

# **Aaple Sarkar Portal Technical Integration Document**

**Maharashtra Information Technology Corporation**

Ver.3.3

Prepared By

**Maharashtra Information Technology Corporation**

<b>Document Name :</b>	<b>Aaple Sarkar Portal Technical Integration Document</b>
<b>Version Number :</b>	<b>Ver.3.2</b>
<b>Release Date :</b>	<b>01<sup>th</sup> June 2018</b>
<b>Author Name :</b>	<b>Suresh Rasal &amp; Sudhir Chavan</b>
<b>Updated By :</b>	<b>Ashwini Bhoite</b>
<b>Owner of the document :</b>	<b>Maharashtra Information Technology Corporation</b>

## Index

<b>1. INTRODUCTION .....</b>	<b>4</b>
1.1 PURPOSE OF THIS DOCUMENT .....	4
<b>2. RIGHT TO SERVICES (RTS) INTEGRATION ARCHITECTURE: .....</b>	<b>5</b>
2.1 STEP 1: REGISTRATION OF CLIENT, DEPARTMENT & NOTIFIED SERVICE .....	6
CLIENT CODE REGISTRATION: .....	6
DEPARTMENT CODE REGISTRATION: .....	6
NOTIFIED SERVICE REGISTRATION: .....	6
2.2 STEP 2: GENERATION OF CLIENT CODE, DEPARTMENT CODE & SERVICE ID .....	7
CLIENT CONFIGURATION: .....	7
2.3 STEP 3: DEPARTMENTAL SERVICE ENABLE AT AAPLE SARKAR PORTAL .....	7
2.4 STEP 4: CALL PUSHWEBSERVICE() WEB-SERVICE FROM DEPARTMENT .....	7
2.5 STEP 5: VALIDATE REQUEST AT AAPLE SARKAR PORTAL END .....	7
2.6 STEP 6: RESPONSE OF PUSHWEBSERVICE() WEB-SERVICE .....	7
2.7 STEP 7: BUSINESS PROCESS AT DEPARTMENT END .....	8
2.8 STEP 8: DISPLAY SERVICE FORM .....	8
2.9 STEP 9: STORE REQUEST DATA INTO DEPARTMENTAL DATA BASE (DB) .....	8
2.10 STEP 10: GENERATE TRACK ID .....	8
2.11 STEP 11: CALL PULLWEBSERVICE() WEB-SERVICE FROM DEPARTMENT .....	8
2.12 STEP 12: BUSINESS PROCESS AT AAPLE SARKAR PORTAL END .....	8
2.13 STEP 13: STORE RESPONSE DATA INTO AAPLE SARKAR PORTAL DATA BASE (DB) .....	8
2.14 STEP 14: ACKNOWLEDGE MESSAGE AT AAPLE SARKAR PORTAL END .....	9
<b>3. STEP WISE DETAILED TECHNICAL PROCESS .....</b>	<b>10</b>
<b>3.1 STEP 4: CALL PUSHWEBSERVICE() WEB-SERVICE FROM DEPARTMENT .....</b>	<b>10</b>
3.1.1 ENCRYPTION ALGORITHM: .....	10
3.1.2 ENCRYPTION AND DECRYPTION CODE: .....	11
3.1.3 HOW TO DECRYPT QUERY STRING TOKEN: .....	14
3.1.4 SAMPLE FORMAT OF DECRYPTED TOKEN: .....	14
3.1.5 TOKEN PARAMETERS: .....	15
3.1.6 CHECKSUM VALIDATION: .....	15
3.1.7 STEPS TO CALCULATE CHECKSUM VALUE AND VALIDATION: .....	20
3.1.8 PUSH WEB-SERVICE CALLING: .....	23
REQUEST PARAMETERS: .....	23
RESPONSE PARAMETERS: .....	24
<b>3.2 STEP 2: CALL PULLWEBSERVICE() WEB-SERVICE FROM DEPARTMENT .....</b>	<b>25</b>
REQUEST STRING PARAMETERS: .....	25
3.2.1 CHECKSUM VALUE GENERATION PROCESS: .....	26
RAW STRING TO GENERATE CHECKSUM VALUE: .....	26
3.2.2 WEB METHOD CALL (UPDATE STATUS ON APPLICATION): .....	26

**3.3 STEP 3: PAYMENT POST DATA FROM DEPARTMENT .....28**

REQUEST STRING PARAMETERS: .....28

3.3.1 POST Data PROCESS (For PAYMENT): .....28

3.3.2 GET Data PROCESS .....31

## **1. Introduction**

**Right to Public Services** legislation in India comprises statutory laws which guarantee time bound delivery of services for various public services rendered by the Government to Citizen and provides mechanism for punishing the errant public servant who is deficient in providing the service stipulated under the statute. Right to Service legislation is meant to reduce corruption among the government officials and to increase transparency and public accountability.

