BCrypt

```
BCryptPasswordEncoder bCryptPasswordEncoder = new BCryptPasswordEncoder();
//对111111进行第一次加密
String encode = bCryptPasswordEncoder.encode("111111");
System.out.println("encode = " + encode);

/*
运行3次结果
* 1:encode = $2a$10$E4REtiAWzkqrrUt6AymwJOr3KzJpgZ1kr2g48bLg8SUKp3WfkzRrW
* 2:encode = $2a$10$7oztYp2cwYXFGWlnSBfoX.BsNxGBj1dyT5Vx2M0XC2n3GT2.R3dte
* 3:encode = $2a$10$zaRVI77HmCCGs6J6UppdKuZIRM5zaEwVfmA6y.QOnHCMXlR7XcCwu
*/

// 进行密码匹配
boolean matches1 = bCryptPasswordEncoder.matches("111111", "$2a$10$zaRVI77HmCCGs6J6UppdKuZIRM5zaEwVfmA6y.QOnHCMXlR7XcCwu");
System.out.println("matches = " + matches1);//true
boolean matches2 = bCryptPasswordEncoder.matches("111111", "$2a$10$7oztYp2cwYXFGWlnSBfoX.BsNxGBj1dyT5Vx2M0XC2n3GT2.R3dte");
System.out.println("matches = " + matches2);//true
```

解释

加密长度固定为60位
$是分割符，无意义;
2a是bcrypt加密版本号;、
10是cost的值;
而后的前22位是salt值;
再然后的字符串就是密码的密文了。

加密用户密码

先在配置类中创建一个密码加密器放到IOC容器中

```
//创建一个密码加密器放到IOC容器中
@Bean
public PasswordEncoder getPasswordEncoder(){
return new BCryptPasswordEncoder();
}
```

在使用的类中注入

```
@Autowired
private PasswordEncoder passwordEncoder;
```

```
//对admin中的密码进行加密
String password = admin.getPassword();
String encode = passwordEncoder.encode(password);
admin.setPassword(encode);
```