



token过期校验

有很多时候，我们并不希望签发的token是永久生效的（上节的token是永久的），所以我们可以为token添加一个过期时间。原因：从服务器发出的token，服务器自己并不做记录，就存在一个弊端就是，服务端无法主动控制某token的立刻失效。

测试用例：

```
@Test
public void testCreatTokenHasExp() {
    //当前系统时间的长整型
    long now = System.currentTimeMillis();
    //过期时间，这里是1分钟后的时间长整型
    long exp = now + 60 * 1000;
    //创建一个JwtBuilder对象
    JwtBuilder jwtBuilder = Jwts.builder()
        //声明的标识{"jti":"888"}
        .setId("888")
        //主体，用户{"sub":"Rose"}
        .setSubject("Rose")
        //创建日期{"ita":"xxxxxx"}
        .setIssuedAt(new Date())
        //签名手段，参数1：算法，参数2：盐
        .signWith(SignatureAlgorithm.HS256, "xxxx")
        //设置过期时间
        .setExpiration(new Date(exp));
    //获取jwt的token
    String token = jwtBuilder.compact();
    System.out.println(token);
}

@Test
public void testParseTokenHasExp() {
    //token
    String token = "eyJhbGciOiJIUzI1NiJ9.eyJqdGkiOiI4ODgiLCJzdzIiOiJSb3NlIiwiaWF0IjoxNTc4ODEMDYyLCJleHAiOjE1Nzg4MTUxMjIsInJvbGVzIjoieWRtaw4iLCJzb2dvIjoic2hzeHQvanBnIn0.hKog0RsZ9_6II_R8kUCp0HLAouUAYXAJVbz3xtLTUh4";
    //解析token获取负载中的声明对象
    Claims claims = Jwts.parser()
        .setSigningKey("xxxx")
        .parseClaimsJws(token)
        .getBody();
    //打印声明的属性
    System.out.println("id:" + claims.getId());
    System.out.println("subject:" + claims.getSubject());
    System.out.println("issuedAt:" + claims.getIssuedAt());
    DateFormat sf = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");
    System.out.println("签发时间:" + sf.format(claims.getIssuedAt()));
    System.out.println("过期时间:" + sf.format(claims.getExpiration()));
}
```

```
System.out.println("当前时间:"+sf.format(new Date()));  
}
```

测试：当未过期时可以正常读取，当过期时会引发`io.jsonwebtoken.ExpiredJwtException`异常。

```
io.jsonwebtoken.ExpiredJwtException: JWT expired at 2020-01-12T15:38:01Z. Current time: 2020-01-12T15:39:54Z, a difference of 113145 milliseconds. Allowed clock skew: 0 milli  
  
at io.jsonwebtoken.impl.DefaultJwtParser.parse(DefaultJwtParser.java:385)  
at io.jsonwebtoken.impl.DefaultJwtParser.parse(DefaultJwtParser.java:481)  
at io.jsonwebtoken.impl.DefaultJwtParser.parseClaimsJws(DefaultJwtParser.java:541)  
at com.ego.jwtdemo.JwtDemoApplicationTests.testParseTokenHasExp(JwtDemoApplicationTests.java:56) <31 internal calls>  
at java.util.ArrayList.forEach(ArrayList.java:1257) <9 internal calls>  
at java.util.ArrayList.forEach(ArrayList.java:1257) <21 internal calls>
```