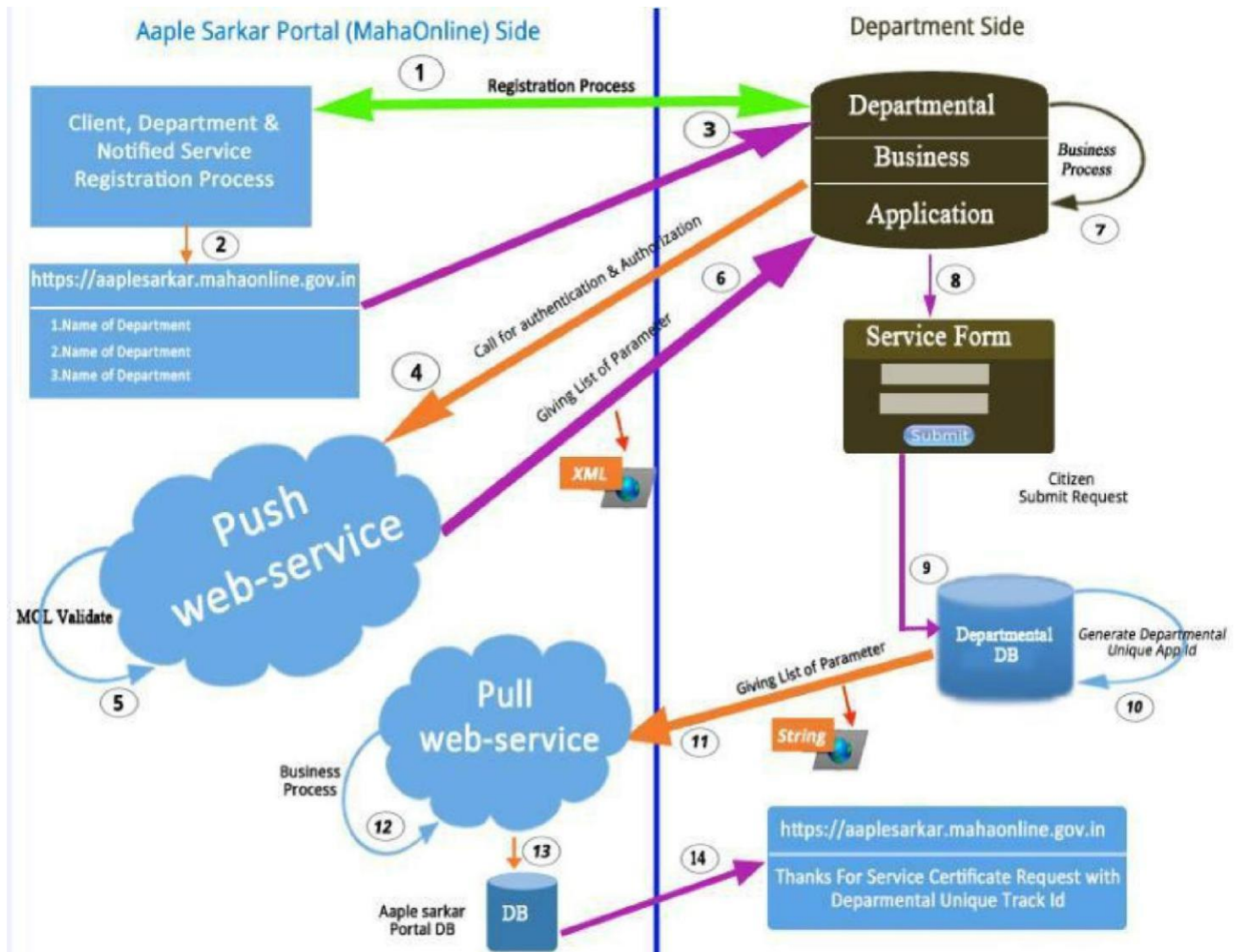
Maharashtra Govt. has been passed Right to Public Service Ordinance on dated 28<sup>th</sup> April 2015.

