

EXPERT GUIDANCE FOR
GOOGLE CLOUD PLATFORM

When migrating apps

select to get started

Spring MVC - WebMvcConfigurerAdapter Examples

Code Snippets Spring MVC

Extending WebMvcConfigurerAdapter to register Custom AnnotationFormatterFactory

```
@EnableWebMvc
@Configuration
@Import(MyViewConfig.class)
public class MyWebConfig extends WebMvcConfigurerAdapter {

    @Override
    public void addFormatters (FormatterRegistry registry) {
        AddressFormatAnnotationFormatterFactory factory = new
            AddressFormatAnnotationFormatterFactory();

        registry.addFormatterForFieldAnnotation(factory);
    }

    @Bean
    public CustomerController customerController () {
        return new CustomerController();
    }

    @Bean
    public CustomerDataService userService () {
        return new InMemoryCustomerDataService();
    }
}
```

[Original Post](#)



Extending WebMvcConfigurerAdapter to add Custom Converter

```
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
import org.springframework.context.annotation.Import;
import org.springframework.format.FormatterRegistry;
import org.springframework.web.servlet.config.annotation.EnableWebMvc;
import org.springframework.web.servlet.config.annotation.WebMvcConfigurerAdapter;
@EnableWebMvc
@Configuration
```

```
@Import(MyViewConfig.class)
public class TradeConfig extends WebMvcConfigurerAdapter {

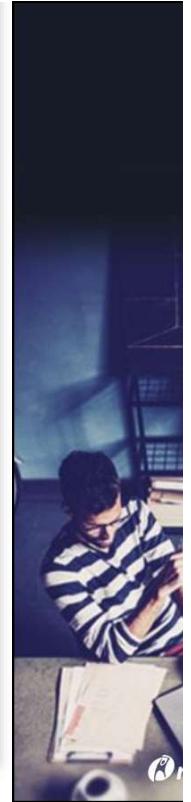
    @Bean
    public TradeController tradeParamRequestController() {
        return new TradeController();
    }

    @Bean
    public TradeService tradeService(){
        return new TradeService();
    }

    @Override
    public void addFormatters (FormatterRegistry registry) {
        registry.addConverter(new TradeIdToTradeConverter(tradeService()));
    }

}
```

[Original Post](#)



Extending WebMvcConfigurerAdapter to add custom Formatter

```
@EnableWebMvc
@Configuration
@Import(MyViewConfig.class)
public class MyWebConfig extends WebMvcConfigurerAdapter {

    @Override
    public void addFormatters (FormatterRegistry registry) {
        AddressFormatter addressFormatter = new AddressFormatter();
        addressFormatter.setStyle(AddressFormatter.Style.REGION);
        registry.addFormatter(addressFormatter);
    }

    @Bean
    public CustomerController customerController () {
        return new CustomerController();
    }

    @Bean
    public CustomerDataService userService () {
        return new InMemoryCustomerDataService();
    }
}
```

[Original Post](#)

Registering a custom Interceptor

```
@EnableWebMvc
@Configuration
public class MyWebConfig extends WebMvcConfigurerAdapter {

    @Override
    public void addInterceptors(InterceptorRegistry registry) {
        registry.addInterceptor(new LoginInterceptor());
    }

    @Bean
    public AppController tradeController() {
        return new AppController();
    }

    @Bean
    public UserService userService() {
        return new InMemoryUserService();
    }

    @Bean
    public ViewResolver viewResolver() {
        InternalResourceViewResolver viewResolver = new InternalResourceViewResolver();
        viewResolver.setPrefix("/WEB-INF/views/");
        viewResolver.setSuffix(".jsp");
        return viewResolver;
    }
}
```

[Original Post](#)

```
@SpringBootApplication
public static class BootApplication extends WebMvcConfigurerAdapter {
    @Override
    public void addInterceptors(InterceptorRegistry registry) {
        registry.addInterceptor(new MyAsyncHandlerInterceptor());
    }
}
```

[Original Post](#)**Java 10 Tutor**[Java 10 Features](#)**Java 9 Tutor**[Java 9 Module Sys](#)[Java 9 Misc Featu](#)[Java 9 JShell](#)**Recent Tutorials**[JPQL LEFT OUTER](#)

