### encrypt()

Encrypts the binary value using the specified algorithm and key.

#### Syntax

encrypt(CIPHER, KEY, DATA[, IV])

Required Parameter

**CIPHER**

Expression that returns a string in the form of algorithm/mode/padding. If you omit the mode and padding and provide only the algorithm, the default cryptographic algorithm is applied:

* If you provide only AES, AES/CBC/NoPadding are used.
* If you provide only RSA, RSA/ECB/PKCS1Padding are used.

For the algorithms, modes, and paddings available, refer to the Java Security Standard Algorithm Names document: <https://docs.oracle.com/en/java/javase/11/docs/specs/security/standard-names.html>

* You can see the algorithms in the **Cipher Algorithm Names** section.
* You can see the modes in the **Cipher Algorithm Modes** section.
* You can see the paddings in the **Cipher Algorithm Paddings** section.

Logpresso supports various cryptographic algorithms provided by Java for compatibility. However, it is recommended not to use insecure cryptographic algorithms or modes such as DES or ECB.For information on cipher classes used in Java, refer to the documentation at the following address: https://docs.oracle.com/en/java/javase/11/docs/api/java.base/javax/crypto/Cipher.html

The following are the formats that all Java implementations must support. DES, DESede algorithms, and ECB modes are insecure and should only be used when there are compatibility issues with external systems. The numbers in parentheses indicate the key length in bits.- AES/CBC/NoPadding (128)- AES/CBC/PKCS5Padding (128)- AES/ECB/NoPadding (128)- AES/ECB/PKCS5Padding (128)- AES/GCM/NoPadding (128)- DES/CBC/NoPadding (56)- DES/CBC/PKCS5Padding (56)- DES/ECB/NoPadding (56)- DES/ECB/PKCS5Padding (56)- DESede/CBC/NoPadding (168)- DESede/CBC/PKCS5Padding (168)- DESede/ECB/NoPadding (168)- DESede/ECB/PKCS5Padding (168)- RSA/ECB/PKCS1Padding (1024, 2048)- RSA/ECB/OAEPWithSHA-1AndMGF1Padding (1024, 2048)- RSA/ECB/OAEPWithSHA-256AndMGF1Padding (1024, 2048)

**KEY**

Binary key of a size that matches the specified cryptographic algorithm. The key length according to the algorithm is as follows:

* AES: 16 characters (128 bits), 24 characters (192 bits), and 32 characters (256 bits).
* RSA: 128 characters (1,024 bits) and 256 characters (2,048 bits).

The encryption key length is the value obtained by dividing the key length required by the encryption algorithm into 8 bytes.

**DATA**

Binary data to encrypt.

Optional Parameter

**IV**

Binary value as an Initial Vector(IV) required in operating mode such as CBC.

#### Usage

json "{}" | eval encrypted=tobase64( encrypt("AES", frombase64("mRcOlK9V47rjVL/RBYQYRw=="), binary("hello, world!") )) => "y7+NQQ9/9xGtbBq5pgBvCA==" # Refer to the usage of decrypt ().