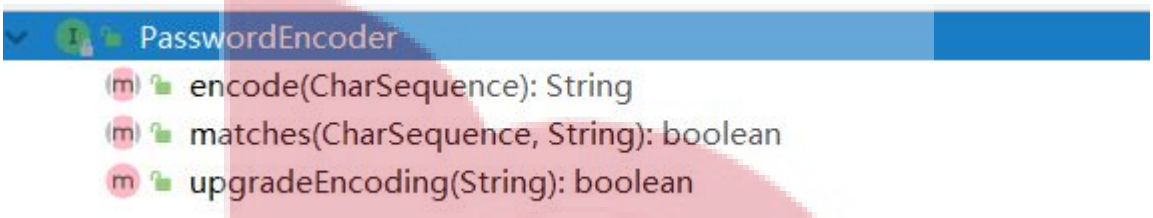
PasswordEncoder 密码解析器详解

Spring Security 要求容器中必须有`PasswordEncoder`实例。所以当自定义登录逻辑时要求必须给容器注入`PaswordEncoder`的bean对象。

接口介绍

- `encode()`：把参数按照特定的解析规则进行解析。
- `matches()`：验证从存储中获取的编码密码与编码后提交的原始密码是否匹配。如果密码匹配，则返回 true；如果不匹配，则返回 false。第一个参数表示需要被解析的密码。第二个参数表示存储的密码。
- `upgradeEncoding()`：如果解析的密码能够再次进行解析且达到更安全的结果则返回 true，否则返回 false。默认返回 false。



内置解析器介绍

在 Spring Security 中内置了很多解析器。

BCryptPasswordEncoder 简介

BCryptPasswordEncoder 是 Spring Security 官方推荐的密码解析器，平时多使用这个解析器。

BCryptPasswordEncoder 是对 `bcrypt`强散列方法的具体实现。是基于Hash算法实现的单向加密。可以通过strength控制加密强度，默认 10.

代码演示

新建测试方法BCryptPasswordEncoder 用法。

```
package com.xxxx.springsecuritydemo;

import org.junit.jupiter.api.Test;
import org.springframework.boot.test.context.SpringBootTest;
```

```java
import org.springframework.security.crypto.bcrypt.BCryptPasswordEncoder;
import org.springframework.security.crypto.password.PasswordEncoder;

/**
 * @author zhoubin
 * @since 1.0.0
 */
@SpringBootTest
public class MyTest {

   @Test
   public void test(){
      //创建解析器
      PasswordEncoder pw = new BCryptPasswordEncoder();
      //对密码加密
      String encode = pw.encode("123");
      System.out.println(encode);

      //判断原字符和加密后内容是否匹配
      boolean matches = pw.matches("1234", encode);
      System.out.println("==================="+matches);
   }

}
```