

Version

Repository

license

Apache 2.0

3.0.4.RELEASE

Central

Organization

The THYMELEAF team

3.0.3.RELEASE

Central

HomePage

<http://www.thymeleaf.org>

4

Date

(Oct 29, 2018)

Files

jar (46 KB) [View All](#)

Repositories

Central

Used By

13 artifacts

Maven

Gradle

SBT

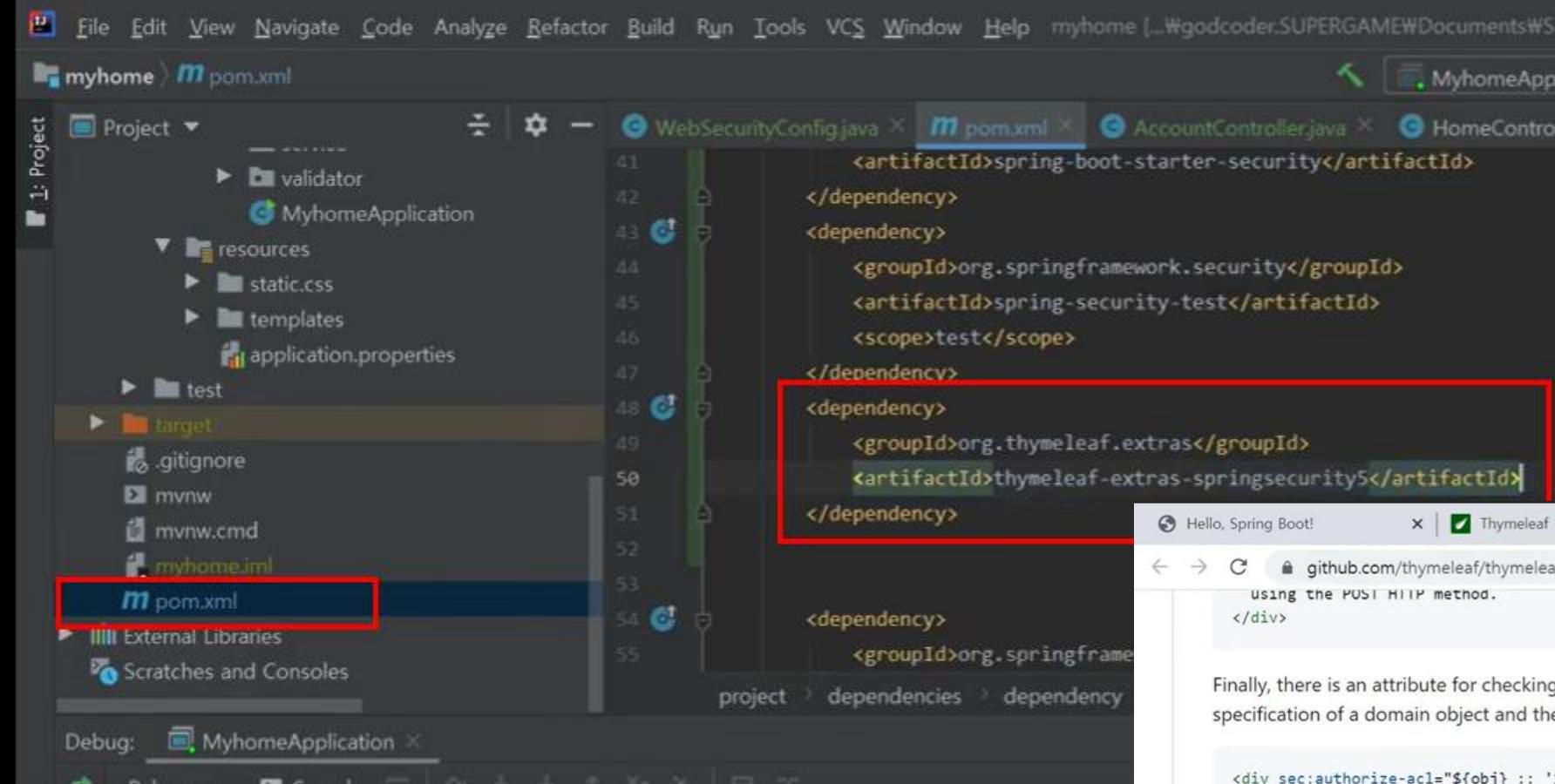
Ivy

Grape

Leiningen

Buildr

```
<!-- https://mvnrepository.com/artifact/org.thymeleaf.extras/thymeleaf-extras-springsecurity5 -->
<dependency>
    <groupId>org.thymeleaf.extras</groupId>
    <artifactId>thymeleaf-extras-springsecurity5</artifactId>
    <version>3.0.4.RELEASE</version>
</dependency>
```



Hello, Spring Boot!

