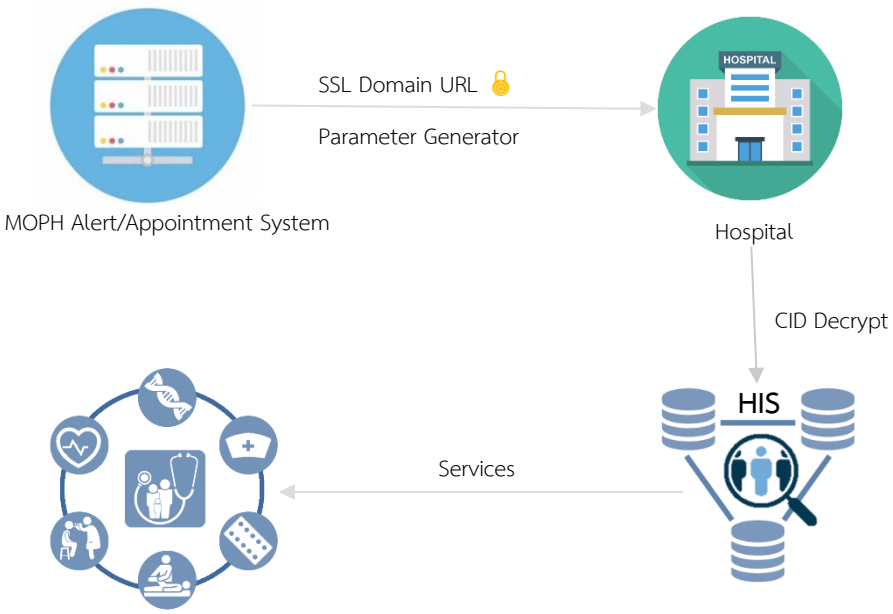
การขอใช้บริการระบบจองคิว บน Line OA หมอพร้อม

(กรณี หน่วยบริการพัฒนาระบบเอง)

(คู่มือการใช้ระบบสำหรับนักพัฒนาซอฟต์แวร์)

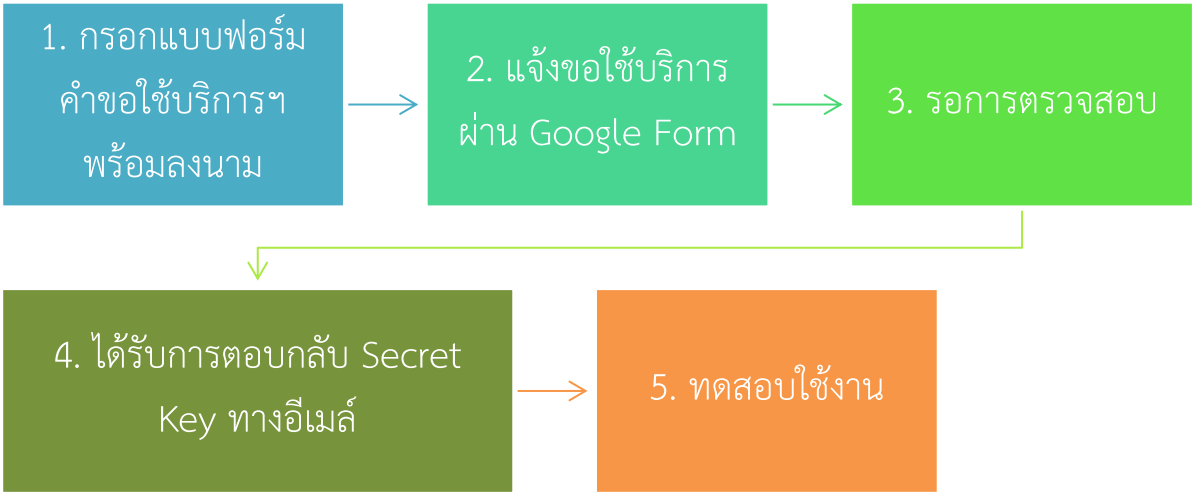
(ฉบับทดสอบ)

