

11

- 1
- 2 POM.xml
- 3 @EnableSso
- 4
- 5
- 6
- 7
- 8

1

- JDK8+
- Springboot 2.3.1
- SpringCloudHoxton.SR8
- SpringCloud Alibaba2.2.3.RELEASE
- Dubbo3.0.4
- Nacos2.0.3

<http://192.168.15.163:7071/> admin

nacosnacos://192.168.15.163:8848

->->@EnableSso->nacos->@PermissionValue->AuthorityRegisterComponent->

2 POM.xml

pom.xml

```
<parent>
  <groupId>com.fourfaith.framework</groupId>
  <artifactId>fourfaith-framework-parent</artifactId>
  <version>2.0-SNAPSHOT</version>
  <relativePath />
</parent>

<distributionManagement>
  <repository>
    <id>maven-snapshots</id>
    <name>Nexus Repository</name>
    <url>http://192.168.9.221:8081/repository/maven-snapshots/</url>
  </repository>
</distributionManagement>
<repositories>
  <repository>
    <id>maven-public</id>
    <name>Nexus Repository</name>
    <url>http://192.168.9.221:8081/repository/maven-public/</url>
  </repository>
</repositories>

<dependencies>
  <dependency>
    <groupId>com.fourfaith.framework</groupId>
    <artifactId>fourfaith-dubbo</artifactId>
    <version>2.0-SNAPSHOT</version>
  </dependency>
</dependencies>
```

3 @EnableSso

MyApp.class

```
@SpringBootApplication
@EnableSso
public class UserCenterAccessDemoApplication {
    public static void main(String[] args) {
        SpringApplication.run(UserCenterAccessDemoApplication.class, args);
    }
}
```

```
ApplicationId id
tenantId id
```

4

`${user.home} dubbo-resolve.properties`

dubbo-resolve.properties

```
com.fourfaith.dubbo.service.UserCenterDubboService=dubbo://127.0.0.1:
21880
```

dubbo

`jvm -Ddubbo.resolve.file=dubbo-resolve.properties`

5

`application.yml properties`

application.yml

```
dubbo:
  application:
    name: ${Spring.application.name} #
  provider:
    protocol: injvm #
  consumer:
    check: false # ,,
    protocol:
      name: dubbo
      port: -1 # , -1
    registry:
      address: nacos://192.168.15.163:8848 # nacos
      version: 1.0.0
    scan:
      base-packages: com.fourfaith.iot
```

6

```
@PermissionValue(key = "...", desc = "")
```

```
    " " " "
```

```
,
```

ReservoirController.class

```
@RestController
@RequestMapping("/reservoir/info")
@RequiredArgsConstructor
public class ReservoirController {

    private final ReservoirService service;

    /**
     * @param entity
     */
    @PermissionValue(key = "reservoir-monitoring-platform.reservoir.info", desc = "")
    @PostMapping("/page")
    public ResultDto page() {
        Assert.notNull(entity.getUnitId(), "id");
        Assert.notNull(entity.getRscd(), "");
        Assert.notNull(entity.getRsnm(), "");
        Assert.notNull(entity.getEslg(), "");
        Assert.notNull(entity.getNrIt(), "");
        return ResultDto.returnSuccessData(server.page);
    }

    /**
     * @param entity
     */
    @PermissionValue(key = "reservoir-monitoring-platform.reservoir.info.add", desc = "")
    @PostMapping("/add")
    public ResultDto add(@RequestBody Reservoir entity) {
        Assert.notNull(entity.getUnitId(), "id");
        Assert.notNull(entity.getRscd(), "");
        Assert.notNull(entity.getRsnm(), "");
        Assert.notNull(entity.getEslg(), "");
        Assert.notNull(entity.getNrIt(), "");
        service.add(entity);
        return ResultDto.returnSuccess();
    }
}
```

7

```
<dependency>
  <groupId>com.fourfaith.framework</groupId>
  <artifactId>fourfaith-dubbo</artifactId>
</dependency>
```

AuthorityRegisterController.class

```
package com.fourfaith.iot.reservoir.controller.reservoir;

import com.fourfaith.common.pojo.ResultDto;
import com.fourfaith.dubbo.annotation.UnCheckToken;
import com.fourfaith.dubbo.configuration.AuthorityRegisterComponent;
import lombok.RequiredArgsConstructor;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

@RestController
@RequestMapping("/api/authority-register")
@RequiredArgsConstructor
public class AuthorityRegisterController {

    @Autowired
    private AuthorityRegisterComponent authorityRegisterComponent;

    @GetMapping("/run")
    @UnCheckToken
    public ResultDto run() {
        authorityRegisterComponent.run();
        return ResultDto.returnSuccess();
    }
}
```

8

请求参数设置

Body

Query

Headers

添加Header

Authorization	jwt token	Bearer eyJ0eXAiOiJVZS1QcCJhbGciOiJIUzI1Ni9eyjYwSSQ29kZS16mN4d2x3IiwiaXVkiOjZGV2aWNLWNsb3VklWNsaWVudCIsmImzcyIjE6MTdlbmJlZS1jbG91ZC1hZG1pbilislmV4C16MTY1MTUSOTcxMwidXNlckShbWUiOiJhZG1pbilislnR5cGU/OiBQONFU1MilCjpwYXQ/OjE2NDg0NDYzMtYslnVzZXJJZCl6JWOTkiLCJqdGkiOiNyN2UzMjNkYmZvOWQwLTQxYTmtYTM1ZC0wMWYyZGNjYmZNGYifQ.xMTelCpqioXUIpQOj94s6bFgcqXTGsuo0aoap_cD8	备注
applicationid	应用id	123	备注
tenantId	租户id	456	备注

UserInfoController.java

```
@RestController
@RequestMapping("/api/user-info")
@RequiredArgsConstructor
public class UserInfoController {

    @GetMapping("/get")
    @PermissionValue(key = "...", desc = "...")
    public ResultDto get() {
        //id
        Long applicationId = SecurityUtils.getApplicationId();
        //id
        Long tenantId = SecurityUtils.getTenantId();
        //id
        Long userId = SecurityUtils.getUserId();
        .....
        return ResultDto.returnSuccess();
    }
}
```