### ***1.1 Purpose of this document***

The purpose of this document is to describe in sufficient detail how the proposed system is to be design and development. The System technical integration design document for notified service integration with Aaple Sarkar Portal. It identifies the high-level system architecture.

This document outlines the proposed system technical integration design document for departmental notified services integration with Aaple Sarkar Portal.

## 2. Right To Services (RTS) Integration Architecture:



## MahaIT: Aaple Sarkar Portal Technical Integration

### 2.1 Step 1: Registration of Client, Department & Notified Service

This is process for registration of Client, Department & Notified Service with Aaple Sarkar Portal. MahaIT provides checklist for Step 1 process (i.e. Registration). Department need to fill provided checklist in English as well as Marathi language. This is one time master data repository process.

#### Client Code Registration:

Sr N	Type of Client	Client Name (English)	Client Name (Marathi)	Client Postal Address	Mobile / Telephone	Email Id	Domain Name	IP Address
1	Department / Corporation	Name of Client name in English	Name of Client name in Marathi	Detail postal address with PinCode, District & Taluka	Contact details	Email address for communication	<a href="http://www.kdmc.gov.in">www.kdmc.gov.in</a>	103.234.23.30

#### Department Code Registration:

Sr No	Department Name (English)	Department Name (Marathi)	Department Postal Address	Mobile / Telephone	Email Id
1	Name of Client name in English	Name of Client name in Marathi	Detail postal address with PinCode, District & Taluka	Contact details	Email address for communication

#### Notified Service Registration:

Sr No	Service Name (English)	Service Name (Marathi)	No of Disposal periods	Department Name	Competent Authority	First Appellant Officer	Second Appellant Officer
1		Birth Certificate	3	Medical	Sub-Registrar/ Senior Medical Officer / Medical Superintendent	Additional Medical Officer	Health Medical Officer (MOH)
2							

## MahaIT: Aaple Sarkar Portal Technical Integration

### **2.2 Step 2: Generation of Client Code, Department Code & Service Id**

Aaple Sarkar Portal (MahaIT) will generate Client Code, Department Code & Service Id as per provided information in Step 1. This is also one time activity.

#### **Client Configuration:**

Sr.No	Parameter	Value	Description
1	ClientCode	*****	MahaIT will provide the client code.
2	EncryptKey	*****	MahaIT will provide the Encrypt Key.
3	EncryptIV	*****	MahaIT will provide the Encrypt IV.
4	ChecksumKey	*****	MahaIT will provide the Checksum Key.

### **2.3 Step 3: Departmental Service Enable at Aaple Sarkar Portal**

MahaIT will enable list of Department Name link at Aaple Sarkar Portal, end user can apply his request on home screen.

### **2.4 Step 4: Call pushWebService() web-service from Department**

Department should be consume pushWebService() web-service (i.e. shared by MahaIT) for Authentication & Authorization of right request (to avoid fake request as well as user).

### **2.5 Step 5: Validate Request at Aaple Sarkar Portal End**

MahaIT will validate departmental request for authentication and authorization.

### **2.6 Step 6: Response of pushWebService() web-service**

MahaIT will give response as list of parameters (citizen register info) in XML format to Department through web-service.

### ***2.7 Step 7: Business Process at Department End***

Department should do business process after received list of parameters and show appropriate Notified Service Form to end user for service request.

### ***2.8 Step 8: Display Service Form***

Citizen will fill the Form as per his request and submit his request.

### ***2.9 Step 9: Store Request data into Departmental Data Base (DB)***

Department will receive citizen request and store into departmental DB.

### ***2.10 Step 10: Generate Track Id***

After storing citizen request into departmental DB. Department will generate Unique Application Indemnification Number (App Id).

### ***2.11 Step 11: Call pullWebService() web-service from Department***

Department should be consume pullWebService() web-service (i.e. shared by MahaIT) and provide list of parameters (String) to update status for acknowledgement.

### ***2.12 Step 12: Business Process at Aaple Sarkar Portal End***

Aaple Sarkar Portal (MahaIT) will receive list of parameters (i.e. request as a String) from Department and do the business process at Aaple Sarkar Portal end.

### ***2.13 Step 13: Store Response data into Aaple Sarkar Portal Data Base (DB)***

Aaple Sarkar Portal will receive departmental response and do business process and store in Aaple Sarkar Portal Data Base (DB).



