

总线发布

maven依赖

```
<dependency>
  <groupId>org.springframework.cloud</groupId>
  <artifactId>spring-cloud-starter-bus-amqp</artifactId>
</dependency>
```

配置

```
spring:
  rabbitmq:
    host: localhost
    port: 5672
    username: guest
    password: guest
```

示例代码

启动类

```
@RemoteApplicationEventScan(basePackages =
"com.github.wxiaoqi.security.auth.common.event")
public class AuthBootstrap {
    public static void main(String[] args) {
        SpringApplication.run(AuthBootstrap.class, args);
    }
}
```

基础事件

```

@Data
public class AuthRemoteEvent extends RemoteApplicationEvent {
    private List<String> allowedClient;

    //jackson序列化反序列化必须有无参构造函数
    public AuthRemoteEvent() {
    }

    public AuthRemoteEvent(Object source, String originService,
String destinationService, List<String> allowedClient) {
        // source is the object that is publishing the event
        // originService is the unique context ID of the publisher
        super(source, originService, destinationService);
        this.allowedClient = allowedClient;
    }
}

```

推送

```

@Service
public class DBClientService {
    private ApplicationContext context;

    @Autowired
    public DBClientService(ApplicationContext context) {
        this.context = context;
    }

    public void do(){
        // 主动推送
        final List<String> clients = null;
        final String myUniqueId = context.getId();
        final AuthRemoteEvent event =
            new AuthRemoteEvent(this, myUniqueId, name,
clients);
        context.publishEvent(event);
    }
}

```

消费事件

maven依赖

```
<dependency>
  <groupId>org.springframework.cloud</groupId>
  <artifactId>spring-cloud-starter-bus-amqp</artifactId>
</dependency>
```

配置

```
spring:
  rabbitmq:
    host: localhost
    port: 5672
    username: guest
    password: guest
```

示例代码

启动类

```
@RemoteApplicationEventScan(basePackages =
"com.github.wxiaoqi.security.auth.common.event")
public class AuthBootstrap {
    public static void main(String[] args) {
        SpringApplication.run(AuthBootstrap.class, args);
    }
}
```

消费类

```
@Configuration
@Slf4j
public class ServiceAuthUtil implements
ApplicationListener<AuthRemoteEvent> {
```

```
@Override
public void onApplicationEvent(AuthRemoteEvent authRemoteEvent)
{
    this.allowedClient = authRemoteEvent.getAllowedClient();
}
}
```