Using a custom TaskExecutor for async processing.

```
package com.logicbig.example;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.context.annotation.Bean;
```

```

import org.springframework.scheduling.concurrent.ThreadPoolTaskExecutor;
import org.springframework.web.servlet.config.annotation.AsyncSupportConfigurer;
import org.springframework.web.servlet.config.annotation.WebMvcConfigurer;
import org.springframework.web.servlet.config.annotation.WebMvcConfigurerAdapter;

@SpringBootApplication
public class AsyncConfigExample {

    @Bean
    WebMvcConfigurer configurer () {
        return new WebMvcConfigurerAdapter() {
            @Override
            public void configureAsyncSupport (AsyncSupportConfigurer configurer) {
                ThreadPoolTaskExecutor t = new ThreadPoolTaskExecutor();
                t.setCorePoolSize(10);
                t.setMaxPoolSize(100);
                t.setQueueCapacity(50);
                t.setAllowCoreThreadTimeOut(true);
                t.setKeepAliveSeconds(120);
                t.initialize();
                configurer.setTaskExecutor(t);
            }
        };
    }

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

```

[Original Post](#)

Adding custom location for static pages in a boot application.

```

package com.logicbig.example;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.EnableAutoConfiguration;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.context.annotation.Bean;
import org.springframework.web.servlet.config.annotation.ResourceHandlerRegistry;
import org.springframework.web.servlet.config.annotation.WebMvcConfigurer;
import org.springframework.web.servlet.config.annotation.WebMvcConfigurerAdapter;

@EnableAutoConfiguration
public class AddCustomLocations {

    @Bean
    WebMvcConfigurer configurer () {
        return new WebMvcConfigurerAdapter() {
            @Override
            public void addResourceHandlers (ResourceHandlerRegistry registry) {
                registry.addResourceHandler("/pages/**").
                    addResourceLocations("classpath:/my-custom-location/");
            }
        };
    }
}

```

[Java Swing - Edit](#)
[Java Swing - Edit](#)
[Spring MVC - App](#)
[Spring MVC - Ses](#)
[Class Based Proxy](#)
[Java Swing - JTab](#)
[JPQL INNER JOIN](#)
[Quick Intro to Spi](#)
[and How to enabl](#)
[Java Swing - A ge](#)
[JMenuBar or JPop](#)
[JTable/JTree](#)
[Spring MVC - Writ](#)
[body](#)
[Spring MVC - Req](#)
[Spring MVC - Usir](#)
[Method Argument](#)
[Java Swing - Cre](#)
[ChangeEvent](#)
[Understanding Tir](#)
[Using Literals in J](#)
[Eliminating Duplic](#)
[DISTINCT statem](#)
[ORDER BY ASC/D](#)
[SELECT-FROM-WH](#)
[Integrating JAX-R](#)
[Application](#)
[Java 10 - java.uti](#)
[Java 10 - Stream](#)
[method to accum](#)
[Java 10 - Collectio](#)
[Unmodifiable Coll](#)
[Spring MVC - URI](#)
[Java 10 - Local-Vi](#)
[Examples](#)
[Cache Store Mode](#)
[Cache Retrieve Mi](#)
[PrimeFaces - HotF](#)
[Spring Cache - Us](#)
[register Custom K](#)
[Spring Cache - Cr](#)
[Spring Cache Evi](#)
[Spring Cache Key](#)
[Understating Spr](#)
[example](#)
[Java 9 Modules -](#)
[Accessing JDK Int](#)
[Spring Boot - Map](#)
[application outsid](#)
[UNSPECIFIED anc](#)
[SharedCacheMode](#)
[Using DISABLE_S](#)
[Using ENABLE_SE](#)
[Accessing Second](#)
[interface](#)
[Enabling Second-](#)

```

    }
}

public static void main (String[] args) {

    SpringApplication app =
        new SpringApplication(AddCustomLocations.class);
    app.run(args);
}
}

```

[Original Post](#)

Amazon CloudFront CDN - Content Delivery Network

Ad Highly secure & flexible CDN. Integrated with key AWS services. Try now!

