# Springboot

> Springboot

2. `Springboot` **classpath:/META-INF/resources/ classpath:/resources/ classpath:/static/ classpath:/public/**
3. `Springboot`

```
#                 /  web.upload-path=D:/temp/study13/

#
spring.mvc.static-path-pattern=/**

#                  static public
#     file:${web.upload-path}  file:                classpath
spring.resources.static-locations=classpath:/META-
INF/resources/,classpath:/resources/,classpath:/static/,classpath:/public/,file:${web.upload-
path}
```

4. `SpringBoot` `Java`

```
import org.springframework.context.annotation.Configuration;
import org.springframework.web.servlet.config.annotation.InterceptorRegistry;
import org.springframework.web.servlet.config.annotation.ResourceHandlerRegistry;
import org.springframework.web.servlet.config.annotation.WebMvcConfigurerAdapter;

@Configuration
// public class WebMvcConfig implements WebMvcConfigurer {
public class WebMvcConfig extends WebMvcConfigurerAdapter {

    @Override
    public void addResourceHandlers(ResourceHandlerRegistry registry) {

        if(!registry.hasMappingForPattern("/static/**")){

registry.addResourceHandler("/static/**").addResourceLocations("classpath:/static/");
        }
```

```
        super.addResourceHandlers(registry);
    }

}
```

5. |Spring Boot| |IDEA| |Spring Boot|  |JS|———  4

```
package com.wzhz.common.core.config;

import java.nio.charset.Charset;
import java.util.ArrayList;
import java.util.List;

import com.google.common.collect.Lists;
import org.springframework.beans.factory.annotation.Value;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
import org.springframework.http.MediaType;
import org.springframework.http.converter.HttpMessageConverter;
import org.springframework.http.converter.StringHttpMessageConverter;
import org.springframework.http.converter.json.MappingJackson2HttpMessageConverter;
import org.springframework.web.cors.CorsConfiguration;
import org.springframework.web.cors.UrlBasedCorsConfigurationSource;
import org.springframework.web.filter.CorsFilter;
import org.springframework.web.servlet.config.annotation.ResourceHandlerRegistry;
import org.springframework.web.servlet.config.annotation.ViewControllerRegistry;
import org.springframework.web.servlet.config.annotation.WebMvcConfigurer;

/**
 * @author wzhz
 * @date 2020 3 24
 */
@Configuration
public class WebMvcConfig implements WebMvcConfigurer {
    @Value("${spring.resource.static-locations}")
    private String staticLocations;

    /**
     *
     *
```

```
     * @return
     */
    @Bean
    public CorsFilter corsFilter() {
        final UrlBasedCorsConfigurationSource urlBasedCorsConfigurationSource = new
UrlBasedCorsConfigurationSource();
        final CorsConfiguration corsConfiguration = new CorsConfiguration();
        /*            */
        corsConfiguration.setAllowCredentials(true);
        /*          */
        corsConfiguration.addAllowedOrigin("*");
        /*              */
        corsConfiguration.addAllowedHeader("*");
        /*        ,GET POST  */
        corsConfiguration.addAllowedMethod("*");
        urlBasedCorsConfigurationSource.registerCorsConfiguration("/**",
corsConfiguration);
        return new CorsFilter(urlBasedCorsConfigurationSource);
    }

    /**
     *        -           Html
     */
    @Override
    public void addResourceHandlers(ResourceHandlerRegistry registry) {

registry.addResourceHandler("/**").addResourceLocations(staticLocations.split(","));
    }

    @Override
    public void addViewControllers(ViewControllerRegistry viewControllerRegistry) {
        //
        viewControllerRegistry.addViewController("/").setViewName("/index.html");
    }

    @Bean
    public StringHttpMessageConverter stringHttpMessageConverter() {
        return new StringHttpMessageConverter(Charset.forName("UTF-8"));
    }
```

```
    /**
     *   ie responsebody  json
     *
     * @return
     */
    @Bean
    public MappingJackson2HttpMessageConverter customJackson2HttpMessageConverter() {
        MappingJackson2HttpMessageConverter jsonConverter = new
MappingJackson2HttpMessageConverter();
        List<MediaType> supportedMediaTypes = Lists.newArrayList();
        supportedMediaTypes.add(new MediaType(MediaType.TEXT_HTML, Charset.forName("UTF-
8")));
        supportedMediaTypes.add(new MediaType(MediaType.APPLICATION_JSON_UTF8,
Charset.forName("UTF-8")));
        jsonConverter.setSupportedMediaTypes(supportedMediaTypes);
        return jsonConverter;
    }

    @Override
    public void configureMessageConverters(List<HttpMessageConverter<?>> converters) {
        converters.add(stringHttpMessageConverter());
        converters.add(customJackson2HttpMessageConverter());
    }

}
```

```
spring:
  profiles:
    active:  dev #   dev| test| prod
  mvc:
    static-path-pattern:  /**
  resource:
    #       ,     WebMvcConfig.java
    static-locations:  classpath:/static/,classpath:/public/
```

---

Revision #1
Created 25 June 2020 05:20:41 by
Updated 25 June 2020 05:34:16 by