AES/CBC/PKCS5PADDING - AES 128 bit Encryption in ECB Mode (Electronic Code Book Mode) No Padding

1. You create a Cipher instance by calling its getInstance() method with a parameter telling what type of encryption algorithm you want to use. Here in securityClient class of creating a Java Cipher instance: An encryption mode specifies details about how the algorithm should encrypt data.

Cipher.*getInstance*("AES/CBC/PKCS5Padding");

1. Before you can use a Cipher instance you must initialize it. Initializing a Cipher is done by calling its init() method. he init() method takes two parameters:

Encryption / decryption cipher operation mode.

Encryption / decryption key.

1. The reason a call to doFinal() is needed for the last block of data is, that some encryption algorithms need to pad the the data to fit a certain cipher block size (e.g. an 8 byte boundary). But - we do not want to pad the intermediate blocks of data encrypted.

cipher.init(Cipher.***ENCRYPT\_MODE***, *secret*, **new** IvParameterSpec(**new** **byte**[cipher.getBlockSize()]));

ciphertext = cipher.doFinal(str.getBytes(StandardCharsets.***UTF\_8***));

1. DatatypeConverter has print and parse methods. print method encodes the data into lexical representation of xsd. And parse method can decode the lexical representation to string. For example

DatatypeConverter.*printHexBinary*(ciphertext)

Same all above steps followed to decryption but init method use ***DECRYPT\_MODE***