aws.amazon.com

[Learn more](#)

[Java - Widening a Conversions](#)

[Spring Boot - Retr Custom ErrorCont](#)

[Spring Boot - Usir ErrorController](#)

[Servlet - Applying](#)

[Spring Boot - Map controller by impl](#)

[Spring MVC - URI setUseTrailingSlas RequestMappingH](#)

[Spring MVC - URI behavior, Using setUseRegistered! RequestMappingH](#)

[Java Swing - JTRe Example](#)

[Java Swing - How Nodes of JTree?](#)

[Spring MVC - URI behavior, Using se method of Reques](#)

[PrimeFaces - Inde](#)

[Auto applying con attribute type](#)

[Type Conversion i key or value](#)

[Type Conversion i](#)

[Type Conversion i](#)

[Entity Attribute Cr](#)

[Spring Boot - Map Codes to a templ](#)

[Spring Boot - Map Codes to Static H](#)

[Spring Boot - Map Codes to Custom](#)

[Spring MVC - Unit](#)

[Spring MVC - Map PATCH request to](#)

[Spring MVC - Han](#)

[Spring MVC - Unit](#)

[Spring MVC - Map PUT request to Ja](#)

[Spring MVC - Map PUT request to Ja](#)

[Using Entity Subg](#)

[Spring Boot - Cus](#)

[Java Swing - Und Creating JTree fro](#)

[Java Swing - Usin JTTree](#)

[Java Swing - Crea listening to select](#)

[Spring MVC - Han](#)

[PrimeFaces - Ajax](#)

[Understanding En @NamedEntityGr Example](#)

```

@Bean
public WebMvcConfigurer configurer () {
    return new WebMvcConfigurerAdapter() {
        @Override
        public void addInterceptors (InterceptorRegistry registry) {
            TzRedirectInterceptor interceptor = new TzRedirectInterceptor();
            InterceptorRegistration i = registry.addInterceptor(interceptor);
            i.excludePathPatterns("/tzHandler", "/tzValueHandler");
        }
    };
}

```

[Original Post](#)

WebMvcConfigurerAdapter#addResourceHandlers adds a resource handler for serving static resources based on the specified URL path patterns. ResourceHandlerRegistry stores the registration of resources.

```

@Bean
public WebMvcConfigurerAdapter webConfigurer () {
    return new WebMvcConfigurerAdapter() {
        @Override
        public void addResourceHandlers (ResourceHandlerRegistry registry) {
            registry.addResourceHandler("/static/**")
                .addResourceLocations("/static/");
        }
    };
}

```

[Original Post](#)

[Spring Boot - Cus](#)[Java Swing - JTab
backend and @Di
columns/rows via](#)[Java Swing - JTab](#)[Spring MVC - Test
MockMultipartFile](#)[Spring MVC - File
StandardServletM](#)[Spring MVC - Und](#)[Bean Validation ir](#)[Spring Boot - Usir](#)[Spring Boot - Usir](#)[PrimeFaces - Cli](#)[Spring Security -](#)[How to configure
HTTPS \(SSL/TLS\)](#)[How to configure
\(SSL/TLS\)](#)[JPA - Entity Audit](#)[JPA - Defining Ent
@EntityListeners](#)[Spring MVC - Ren
with SessionTher](#)[Spring MVC - Ren
with CookieTheme](#)[PrimeFaces - Data](#)[Java Swing - JTab](#)[Jackson JSON - U
constructors or fa
deserialization.](#)[Spring Boot - Usir](#)[Spring MVC - Cus
ResourceBundleTh
a Bean](#)[Spring MVC - The](#)[Spring MVC - Cus
FixedThemeResol](#)[JPA - Basic Auditii
Listeners](#)[JPA - Understandi
Methods](#)[Installing and seti](#)[Java - Phaser](#)[Spring Boot - Usir](#)[Spring MVC - Wor
Settings](#)[Java Swing - JTab](#)[Spring MVC - Mus](#)[Jackson JSON - U
@JsonIgnoreProp](#)[Jackson JSON - U
@JsonSetter and
property names](#)[PrimeFaces - Bloc](#)[Spring ORM - Spr
transaction](#)

```
@SpringBootApplication
public class Main {

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

    @Bean
    public WebMvcConfigurerAdapter webConfigurer () {
        return new WebMvcConfigurerAdapter() {
            @Override
            public void addResourceHandlers (ResourceHandlerRegistry registry) {
                registry.addResourceHandler("/static/**")
                    .addResourceLocations("/static/")
                    .setCachePeriod(30);
            }
        };
    }
}
```