ปรับปรุง 21 กุมภาพันธ์ 2566



1. แนวทางการขอใช้บริการระบบจองคิว บน Line OA หมอพร้อม (กรณี หน่วยบริการพัฒนาระบบเอง)

การใช้บริการเชื่อมต่อระบบจองคิวของหน่วยบริการ/โรงพยาบาล ผ่าน Line OA หมอพร้อม กรณีหน่วยบริการพัฒนาระบบเอง ถือเป็นอีก 1 ช่องทางการติดต่อสื่อสารระหว่างหน่วยบริการกับประชาชาชนผู้ใช้บริการสุขภาพต่างๆ ในหน่วยบริการ ด้วยการส่งข้อมูล CID พร้อมลิงค์ (URL) โดเมนของหน่วยบริการ โดย Line OA หมอพร้อม จะเป็นตัวกลางเชื่อมโยงการบริการระหว่างหน่วยบริการสุขภาพ/โรงพยาบาลกับผู้ใช้บริการ/ประชาชนที่ลงทะเบียนใช้งาน Line OA หมอพร้อม สำเร็จ



เงื่อนไขและรายละเอียดการขอใช้บริการ

- ☐ กรอกแบบฟอร์มคำขอใช้บริการฯ พร้อมลงนามผู้อำนวยการหน่วยบริการสุขภาพ https://mohpromt.moph.go.th/mpc/mp-pf/moph-alert/
- ☐ ผู้ขอใช้บริการเป็นเจ้าหน้าที่ในหน่วยบริการ และผ่านการพิสูจน์และยืนยันตัวตน หมอพร้อม Digital ID สำเร็จ เท่านั้น
- ☐ แจ้งขอใช้บริการ ผ่าน Google Form : https://forms.gle/oCTM1nMMbwo3GHZD6
- ☐ คู่มือการใช้งาน (ฉบับทดสอบ) เพิ่มเติม เว็บไซต์หมอพร้อม https://mohpromt.moph.go.th
- ☐ หน่วยบริการมีการจดโดเมน (URL) อย่างเป็นทางการ ถูกต้อง และตรวจสอบได้ พร้อมทั้งผ่านการติดตั้ง SSL สำเร็จ เท่านั้น

2. ข้อมูลเบื้องต้น

2.1 การจัดเตรียมระบบเพื่อการขอใช้บริการ

ผู้ยื่นคำขอใช้บริการ

- เป็นเจ้าหน้าที่ของหน่วยบริการตามแบบคำขอใช้บริการฯ
- ผ่านการพิสูจน์และยืนยันตัวตน หมอพร้อม Digital ID สำเร็จ
- กรอกแบบฟอร์มคำขอใช้ฯ พร้อมลงนามผู้บริหารหน่วยงาน