Thymeleaf + Spring Security integration

github.com/thymeleaf/thymeleaf-extras-springsecurity

using the POST HTTP method.

</div>

Finally, there is an attribute for checking authorization using Spring Security's Access Control Lists, which needs the specification of a domain object and the permissions defined on it that we are asking for.

```
<div sec:authorize-acl="${obj} :: '1,3'">
    This will only be displayed if authenticated user has permissions "1" and "3"
    on domain object referenced by context variable "obj".
</div>
```

In this attribute, both domain object and permission specifications are considered to be thymeleaf *Standard Expressions*.

## Namespace

The namespace for all versions of this dialect is <http://www.thymeleaf.org/extras/spring-security>.

```
<html xmlns:sec="http://www.thymeleaf.org/extras/spring-security">
```

Getting the namespace incorrect won't impact processing of your template. It might however impact your IDE when it comes to things like suggestions/auto-completion in your templates.

```
login.html register.html common.html X  
templates > fragments > common.html > html  
1  <!DOCTYPE html>  
2  | html xmlns:th="http://www.thymeleaf.org" xmlns:sec="http://www.thymeleaf.org/extras/spring-security"  
3  | <head th:fragment="head(title)">  
4  |   <!-- Required meta tags -->  
5  |   <meta charset="utf-8">  
6  |   <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">  
7  |  
8  |   <!-- Bootstrap CSS -->  
9  |   <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/css/bootstrap.min.css"  
10 |     integrity="sha384-9AI2nRpC12Uk9gS9baDl411NOApFmC26EwAOH8WgZ15MYxFfc+NcPb1dKGi7Sk" crossorigin="a
```

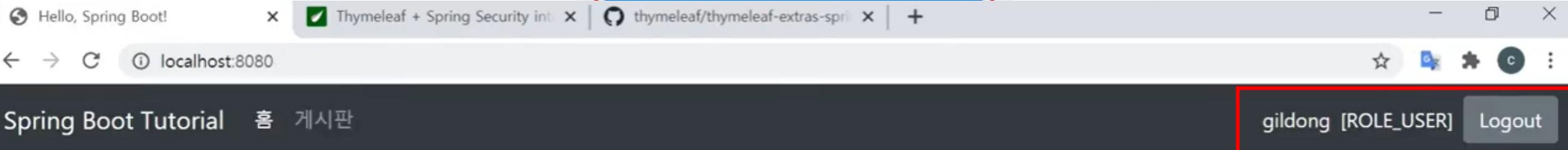
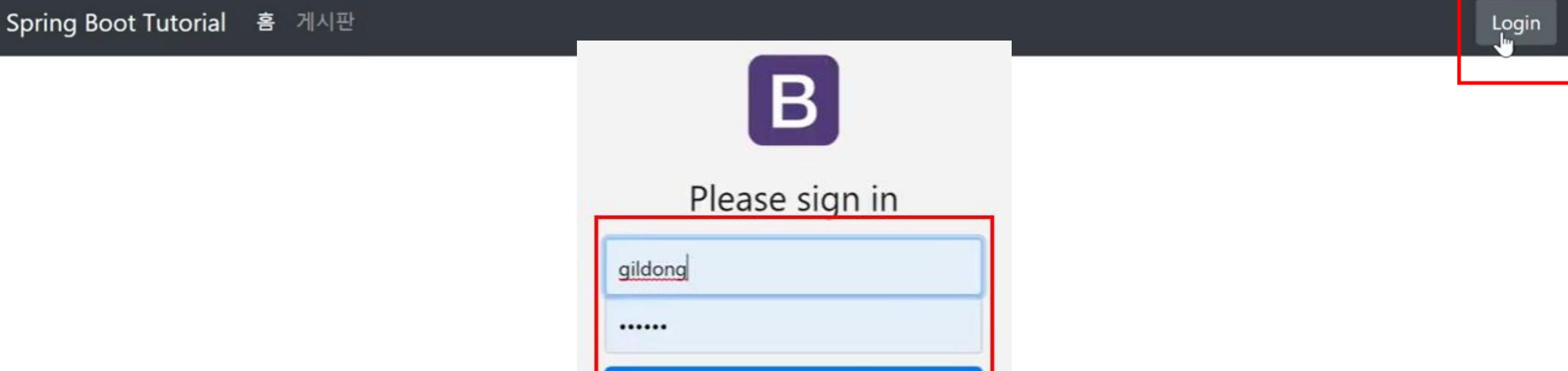
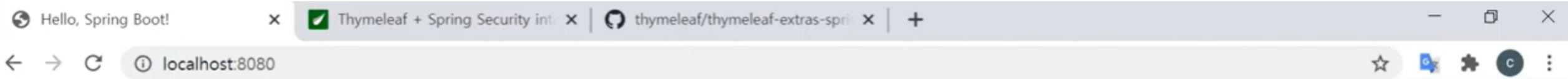
The **sec:authorize** attribute renders its content when the attribute expression is evaluated to **true**:

```
<div sec:authorize="isAuthenticated()>  
  This content is only shown to authenticated users.  
</div>  
<div sec:authorize="hasRole('ROLE_ADMIN')>  
  This content is only shown to administrators.  
</div>  
<div sec:authorize="hasRole('ROLE_USER')>  
  This content is only shown to users.  
</div>
```