Original Post



```
@SpringBootApplication
public class Main extends WebMvcConfigurerAdapter {

    @Override
    public void addViewControllers (ViewControllerRegistry registry) {
        ViewControllerRegistration r = registry.addViewController("/test");
        r.setViewName("myView");
        r.setStatusCode(HttpStatus.GONE);
    }

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

```
@SpringBootApplication
public class Main extends WebMvcConfigurerAdapter {

    @Override
    public void addViewControllers (ViewControllerRegistry registry) {
        registry.addStatusController("/test2", HttpStatus.SERVICE_UNAVAILABLE);
    }
}
```

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

```
@SpringBootApplication
public class Main extends WebMvcConfigurerAdapter {

    @Override
    public void addViewControllers (ViewControllerRegistry registry) {
        RedirectViewControllerRegistration r =
            registry.addRedirectViewController("/test", "/test2");
        //r.setStatusCode(HttpStatus.SEE_OTHER);
        //r.setContextRelative(false)
    }

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



```
@SpringBootApplication
public class Main extends WebMvcConfigurerAdapter {

    @Override
    public void addViewControllers (ViewControllerRegistry registry) {

        //mapping url to a view
        ViewControllerRegistration r = registry.addViewController("/test");
        r.setViewName("myView");
        //setting status code
        r.setStatusCode(HttpStatus.GONE);

        //mapping another url to a status code
        registry.addStatusController("/test2", HttpStatus.SERVICE_UNAVAILABLE);
    }

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

[Original Post](#)

- [Spring Boot - Reg ConfigurableWebE](#)
- [Java Swing - Is it of an unused Com](#)
- [Java - Eligibility F](#)
- [Java Swing - Gro \(JToggleButton\) f](#)
- [Spring MVC - Cus ConfigurableWebE](#)
- [JAX-RS - Applying](#)
- [JPA - PessimisticL](#)
- [JPA - Using Multip](#)
- [JPA - Using Pessir](#)
- [PrimeFaces - Usin Feed](#)
- [PrimeFaces - Dial](#)
- [Using WebSocket:](#)
- [JPA - When to use OPTIMISTIC_FOR](#)
- [JPA - Using OPTIN mode to force ver](#)
- [JPA - Explicitly se](#)
- [Spring Boot - Seti Resources](#)
- [Java 9 - Input Ou](#)
- [JPA - Updating En Attribute](#)
- [Java Swing - Addi Container](#)
- [Spring - RestTempl](#)
- [PrimeFaces - Pass Page by using dia](#)
- [JPA - OptimisticLc](#)
- [JPA - Optimistic L annotation](#)
- [Using Spring 5.0 ' methods](#)
- [Java Swing - Editi](#)
- [Spring Security - using Persistent T](#)
- [JAX-RS - Injecting @Context annota](#)
- [Registering a Cus](#)
- [Understanding Sp Configuration](#)
- [Spring MVC - Cus @RequestParam e](#)
- [Jackson JSON - J@ Configuration Exa](#)
- [JPA - Using Persis EntityManager.co a detached entity](#)
- [JPA - Getting Enti PersistenceUtil.isl](#)
- [JPA - Getting Enti EntityManager.co](#)

[Spring Security - Authentication using TokenBasedReme](#)

[PrimeFaces - Sending RequestContext object](#)

[Java Swing - JComboBox With Custom Popups](#)

[Java 9 - Regex In Java](#)

[JPA - Efficiently Performing associations with EntityManager.get](#)

[JPA - Efficiently Performing association with EntityManager.get](#)

[JPA - Getting EntityManager.get](#)

[JAX-RS - Injecting @Context annotations](#)

[Jackson JSON parser](#)

[Spring MVC - Form Validator Interface](#)

[Java Swing - JComboBox Highlighting Example](#)

[Spring MVC - Form Validation API and Annotations](#)

[JPA - Detached EntityManager](#)

[Java 9 - Platform](#)

[Java Util Logging](#)

[Spring ORM - Using LocalContainerEntityManager setting up JPA EntityManager Programmatically](#)

[PrimeFaces - Open Content Programming](#)

[JAX-RS - Configuration](#)

[JAX-RS - Resources](#)

[JPA - Merging Detached Entities](#)

[JPA - Detaching a Persistence Context](#)

[JPA - Refreshing a Detached Entity](#)

[Java Swing - JComboBox](#)

[Java Swing - JComboBox ItemListener and Listener](#)

[Performance differences between String based replication](#)

[Spring - Message Codes Resource error messages](#)

[Spring Boot - Bean registration](#)

[Java Swing - JList with HTML Example](#)

[Spring ORM - Implementing application](#)

[Spring JPA Support LocalEntityManager](#)

[Java Swing - JList Example](#)

[Java Swing - JList with Listener](#)

