**AES 256 in CBC mode with Dynamic IV (Hexadecimal AES key)**

**Secret Key**

73616d706c65496466634145536b6571

**Process of Decryption**

Dynamic IV Symmetric Encryption

**Pre Requisites**

* Hexadecimal AES key of size 16/24/32 Bytes.
* String payload to encrypt/decrypt

**Dynamic IV Generation**

A 16-byte random character string is generated in the code. All the characters in the resulting string will be in the ASCII Code Range from 47 to 126 randomly.

**Process of Encryption**

Name of class: **DynamicIVJce**

Name of method:  **encrypt**

Required Parameters and Datatypes:  String dataToEncrypt, String secretKey (Hexadecimal)

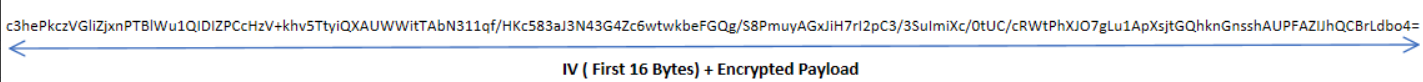
In the dynamic IV symmetric Encryption, the user will supply only the AES Key String in hexadecimal format at the time of encryption. Java code will first generate a random IV, then encrypt the string using this generated IV and the key provided by user to return the final output.

Resulting final output of code would be a Base 64 Encoded String which will be combination of following -

1. Dynamic IV String (generated inside the code)
2. Encrypted Payload

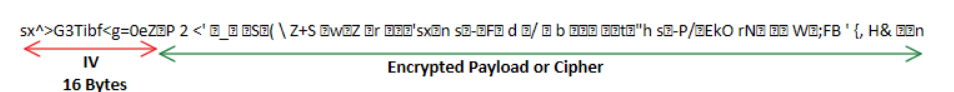
i.e. the dynamic IV (16 bytes) will be prefixed to the payload and this combination will together get encrypted. Hence first 16 bytes would contain IV and rest would be the payload.

**Example Output**



This Ouptut Base64 Encoded string contains the IV (first 16 bytes) and encrypted payload/cipher combination.

Decoding the Base64 String output returns:



**Process of Decryption**

Name of class: **DynamicIVJce**

Name of method:  **decrypt**

Required Parameters and Datatypes: String encrypted, String secretKey (Hexadecimal)

In the dynamic IV symmetric Decryption, the user will supply only the AES 256-bit Key at the time of Decryption along with the encrypted payload. Java code will first fetch the random IV from the first part of the splitted string. Then it will decrypt the string using this fetched IV and the key provided by user to return the final output. The Final output will be decrypted string.

**Plain text payload**

{

    "prefetchAccountReq": {

        "accountNumber": "999999999"

    }

}

**Base 64 encoded Encrypted payload**

**eX5LcEkzXnFWTjxCdX49Ol26cWPTlgOE507P+uvn3p7+h/9gSKjZtvYsEOJa7PPGIlWtWfuGgfw58qS+lGpGqgbWJaoWG/ez/IIpADfNqsghAiNOzWirPSI/AXT7Auu2**

**Reference:**

https://confluence.devops.idfcbank.com/pages/viewpage.action?spaceKey=API&title=AES+Encryption+-+IDFC+FIRST+BANK

<https://confluence.devops.idfcbank.com/pages/viewpage.action?spaceKey=API&title=AES+Encryption+-+IDFC+FIRST+BANK>



--------------------------------------------------End of document-------------------------------------------------------------