

# Spring Session - WebFlux with Custom Cookie

Eleftheria Stein-Kousathana

Version 2.3.0.RELEASE

# Table of Contents

Spring Boot Configuration .....	2
webflux-custom-cookie Sample Application .....	3
Running the webflux-custom-cookie Sample Application .....	3
Exploring the webflux-custom-cookie Sample Application .....	3

This guide describes how to configure Spring Session to use custom cookies in a WebFlux based application. The guide assumes you have already set up Spring Session in your project using your chosen data store. For example, [HttpSession with Redis](#).

**NOTE**

You can find the completed guide in the [WebFlux Custom Cookie sample application](#).

[Index](#)

# Spring Boot Configuration

Once you have set up Spring Session, you can customize how the session cookie is written by exposing a `WebSessionIdResolver` as a Spring bean. Spring Session uses a `CookieWebSessionIdResolver` by default. Exposing the `WebSessionIdResolver` as a Spring bean augments the existing configuration when you use configurations like `@EnableRedisHttpSession`. The following example shows how to customize Spring Session's cookie:

```
@Bean
public WebSessionIdResolver webSessionIdResolver() {
    CookieWebSessionIdResolver resolver = new CookieWebSessionIdResolver();
    resolver.setCookieName("JSESSIONID"); ①
    resolver.addCookieInitializer((builder) -> builder.path("/")); ②
    resolver.addCookieInitializer((builder) -> builder.sameSite("Strict")); ③
    return resolver;
}
```

- ① We customize the name of the cookie to be `JSESSIONID`.
- ② We customize the path of the cookie to be `/` (rather than the default of the context root).
- ③ We customize the `SameSite` cookie directive to be `Strict`.

# webflux-custom-cookie Sample Application

This section describes how to work with the `webflux-custom-cookie` sample application.

## Running the `webflux-custom-cookie` Sample Application

You can run the sample by obtaining the [source code](#) and invoking the following command:

```
$ ./gradlew :spring-session-sample-boot-webflux-custom-cookie:bootRun
```

### NOTE

For the sample to work, you must [install Redis 2.8+](#) on localhost and run it with the default port (6379). Alternatively, you can update the `RedisConnectionFactory` to point to a Redis server. Another option is to use [Docker](#) to run Redis on localhost. See [Docker Redis repository](#) for detailed instructions.

You should now be able to access the application at <http://localhost:8080/>

## Exploring the `webflux-custom-cookie` Sample Application

Now you can use the application. Fill out the form with the following information:

- **Attribute Name:** `username`
- **Attribute Value:** `rob`

Now click the **Set Attribute** button. You should now see the values displayed in the table.

If you look at the cookies for the application, you can see the cookie is saved to the custom name of `JSESSIONID`.