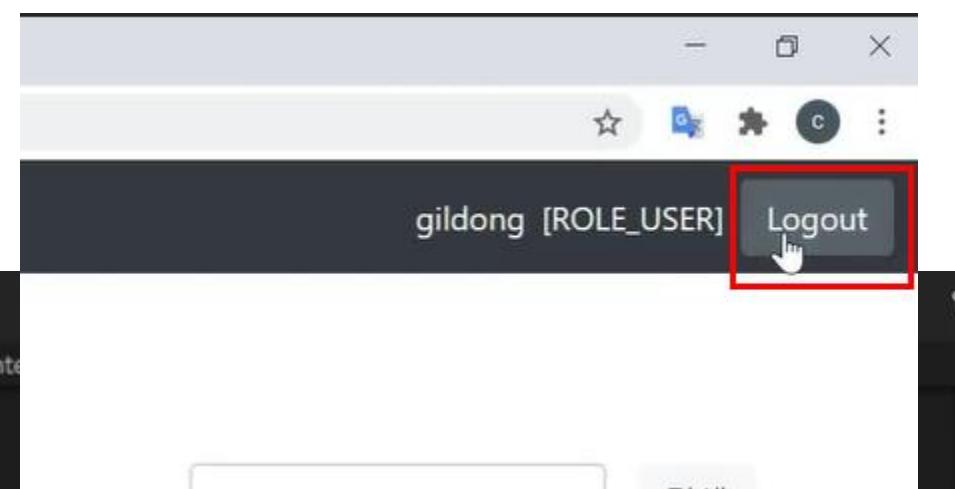
The **sec:authentication** attribute is used to print logged user name and roles:

```
Logged user: <span sec:authentication="name">Bob</span>  
Roles: <span sec:authentication="principal authorities">[ROLE_USER, ROLE_ADMIN]</span>
```

