

# Spring Boot Web Application

---



AUBURN UNIVERSITY

---

---

**CPSC 4970 Applied Cyber Security**



# Why Spring Boot?

---

- Extremely fast to create web applications
- Eliminates complexities of build Java Web App directly (war file)
- Production grade software
- Eliminates “rolling your own” for many web application common capabilities
  - Security
  - Form Handling
  - Database & Object to Relationship Mapping
  - APIs
- Convention over Configuration
  - Minimal configuration files
  - Extensive use of Java Annotations
- Embedded Tomcat or Jetty web server
- Requirements
  - Java JDK
  - Maven or Gradle

# Spring Boot\_INITIALIZER

<https://start.spring.io/>



AUBURN



## Project

☒ Maven Project

☐ Gradle Project

## Language

☒ Java

☐ Kotlin

☐ Groovy

## Spring Boot

☐ 3.0.0 (SNAPSHOT)

☐ 3.0.0 (M3)

☐ 2.7.1 (SNAPSHOT)

☒ 2.7.0

☐ 2.6.9 (SNAPSHOT)

☐ 2.6.8

## Project Metadata

Group

Artifact

Name

Description

Package name

Packaging ☒ Jar ☐ War

Java ☐ 18 ☒ 17 ☐ 11 ☐ 8

## Dependencies

ADD DEPENDENCIES... ⌘ + B

### 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.

GENERATE ⌘ + ↵

EXPLORE CTRL + SPACE









SHARE...



# Spring Boot Maven Plugin

```
<build>
  <plugins>
    <plugin>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-maven-plugin</artifactId>
    </plugin>
  </plugins>
</build>
```

- Maven Tasks/Goals specific for Spring Boot Application
- Run Embedded web server
- Package into a single executable jar file

```
▼  spring-boot (org.springframework.boot:spring-boot)
   spring-boot:build-image
   spring-boot:build-info
   spring-boot:help
   spring-boot:repackage
   spring-boot:run
   spring-boot:start
   spring-boot:stop
```



# Simple Web App - All the Code

Annotation:  
HTTP Request  
Handler Class

Annotation: Spring  
Boot App Class

Annotation Indicating  
method to handle  
URL path

Spring Application  
Startup

```
import ...

@RestController
@SpringBootApplication
public class WebappApplication {

    @RequestMapping("/")
    public String indexPage() { return "War Eagle!!!!!!"; }

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



# Application Properties File

---

- Key configuration file
  - application.yaml or application.properties
  - located on classpath
- Configuration of application configuration values
  - Example configuring embedded server can be as easy as:

## Properties Syntax

```
server.address=127.0.0.1  
server.port=5566
```

## YAML Syntax

```
server:  
  address: 127.0.0.1  
  port: 5566
```



# Build Executable JAR File

---

- Package into single jar file
  - “**mvn package**” command
  - Spring Boot will repackage it into an executable one.
- Run jar file
  - “**java -jar target/webapp-0.0.1-SNAPSHOT.jar**”
- Runs embedded web server on default port
  - <http://localhost:8080>
- The jar is completely self-contained, you can deploy and run it anywhere (with Java)
  - Ideal for microservices and docker containers



AUBURN UNIVERSITY

---