```
@SpringBootApplication
public class Main extends WebMvcConfigurerAdapter {

    @Override
    public void addViewControllers (ViewControllerRegistry registry) {
        RedirectViewControllerRegistration r =
            registry.addRedirectViewController("/test", "/test2");
        //r.setStatusCode(HttpStatus.SEE_OTHER);
        //r.setContextRelative(false)
    }

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

[Original Post](#)

Registering a custom HttpMessageConverter

```
import org.springframework.context.annotation.ComponentScan;
import org.springframework.http.converter.HttpMessageConverter;
import org.springframework.web.servlet.config.annotation.EnableWebMvc;
import org.springframework.web.servlet.config.annotation.WebMvcConfigurerAdapter;

import java.util.List;

@EnableWebMvc
@ComponentScan("com.logicbig.example")
public class AppConfig extends WebMvcConfigurerAdapter {

    @Override
    public void extendMessageConverters (List<HttpMessageConverter<?>> converters) {
        converters.add(new YamlHttpMessageConverter());
    }
}
```

[Original Post](#)

```
@EnableWebMvc
@Configuration
public class AppConfig extends WebMvcConfigurerAdapter {

    @Bean
    public SimpleUrlHandlerMapping simpleUrlHandlerMapping () {
        SimpleUrlHandlerMapping mapping = new SimpleUrlHandlerMapping();
```

```

        Map<String, Object> map = new HashMap<>();
        map.put("/app", httpRequestHandler());
        mapping.setUrlMap(map);
        mapping.setInterceptors(new Object[]{new MyInterceptor()});
        return mapping;
    }

    @Bean
    HttpRequestHandler httpRequestHandler () {
        return new MyHttpRequestHandler();
    }

    @Override
    public void addInterceptors (InterceptorRegistry registry) {
        InterceptorRegistration ir = registry.addInterceptor(new MyInterceptor());
    }

    @Bean
    ExampleController controller () {
        return new ExampleController();
    }
}

```

[Original Post](#)

Java Swing - JList
[ListSelectionListener](#)
 Java Swing - List
[Spring Support for Java Swing](#)
[Java 9 - java.util.List](#)
[Understanding Spring Data Access Exceptions](#)
[JUnit 5 Quick Features](#)
[Example](#)
[PrimeFaces - Dialog Example](#)
[PrimeFaces - Dialog Example](#)
[Spring Boot - Property configuration selection](#)
[Java 9 Modules - opens clauses](#)
[Spring Boot - Sets and logging.path](#)
[Java 9 - ProcessHandle](#)
[Example](#)
[Spring 5 - WebClient](#)
[Understanding Java 9 Example](#)
[Spring 5 WebFlux Functional Endpoints](#)
[Spring 5 WebFlux Getting Started Example](#)
[Java 9 - java.util.Stream API](#)
[Spring MVC - JSON Processing](#)
[Understanding Java 9 Stream API](#)
[Spring Boot - Custom Format](#)
[Java 9 Try-With-Resources](#)
[Java 9 Modules -](#)
[Java 9 StackWalker and SHOW_REFLECTION](#)
[Java 9 StackWalker](#)
[Java 9 StackWalker](#)
[Java 9 Modules - Structure](#)
[JShell - Tab Completion](#)
[Java 9 Modules - public static providers](#)
[Java 9 Stream API](#)
[Java 9 Collection API](#)
[PrimeFaces - Basic Components](#)
[Spring MVC - Annotations](#)
[Spring MVC - RSS Feeds](#)
[Java 9 Modules - Automatic Module Discovery](#)
[Java 9 Modules - Modules](#)
[Java 9 Modules - example](#)
[Java 9 Modules -](#)

```

@EnableWebMvc
@Configuration
@ComponentScan
public class MyWebConfig extends WebMvcConfigurerAdapter {

    @Override
    public void configureContentNegotiation (ContentNegotiationConfigurer configurer) {
        configurer.favorParameter(true);
        // configurer.ignoreAcceptHeader(true);
    }
}