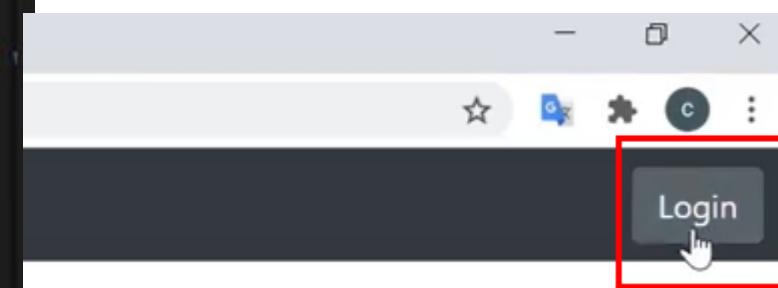
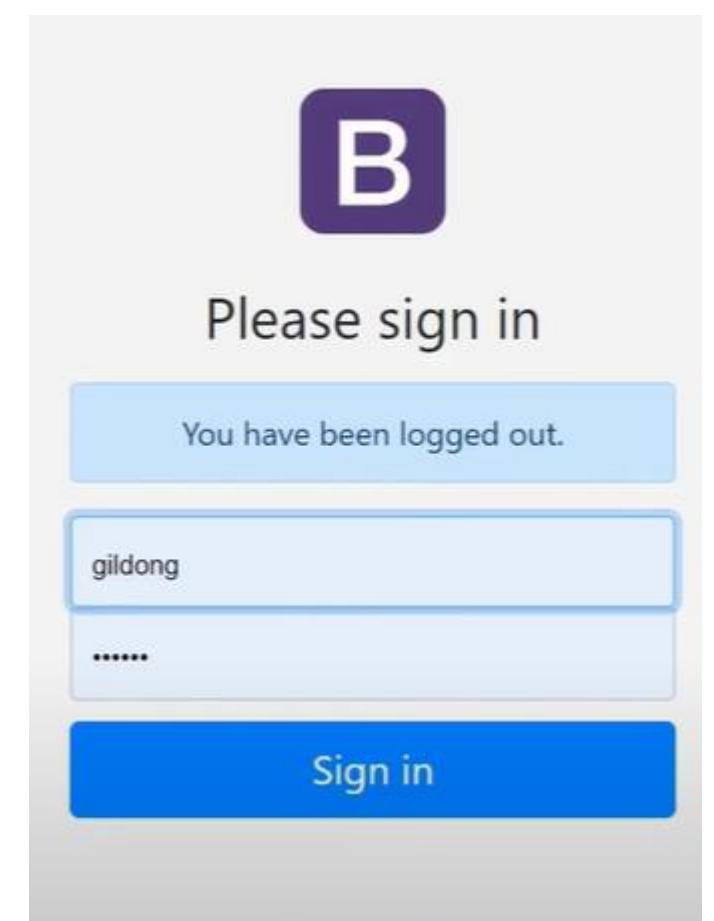
```
<a class="btn btn-secondary my-2 my-sm-0" th:href="@{/account/login}"  
sec:authorize="!isAuthenticated()">Login</a>  
<form class="form-inline my-2 my-lg-0" sec:authorize="isAuthenticated()">  
!--<input class="form-control mr-sm-2" type="text" placeholder="Search" aria-label="Search" />  
<span class="text-white" sec:authentication="name">사용자</span>  
<span class="text-white mx-2" sec:authentication="principal.authorities">권한</span>  
<button class="btn btn-secondary my-2 my-sm-0" type="submit">Logout</button>  
</form>
```



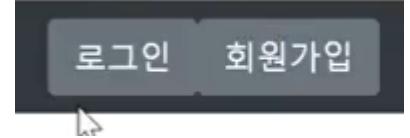
```
<a class="btn btn-secondary my-2 my-sm-0" th:href="@{/account/login}"  
    sec:authorize="!isAuthenticated()">Login</a>  
<form class="form-inline my-2 my-lg-0" th:action="@{/logout}" method="POST" sec:authorize="isA  
    <!-- <input class="form-control mr-sm-2" type="text" placeholder="Search" aria-label="Search-->  
    <span class="text-white" sec:authentication="name">사용자</span>  
    <span class="text-white mx-2" sec:authentication="principal authorities">권한</span>  
    <button class="btn btn-secondary my-2 my-sm-0" type="submit">Logout</button>  
</form>  
</div>
```



```
login.html X register.html common.html  
templates > account > login.html > html > body.text-center  
13  
14     <title>Login</title>  
15 </head>  
16  
17 <body class="text-center">  
18     <form class="form-signin" th:action="@{/account/login}" method="post">  
19           
20         <h1 class="h3 mb-3 font-weight-normal">Please sign in</h1>  
21         <div th:if="${param.error}" class="alert alert-danger" role="alert">  
22             Invalid username and password.  
23         </div>  
24         <div th:if="${param.logout}" class="alert alert-primary" role="alert">  
25             You have been logged out.  
26         </div>
```



```
<a class="btn btn-secondary my-2 my-sm-0" th:href="@{/account/login}"  
    sec:authorize="!isAuthenticated()">로그인</a>  
<a class="btn btn-secondary my-2 my-sm-0" th:href="@{/account/register}"  
    sec:authorize="!isAuthenticated()">회원가입</a>  
  
<form class="form-inline my-2 my-lg-0" th:action="@{/logout}" method="POST" sec:authorize="isA  
    <!-- <input class="form-control mr-sm-2" type="text" placeholder="Search" aria-label="Sear  
    <span class="text-white" sec:authentication="name">사용자</span>  
    <span class="text-white mx-2" sec:authentication="principal authorities">권한</span>  
    <button class="btn btn-secondary my-2 my-sm-0" type="submit">Logout</button>  
</form>  
</div>  
</nav>  
</body>  
</html>
```



```
<a class="btn btn-secondary my-2 my-sm-0" th:href="@{/account/login}"  
    sec:authorize="!isAuthenticated()">로그인</a>  
<a class="btn btn-secondary my-2 my-sm-0" th:href="@{/account/register}"  
    sec:authorize="!isAuthenticated()">회원가입</a>
```