ระบบสารสนเทศและฐานข้อมูลหน่วยบริการ

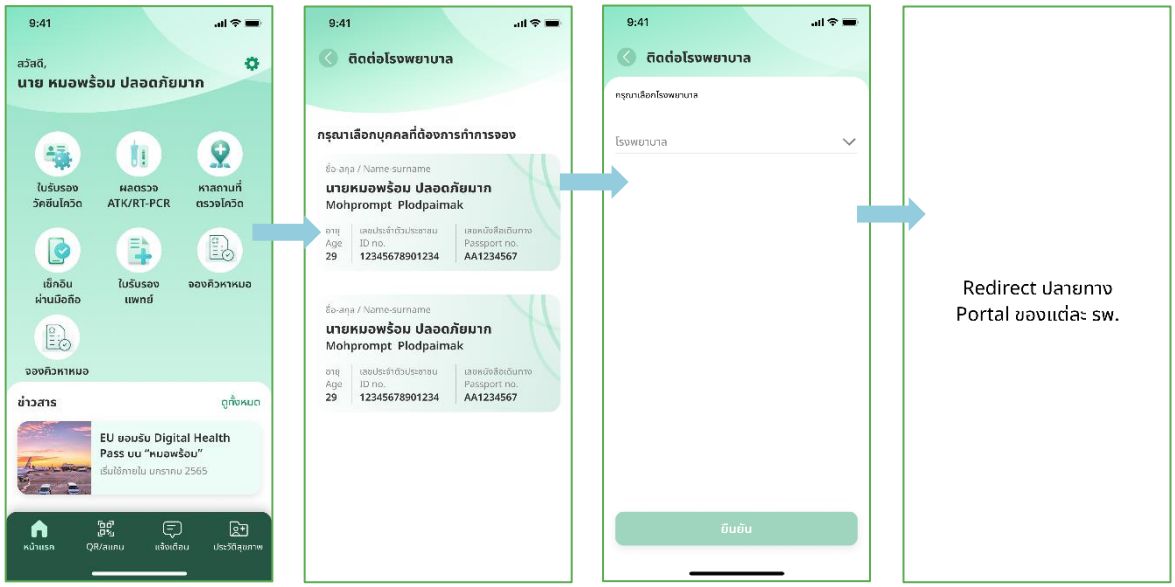
- มีระบบรับค่า CID และฟังก์ชันการถอดรหัส (Decrypt) ของผู้ใช้บริการ จาก หมอพร้อม
- มีระบบในการตรวจสอบผู้รับบริการของหน่วยบริการ หรือ HIS

การจดโดเมน (Domain Name Server) ของหน่วยบริการ

- มีการจดโดเมน (URL) สำหรับให้บริการจองระบบจองคิว อย่างเป็นทางการและถูกต้อง
- มีการติดตั้งใช้งาน SSL สำเร็จ เท่านั้น

2.2 การเรียกใช้งานฟังก์ชันจองคิวบน LINE OA หมอพร้อม(อยู่ระหว่างพัฒนา)

ผู้ใช้งาน LINE OA หมอพร้อม สามารถเรียกใช้งานระบบบริการจองคิวได้ผ่านเมนู จองคิวหาหมอ (อยู่ระหว่างพัฒนา) และสามารถเลือกหน่วยบริการได้



2.3 การถอดรหัส (Decrypt)

Envorinment	Secret Key :
Testing (UAT)	yE1l0WMOH2023Abs
Production (REAL)	xxx
Size Block Key	32 bytes/256 bits
Encrypt	AES-256-CBC

3. ตัวอย่างการ Redirect ข้อมูล

Method : GET

Host : Host : {URL}

Request

Request Header :

CID : udvldMPmV20XJu0q0V1Zg35Vz0GzWuLHsWyXeHTM01I=

Parameters	Requires	Type	Description
cid	ใช่	string	CID

```
Encrypted String : udvldMPmV20XJu0q0V1Zg35Vz0GzWuLHsWyXeHTM01I=
Decrypted String : 1209700368366
```