```

[Original Post](#)

[Java 9 Try-With-Resources](#)
[Java 9 Modules -](#)
[Java 9 StackWalker and SHOW_REFLECTION](#)
[Java 9 StackWalker](#)
[Java 9 StackWalker](#)
[Java 9 Modules - Structure](#)
[JShell - Tab Completion](#)
[Java 9 Modules - public static providers](#)
[Java 9 Stream API](#)
[Java 9 Collection API](#)
[PrimeFaces - Basic Components](#)
[Spring MVC - Annotations](#)
[Spring MVC - RSS Feeds](#)
[Java 9 Modules - Automatic Module Discovery](#)
[Java 9 Modules - Modules](#)
[Java 9 Modules - example](#)
[Java 9 Modules -](#)

```

@EnableWebMvc
@Configuration
@ComponentScan
public class MyWebConfig extends WebMvcConfigurerAdapter {

    @Override
    public void configureContentNegotiation (ContentNegotiationConfigurer configurer) {
        configurer.defaultContentType(MediaType.APPLICATION_JSON);
    }
}

```

[Original Post](#)

- [Java 9 Modules - Providers](#)
- [Spring - Passing J](#)
- [PrimeFaces - Sele](#)
- [Java 9 JShell Gett](#)
- [Spring Boot - Log logging configura](#)
- [Using SLF4J/JUL v](#)
- [Java 9 Modules - .](#)
- [Spring MVC - Jas](#)
- [Java 9 Modules w](#)
- [Spring - Returning SimpleJdbcCall](#)
- [Connecting Oracle](#)
- [Java 9 Modules w](#)
- [PrimeFaces - Addi](#)
- [using RequestCon](#)
- [Java 9 Modules Q](#)
- [Using SLF4J with](#)
- [Redirecting Java l](#)
- [Log4j2 with jul-to](#)
- [SLF4J with Log4j2](#)
- [Java Util Logging](#)
- [Spring MVC - PDF](#)
- [JAX-RS - Injecting @Context annota](#)
- [JAX-RS - Containe](#)
- [PrimeFaces - Inv](#)
- [JPA - Flushing Per](#)
- [JPA - Persisting Ei](#)
- [Spring - Calling a SimpleJdbcCall](#)
- [Connecting MySql](#)
- [Spring - Calling a SimpleJdbcCall](#)
- [Spring - Retrievin SimpleJdbcInsert](#)
- [Spring - Inserting](#)
- [PrimeFaces - Tim](#)
- [JSF - Using Custo the same Input co](#)
- [Spring Boot - Ena](#)
- [Spring MVC - Exc](#)
- [Spring - Handling JdbcTemplate and](#)
- [Spring - Rollback](#)
- [Spring Security - Control with @Pre](#)
- [Spring Security - @RolesAllowed](#)
- [Understanding Sp Configuration](#)
- [Java Util Logging](#)

```
@EnableWebMvc
@Configuration
@ComponentScan
public class MyWebConfig extends WebMvcConfigurerAdapter {

    @Override
    public void configureContentNegotiation (ContentNegotiationConfigurer configurer) {
        // configurer.defaultContentType(MediaType.APPLICATION_JSON);
        configurer.defaultContentTypeStrategy(new MyCustomContentNegotiationStrategy());
    }
}
```

