在Redis中存储token

之前的代码我们将token直接存在内存中，这在生产环境中是不合理的，下面我们将其改造成存储在Redis中

添加依赖及配置

pom.xml

```
<!--redis 依赖-->
<dependency>
   <groupId>org.springframework.boot</groupId>
   <artifactId>spring-boot-starter-data-redis</artifactId>
</dependency>
<!-- commons-pool2 对象池依赖 -->
<dependency>
   <groupId>org.apache.commons</groupId>
   <artifactId>commons-pool2</artifactId>
</dependency>
```

application.properties

```
# Redis配置
spring.redis.host=192.168.10.100
```

编写Redis配置类

RedisConfig.java

```
package com.xxxx.springsecurityoauth2demo.config;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
import org.springframework.data.redis.connection.RedisConnectionFactory;
import org.springframework.security.oauth2.provider.token.TokenStore;
import org.springframework.security.oauth2.provider.token.store.redis.RedisTokenStore;

/**
 * 使用redis存储token的配置
 * @author zhoubin
 * @since 1.0.0
 */
@Configuration
public class RedisConfig {
   @Autowired
   private RedisConnectionFactory redisConnectionFactory;

   @Bean
   public TokenStore redisTokenStore(){
```

```
        return new RedisTokenStore(redisConnectionFactory);
    }

}
```

