SecurityUtil

Description

- org.joget.commons.util.SecurityUtil
- Under wflow-commons module
- Utility methods used by security feature

Code Sample

```
import org.joget.commons.util.SecurityUtil;


String value = "this is a test string";
String encryptedValue = SecurityUtil.encrypt(value);
```

Fields

ENVELOPE

```
public final static String ENVELOPE = "%%%%";
```

A string used to prepend and append to a encrypted data for differential purpose.

Methods

computeHash

```
public static java.lang.String computeHash(java.lang.String rawContent, java.lang.String randomSalt)
```

Computes the hash of a raw content if data encryption implementation is exist

decrypt

```
public static java.lang.String decrypt(java.lang.String protectedContent)
```

Decrypt protected content if data encryption implementation is exist

encrypt

```
public static java.lang.String encrypt(java.lang.String rawContent)
```

Encrypt raw content if data encryption implementation is exist

generateNonce

```
public static java.lang.String generateNonce(java.lang.String[] attributes, int lifepanHour)
```

Generate a nonce value based on attributes if Nonce Generator implementation is exist

generateRandomSalt

```
public static java.lang.String generateRandomSalt()
```

Generate a random salt value if data encryption implementation is exist

getApplicationContext

```
public static org.springframework.context.ApplicationContext getApplicationContext()
```

Utility method to retrieve the ApplicationContext of the system

getCsrfTokenName

```
public static java.lang.String getCsrfTokenName()
```

Returns the name of the CRSF token

getCsrfTokenValue

```
public static java.lang.String getCsrfTokenValue(javax.servlet.http.HttpServletRequest request)
```

Returns the value of the CRSF token in the request

getDataEncryption

```
public static org.joget.commons.util.DataEncryption getDataEncryption()
```

Gets the data encryption implementation

getDomainName

```
public static java.lang.String getDomainName(java.lang.String url)
```

Gets the domain name from a given URL

getNonceGenerator

```
public static org.joget.commons.util.NonceGenerator getNonceGenerator()
```

Gets the nonce generator implementation

hasSecurityEnvelope

```
public static boolean hasSecurityEnvelope(java.lang.String content)
```

Check the content is a wrapped in a security envelop if data encryption implementation is exist

isAllowedDomain

```
public static boolean isAllowedDomain(java.lang.String domain, java.util.List<java.lang.String> whitelist)
```

Verify the domain name against a whitelist

setDataEncryption

```
public void setDataEncryption(org.joget.commons.util.DataEncryption deImpl)
```

Sets a data encryption implementation

setNonceGenerator

```
public void setNonceGenerator(org.joget.commons.util.NonceGenerator ngImpl)
```

Sets a nonce generator implementation

verifyHash

```
public static java.lang.Boolean verifyHash(java.lang.String hash, java.lang.String randomSalt, java.lang.
String rawContent)
```

Verify the hash is belong to the raw content if data encryption implementation is exist

verifyNonce

```
public static boolean verifyNonce(java.lang.String nonce, java.lang.String[] attributes)
```

Verify the nonce is a valid nonce against the attributes if Nonce Generator implementation is exist