[Original Post](#)

```
@EnableWebMvc
@Configuration
public class MyWebConfig extends WebMvcConfigurerAdapter {

    @Override
    public void addViewControllers (ViewControllerRegistry registry) {
        registry.addViewController("/test").setViewName("myView");
    }

    public void others () {
    }
}
```

```
@EnableWebMvc
@Configuration
@ComponentScan
public class MyWebConfig extends WebMvcConfigurerAdapter {

    @Override
    public void configureViewResolvers (ViewResolverRegistry registry) {
        //by default prefix = "/WEB-INF/" and suffix = ".jsp"
        registry.jsp().prefix("/WEB-INF/views/");
    }
}
```

[Original Post](#)

[Redirecting Java log4j with jul-to-](#)[JSF - Creating a C](#)[JSF - Creating a C](#)[JSF - Creating a C](#)[Spring - Logging i](#)[SLF4J with Logba](#)[JAX-RS - Handling headers with Requ](#)[JAX-RS - Injecting annotation](#)[PrimeFaces - Data](#)[Spring MVC - XSL](#)[Spring - Using @J messages](#)[JPA - Removing E](#)[CSS - How to kee body content shoi](#)[Spring - Handling JdbcTemplate](#)[Spring - Logging .](#)[Spring Boot - Reg spring.factories](#)[Spring Boot - Usir SpringApplication](#)[Spring Boot - List @EventListener](#)[Using Apache Tile](#)[Spring - Declarati with @Transactio](#)

See Also

[JAX-RS - Configuration Examples](#)[Spring Boot - @EnableAutoConfiguration Examples](#)[Spring Boot - SpringApplication Examples](#)[Spring Boot - @SpringBootApplication Examples](#)[Spring Framework - @Bean Examples](#)[Spring Framework - @ComponentScan Examples](#)[Spring Framework - @Configuration Examples](#)[Spring Framework - FormatterRegistry Examples](#)[Spring Framework - @Import Examples](#)[Spring Framework - ThreadPoolTaskExecutor Examples](#)[Spring MVC - @Configuration Examples](#)[Spring MVC - HttpMessageConverter Examples](#)[Spring MVC - InterceptorRegistry Examples](#)[Spring MVC - InternalResourceViewResolver Examples](#)[Spring MVC - ResourceHandlerRegistry Examples](#)[Spring MVC - ViewControllerRegistry Examples](#)[Spring MVC - ViewResolver Examples](#)[Spring MVC - ViewResolverRegistry Examples](#)[Spring MVC - WebMvcConfigurer Examples](#)[Spring Web MVC Tutorials](#)[Spring Framework Tutorials](#)

Java 10 Tutorials

[Java 10 Features](#)

Java 9 Tutorials

[Java 9 Module System](#)[Java 9 Misc Features](#)

Recent Tutorials

[JPQL LEFT OUTER JOIN](#)[Java Swing - Editable JTable With Edit Icons](#)[Java Swing - Editable JTable Example](#)[Java 9 JShell](#)[Spring MVC - Application Scoped Bean Example](#)[Spring MVC - Session Scoped Bean created via Class Based Proxy Example](#)[Java Swing - JTable as Editable List](#)

Follow On:

[Twitter](#)[Google+](#)[Faceb](#)

[Like](#) 3.5K people like this. Sign Up to see what your friends like.

If you find our tutorials/p please consider making help us stay on

[JPQL INNER JOIN](#)[Quick Intro to Spring Boot Actuator Endpoints and How to enable them?](#)[Java Swing - A generic builder to create JMenuBar or JPopupMenu invoking on JTable/JTree](#)[Spring MVC - Writing Java Objects to response body](#)[Spring MVC - Request Scoped Bean](#)[Spring MVC - Using HttpSession as Controller Method Argument](#)[Java Swing - Creating JSlider and Listening to ChangeEvent](#)[Understanding Time-Based Release Versioning](#)[Using Literals in JPQL](#)[Eliminating Duplicate Values by using SELECT-DISTINCT statement](#)

Copyright © 2016-2018 [LogicBig](#). 