4. ตัวอย่าง การถอดหรัส (Decrypts) ด้วยภาษาต่างๆ

4.1 GOLANG

```go
package main

import (
    "bytes"
    "crypto/aes"
    "crypto/cipher"
    "encoding/base64"
    "fmt"
)

func main() {
    key := []byte("yE1l0WMOH2023Abs")

    result, err := AesEncrypt([]byte("exampleplaintext"), key)
    if err != nil {
```

```
        panic(err)
    }
    fmt.Println(base64.StdEncoding.EncodeToString(result))
//VRMzYJwT5xx0bvqud3Np+g==

    r, _ := base64.StdEncoding.DecodeString("VRMzYJwT5xx0bvqud3Np+g==") // use
CryptoJs encrypted
    //r := result  // decrypt go encrypted
    origData, err := AesDecrypt(r, key)
    if err != nil {
        panic(err)
    }
    fmt.Println(string(origData)) // exampleplaintext
}

func AesEncrypt(origData, key []byte) ([]byte, error) {
    block, err := aes.NewCipher(key)
    if err != nil {
        return nil, err
    }
    blockSize := block.BlockSize()
    origData = PKCS5Padding(origData, blockSize)
    iv := []byte("1234567812345678")
    blockMode := cipher.NewCBCEncrypter(block, iv)
    crypted := make([]byte, len(origData))
    blockMode.CryptBlocks(crypted, origData)
    return crypted, nil
}
func AesDecrypt(crypted, key []byte) ([]byte, error) {
    block, err := aes.NewCipher(key)
    if err != nil {
        return nil, err
    }
    iv := []byte("1234567812345678")
    blockMode := cipher.NewCBCDecrypter(block, iv)
    origData := make([]byte, len(crypted))
    blockMode.CryptBlocks(origData, crypted)
    origData = PKCS5UnPadding(origData)
    return origData, nil
}
func PKCS5Padding(ciphertext []byte, blockSize int) []byte {
    padding := blockSize - len(ciphertext)%blockSize
    padtext := bytes.Repeat([]byte{byte(padding)}, padding)
    return append(ciphertext, padtext...)
}
func PKCS5UnPadding(origData []byte) []byte {
    length := len(origData)
    unpadding := int(origData[length-1])
    return origData[:(length - unpadding)]
}
```

4.2 JavaScript

```
<template>
  <div>
  </div>
</template>

<script>
  import CryptoJs from "crypto-js"
  export default {
    name: "",
    data(){
      return {}
    },
    mounted(){
      let str = "exampleplaintext";
      let key = "yE1l0WMOH2023Abs";
      let r = this.encrypt(str, key);
      console.log(r); //dpepq82KQTeL+9qqcNJ5DMCFXlQH3Zc2Kh49+Ro1gHY=

      console.log(this.decrypt(r, key)) //exampleplaintext
    },
    methods: {
      encrypt(str, key){
        key = CryptoJs.enc.Utf8.parse(key);
        let iv = CryptoJs.enc.Utf8.parse("1234567812345678");
        let encrypted = CryptoJs.AES.encrypt(str, key, {
          iv: iv,
          padding: CryptoJs.pad.Pkcs7
        });
        return encrypted.toString();
      },
      decrypt(str, key) {
        key = CryptoJs.enc.Utf8.parse(key);
        let iv = CryptoJs.enc.Utf8.parse("1234567812345678");
        let encrypted = CryptoJs.AES.decrypt(str.toString(), key, {
          iv: iv,
          padding: CryptoJs.pad.Pkcs7
        });
        return encrypted.toString(CryptoJs.enc.Utf8);
      }
    }
  }</script>
```

4.3 Node JS

```
const run_test = () => {
  let str = "exampleplaintext";
  let key = "example key 1234";

  // Encrypt
  let r = encrypt(str, key);
  console.log(r); //VRMzYJwT5xx0bvqud3Np+g==


  // Decrypt
  console.log(decrypt(r, key)) //exampleplaintext
}
const encrypt = (str, key)=>{
  key = CryptoJs.enc.Utf8.parse(key);
  let iv = CryptoJs.enc.Utf8.parse("1234567812345678");
  let encrypted = CryptoJs.AES.encrypt(str, key, {
    iv: iv,
    padding: CryptoJs.pad.Pkcs7
  });
  return encrypted.toString();
}
const decrypt = (str, key) => {
  key = CryptoJs.enc.Utf8.parse(key);
  let iv = CryptoJs.enc.Utf8.parse("1234567812345678");
  let encrypted = CryptoJs.AES.decrypt(str.toString(), key, {
    iv: iv,
    padding: CryptoJs.pad.Pkcs7
  });
  return encrypted.toString(CryptoJs.enc.Utf8);
}
```

ข้อมูลเพิ่มเติม

GO : https://www.melvinvivas.com/how-to-encrypt-and-decrypt-data-using-aes

NodeJS : https://gist.github.com/brettscott/2ac58ab7cb1c66e2b4a32d6c1c3908a7

PHP : https://programmierfrage.com/items/aes-256-cbc-encryption-golang-and-php