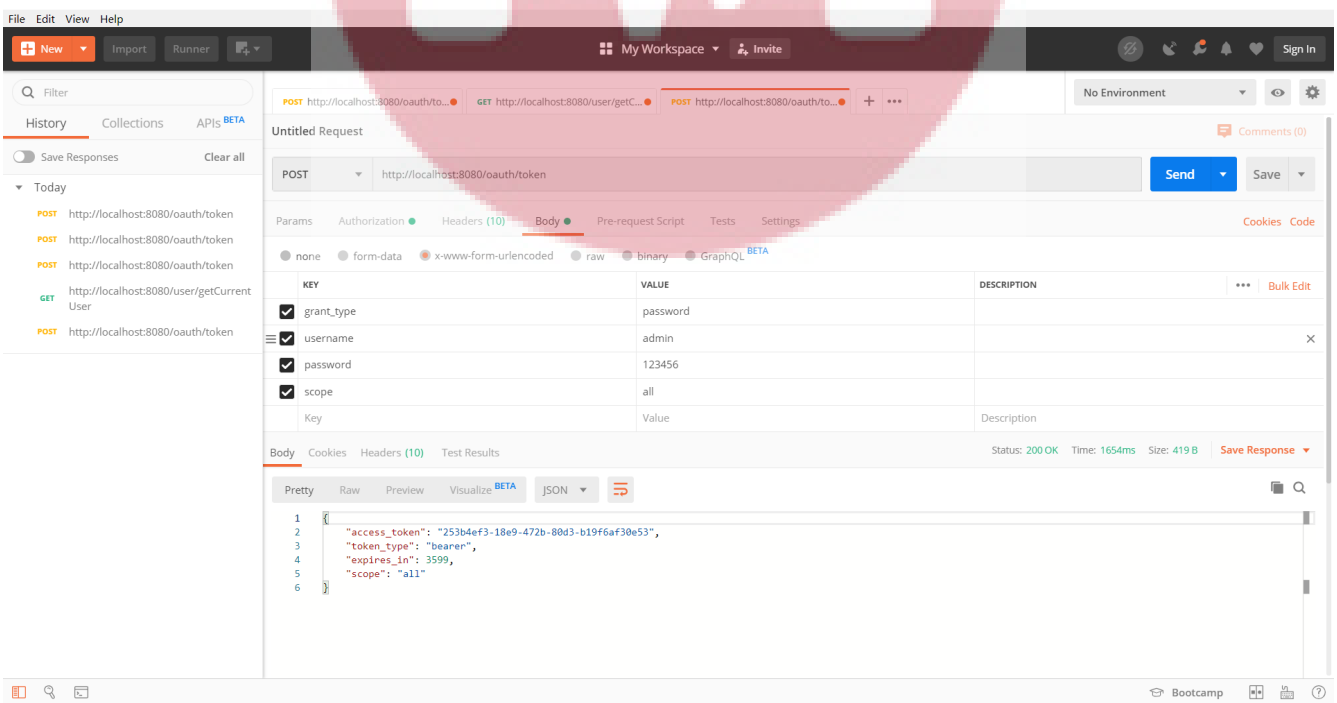
在认证服务器配置中指定令牌的存储策略为Redis

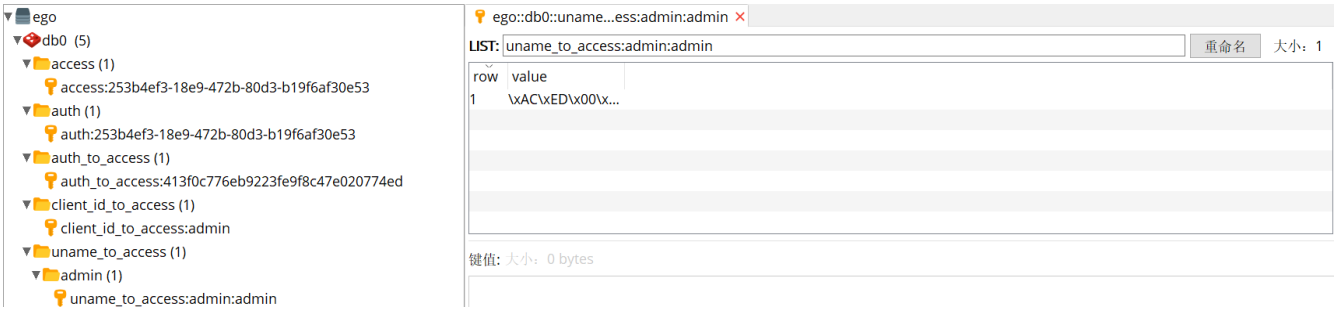
```
package com.xxxx.springsecurityoauth2demo.config;

import com.xxxx.springsecurityoauth2demo.service.UserService;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.beans.factory.annotation.Qualifier;
import org.springframework.context.annotation.Configuration;
import org.springframework.security.authentication.AuthenticationManager;
import org.springframework.security.crypto.password.PasswordEncoder;
import
org.springframework.security.oauth2.config.annotation.configurers.ClientDetailsServiceConfi
gurer;
import
org.springframework.security.oauth2.config.annotation.web.configuration.AuthorizationServer
ConfigurerAdapter;
import
org.springframework.security.oauth2.config.annotation.web.configuration.EnableAuthorization
Server;
import
org.springframework.security.oauth2.config.annotation.web.configurers.AuthorizationServerEn
dpointsConfigurer;
import org.springframework.security.oauth2.provider.token.TokenStore;

/**
 * 授权服务器配置
 * @author zhoubin
 * @since 1.0.0
 */
@Configuration
@EnableAuthorizationServer
public class AuthorizationServerConfig extends AuthorizationServerConfigurerAdapter {

    @Autowired
    private PasswordEncoder passwordEncoder;

    @Autowired
    private AuthenticationManager authenticationManager;

    @Autowired
    private UserService userService;

    @Autowired
    @Qualifier("redisTokenStore")
    private TokenStore tokenStore;

    /**
     * 使用密码模式需要配置
```

```
     */
    @Override
    public void configure(AuthorizationServerEndpointsConfigurer endpoints) {
        endpoints.authenticationManager(authenticationManager)
                .userDetailsService(userService)
                .tokenStore(tokenStore);
    }

    @Override
    public void configure(ClientDetailsServiceConfigurer clients) throws Exception {
        clients.inMemory()
                //配置client_id
                .withClient("admin")
                //配置client-secret
                .secret(passwordEncoder.encode("112233"))
                //配置访问token的有效期
                .accessTokenValiditySeconds(3600)
                //配置刷新token的有效期
                .refreshTokenValiditySeconds(864000)
                //配置redirect_uri，用于授权成功后跳转
                .redirectUris("http://www.baidu.com")
                //配置申请的权限范围
                .scopes("all")
                //配置grant_type，表示授权类型
                .authorizedGrantTypes("password");
    }
}
```

测试：

使用密码模式请求token




ego
db0 (5)
access (1)
access:253b4ef3-18e9-472b-80d3-b19f6af30e53
auth (1)
auth:253b4ef3-18e9-472b-80d3-b19f6af30e53
auth_to_access (1)
auth_to_access:413f0c776eb9223fe9f8c47e020774ed
client_id_to_access (1)
client_id_to_access:admin
uname_to_access (1)
admin (1)
uname_to_access:admin:admin
ego::db0::uname...ess:admin:admin
LIST: uname_to_access:admin:admin
重命名
大小：1
row
value
1
\xAC\xED\x00\x...
键值: 大小：0 bytes