Securing a Web Application

This guide walks you through the process of creating a simple web application with resources that are protected by Spring Security.

What you'll build

You'll build a Spring MVC application that secures the page with a login form backed by a fixed list of users.

What you'll need

- About 15 minutes
- A favorite text editor or IDE
- JDK 1.6 or later
- Gradle 1.11+ or Maven 3.0+
- You can also import the code from this guide as well as view the web page directly into Spring Tool Suite (STS) and work your way through it from there.

How to complete this guide

Like most Spring Getting Started guides, you can start from scratch and complete each step, or you can bypass basic setup steps that are already familiar to you. Either way, you end up with working code.

To **start from scratch**, move on to Build with Gradle.

To **skip the basics**, do the following:

- Download and unzip the source repository for this guide, or clone it using Git:

  `git clone https://github.com/spring-guides/gs-securing-web.git`
- cd into `gs-securing-web/initial`
- Jump ahead to Set up Spring Security.

When you're finished, you can check your results against the code in `gs-securing-web/complete` .

Build with Gradle

Build with Maven

Build with Spring Tool Suite

Create an unsecured web application

Before you can apply security to a web application, you need a web application to secure. The steps in this section walk you through creating a very simple web application. Then you secure it with Spring Security in the next section.

The web application includes two simple views: a home page and a "Hello World" page. The home page is defined in the following Thymeleaf template:

`src/main/resources/templates/home.html`

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml" xmlns:th="http://www.thymeleaf.org" xmlns:sec="http://www.thymeleaf.
    <head>
        <title>Spring Security Example</title>
    </head>
```