



《Spring Boot 基础教程》

第 11 节

web 应用开发-CORS 支持

一、Web 开发经常会遇到跨域问题，解决方案有：jsonp, iframe, CORS 等等

CORS 与 JSONP 相比

- 1、JSONP 只能实现 GET 请求，而 CORS 支持所有类型的 HTTP 请求。
- 2、使用 CORS，开发者可以使用普通的 XMLHttpRequest 发起请求和获得数据，比起 JSONP 有更好的错误处理。
- 3、JSONP 主要被老的浏览器支持，它们往往不支持 CORS，而绝大多数现代浏览器都已经支持了 CORS

浏览器支持情况

Chrome 3+

Firefox 3.5+

Opera 12+

Safari 4+

Internet Explorer 8+

二、在 spring MVC 中可以配置全局的规则，也可以使用@CrossOrigin 注解进行细粒度的配置。

全局配置：

@Configuration

```
public class CustomCorsConfiguration {
```

```
    @Bean
```

```
    public WebMvcConfigurer corsConfigurer() {
```

```
        return new WebMvcConfigurerAdapter() {
```

```
            @Override
```

```
            public void addCorsMappings(CorsRegistry registry) {
```

```
                registry.addMapping("/api/**").allowedOrigins("http://localhost:8080");
```

```
            }
```

```
        };
```

```
    }
```

```
}
```

或者是

```
/**
```

```
 * 全局设置
```

```
 *
```

```
 * @author wujing
```

```
 */
```

```
@Configuration
```

```
public class CustomCorsConfiguration2 extends WebMvcConfigurerAdapter {
```



```
@Override
public void addCorsMappings(CorsRegistry registry) {
    registry.addMapping("/api/**").allowedOrigins("http://localhost:8080");
}
}
```

定义方法：

```
/**
 * @author wujing
 */
@RestController
@RequestMapping("/api")
public class ApiController {

    @RequestMapping(value = "/get")
    public HashMap<String, Object> get(@RequestParam String name) {
        HashMap<String, Object> map = new HashMap<String, Object>();
        map.put("title", "hello world");
        map.put("name", name);
        return map;
    }
}
```

测试 js:

```
$.ajax({
    url: "http://localhost:8081/api/get",
    type: "POST",
    data: {
        name: "测试"
    },
    success: function(data, status, xhr) {
        console.log(data);
        alert(data.name);
    }
});
```

细粒度配置

```
/**
 * @author wujing
 */
@RestController
@RequestMapping(value = "/api", method = RequestMethod.POST)
public class ApiController {
```



```
@CrossOrigin(origins = "http://localhost:8080")
@RequestMapping(value = "/get")
public HashMap<String, Object> get(@RequestParam String name) {
    HashMap<String, Object> map = new HashMap<String, Object>();
    map.put("title", "hello world");
    map.put("name", name);
    return map;
}
}
```

更多课程信息，请关注 龙果学院 官方网站 <http://www.roncoo.com/>
或关注 龙果 微信公众号 RonCoo_com

