

Swagger 整合 Oauth2

作者: [jianzh5](#)

原文链接: <https://ld246.com/article/1583133725353>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

一旦项目中加入了Oauth2.0，那么每次请求接口都需要在header上带上Authorization参数才可以正常访问，如下所示：

The screenshot shows the Postman interface for a GET request to `http://localhost:8010/account/getByCode/javadaily`. In the 'Headers' tab, there is a single entry for 'Authorization' with the value 'Bearer 36034ff7-7eea-4935-a3b7-5787d7a1'. Below the request, the response status is '200 OK' with a response body containing JSON data:

```
1 [ { 2 "status": 100, 3 "message": "请求成功", 4 "data": { 5 "id": 1, 6 "accountCode": "javadaily", 7 "accountName": "JAVA日知录", 8 "amount": 9847 9 }, 10 "success": true, 11 "timestamp": 1583127522443 12 } ]
```

项目用了Swagger在线接口文档组件，那么如何结合Oauth2.0，让调用接口的时候自动带上认证参数呢？

关键代码

```
@Configuration
@EnableSwagger2
public class SwaggerConfig {
    private static final String VERSION = "1.0.0";
    /**
     * 创建API
     */
    @Bean
    public Docket createRestApi(){
        return new Docket(DocumentationType.SWAGGER_2)
            .apiInfo(apiInfo())
            .select()
            //指定接口包所在路径
            .apis(RequestHandlerSelectors.withClassAnnotation(Api.class))
            .paths(PathSelectors.any())
            .build()
            //整合oauth2
            .securitySchemes(Collections.singletonList(apiKey()))
            .securityContexts(Collections.singletonList(securityContext()));
    }

    /**
     * 添加摘要信息
     */
    private ApiInfo apiInfo() {
        return new ApiInfoBuilder()
```

```

        .contact(new Contact("JAVA日知录","http://javadaily.cn","jianzh5@163.com"))
        .title("account-server接口文档")
        .description("account-server接口文档")
        .termsOfServiceUrl("http://javadaily.cn")
        .version(VERSION)
        .build();
    }

private ApiKey apiKey() {
    return new ApiKey("Bearer", "Authorization", "header");
}

/**
 * swagger2 认证的安全上下文
 */
private SecurityContext securityContext() {
    return SecurityContext.builder()
        .securityReferences(defaultAuth())
        .forPaths(PathSelectors.any())
        .build();
}

private List<SecurityReference> defaultAuth() {
    AuthorizationScope authorizationScope = new AuthorizationScope("web", "access_token");
    AuthorizationScope[] authorizationScopes = new AuthorizationScope[1];
    authorizationScopes[0] = authorizationScope;
    return Collections.singletonList(new SecurityReference("Bearer",authorizationScopes));
}
}

```

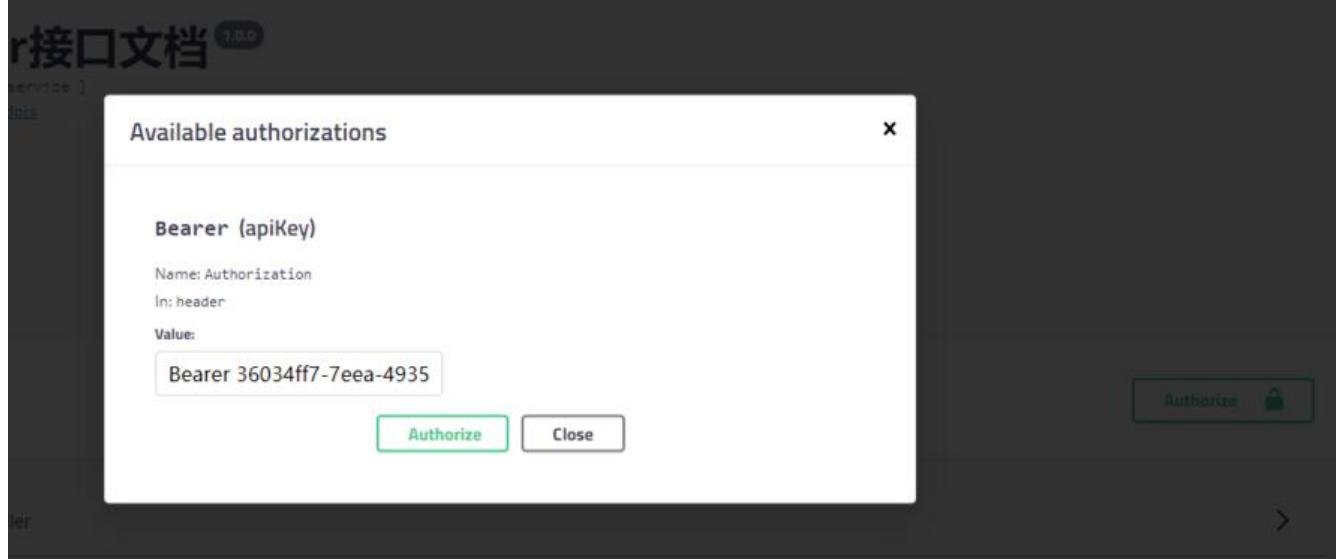
使用步骤

- 调用认证中心获取access_token

<http://localhost:8090/auth-service/oauth/token>

```
{
  "access_token": "36034ff7-7eea-4935-a3b7-5787d7a65827",
  "token_type": "bearer",
  "refresh_token": "4baea735-3c0d-4dfd-b826-91c6772a0962",
  "expires_in": 36931,
  "scope": "web"
}
```

- 访问Swagger接口页面，点击Authorize接口进行认证，在弹出框中输入 **Bearer 36034ff7-7eea-4935-a3b7-5787d7a65827** 并点击认证按钮。



- 在Swagger中正常请求接口

Responses Response content type */*

Curl

```
curl -X GET "http://localhost:8090/account-service/account/getByCode/jianzh5" -H "accept: */*" -H "Authorization: Bearer 36034ff7-7eea-4935-a3b7-5787d7a65827"
```

Request URL

```
http://localhost:8090/account-service/account/getByCode/jianzh5
```

Server response

Code	Details
200	Response body

```
{  
    "status": 100,  
    "message": "请求成功",  
    "data": {  
        "id": 17,  
        "accountCode": "jianzh5",  
        "accountName": "jianzh5",  
        "amount": 94  
    },  
    "success": true,  
    "timestamp": 1583133579396  
}
```

Download

经过以上几步可以看到接口请求会默认带上认证参数，小伙伴们又可以愉快的玩耍了！