token的生成和验证合法性

```
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;
import java.util.UUID;

/**
 * Created by mjchow
 * Date       2016/2/19
 * Time       11:00
 */
public class TokenUtil {

    private static String key = "mjchow";

    private static String getToken() {
        String token = UUID.randomUUID().toString();
        return token.replace("-", "");
    }

    public static String getEncryptToken() {
        String token = getToken();
        return token+getMd5(token+key);
    }

    public static boolean validateToken(String encryptToken) {
        String token = encryptToken.substring(0, 32);
        String encryptKey = encryptToken.substring(32);
        if(encryptKey.equals(getMd5(token+key))) {
            return true;
        }
        return false;
    }

    //静态方法，便于作为工具类
    private static String getMd5(String plainText) {
        try {
            MessageDigest md = MessageDigest.getInstance("MD5");
            md.update(plainText.getBytes());
            byte b[] = md.digest();
            int i;
            StringBuffer buf = new StringBuffer("");
            for (int offset = 0; offset < b.length; offset++) {
                i = b[offset];
                if (i < 0)
                    i += 256;
                if (i < 16)
                    buf.append("0");
                buf.append(Integer.toHexString(i));
            }
            //32位加密
            return buf.toString();
            // 16位的加密
            //return buf.toString().substring(8, 24);
        } catch (NoSuchAlgorithmException e) {
            e.printStackTrace();
            return null;
        }

    }


    public static void main(String[] args) {
        String encryptToken = getEncryptToken();
        System.out.println("+---------->"+encryptToken);
```

```
        System.out.println("validateSuccess->" + validateToken(encryptToken));
    }
}
```

生成的字符有64位长，其中32位为UUID生成的唯一随机串，后面32为加密前面32位的加密串，自带验证其合法性。