***2.14 Step 14: Acknowledge Message at Aaple Sarkar Portal End***

Aaple Sarkar Portal / Department will display acknowledge message to end user with Unique Departmental Application Identification Number (App Id). End user can also receive acknowledge message through SMS & Email.

### **3. Step wise detailed Technical Process**

#### **3.1 Step 4: Call pushWebService() web-service from Department**

##### ***3.1.1 Encryption Algorithm:***

#### **Triple Data Encryption Standard algorithms (TripleDES)**

Triple DES is another mode of DES operation. It takes three 64-bit keys, for an overall key length of 192 bits. In Stealth, you simply type in the entire 192-bit (24 character) key rather than entering each of the three keys individually. The Triple DES DLL then breaks the user-provided key into three sub-keys, padding the keys if necessary so they are each 64 bits long.

1. Initialization vector (as per the Step I, MahaIT will provide)
2. Secret key (as per the Step I, MahaIT will provide)
3. Mode for operation of the symmetric algorithm (CBC in .net and "TripleDES / CBC / NoPadding")
4. Padding mode used in the symmetric algorithm (Zeros in .net and NoPadding in java)
5. Character Encoding (UTF-8)

### **3.1.2 Encryption and Decryption Code:**

#### **I. .Net Sample code**

##### **Encryption Function:**

```
String SimpleTripleDes(String Data, string strKey, string striv)
{
    byte[] key = Encoding.UTF8.GetBytes(strKey);
    byte[] iv = Encoding.UTF8.GetBytes(striv);
    byte[] data = Encoding.UTF8.GetBytes(Data);
    byte[] enc = new byte[0];
    TripleDES tdes = TripleDES.Create();
    tdes.IV = iv;
    tdes.Key = key;
    tdes.Mode = CipherMode.CBC; tdes.Padding =
    PaddingMode.Zeros; ICryptoTransform ict =
    tdes.CreateEncryptor();
    enc = ict.TransformFinalBlock(data, 0,
    data.Length); return ByteArrayToString(enc);
}
```

##### **Decryption Function:**

```
String SimpleTripleDesDecrypt(String Data, string strKey, string striv)
{
    byte[] key = Encoding.UTF8.GetBytes(strKey);
    byte[] iv = Encoding.UTF8.GetBytes(striv);
    byte[] data = StringToByteArray(Data); byte[]
    enc = new byte[0];
    TripleDES tdes = TripleDES.Create();
    tdes.IV = iv;
    tdes.Key = key;
    tdes.Mode = CipherMode.CBC; tdes.Padding =
    PaddingMode.Zeros; ICryptoTransform ict =
    tdes.CreateDecryptor();
    enc = ict.TransformFinalBlock(data, 0, data.Length);
    return Encoding.UTF8.GetString(enc).TrimEnd('\0');
}
```

**String to Byte Array:**

```
byte[] StringToByteArray(String hex)
{
    int NumberChars = hex.Length;
    byte[] bytes = new byte[NumberChars / 2];
    for (int i = 0; i < NumberChars; i += 2)
        bytes[i / 2] = Convert.ToByte(hex.Substring(i, 2),
        16); return bytes;
}
```

**Byte Array to String:**

```
string ByteArrayToString(byte[] ba)
{
    string hex = BitConverter.ToString(ba);
    return hex.Replace("-", "");
}
```

## II. JAVA Sample code

### **Encryption Function:**

```
String SimpleTripleDes(String Data, String strKey, String striv)
{
    Byte[] cipherBytes =
    StringToByteArray(Data); byte[] iv;
    Byte[] result = null;
    try {
        iv = striv.getBytes("UTF-8");

        byte[] keyBytes = strKey.getBytes("UTF-8");

        SecretKey aesKey = new SecretKeySpec(keyBytes, "TripleDES");

        Cipher cipher = Cipher.getInstance("TripleDES/CBC/Nopadding");
        cipher.init(Cipher.ENCRYPT_MODE, aesKey, new IvParameterSpec(iv));

        result = cipher.doFinal(cipherBytes);

    } catch (Exception e) {
        e.printStackTrace();
    }

    return ByteArrayToString(result);
}
```

### **Decryption Function:**

```
String SimpleTripleDesDecrypt(String Data, String strKey, String
striv) {
    Byte[] cipherBytes =
    StringToByteArray(Data); byte[] iv;
    Byte[] result = null;
    try {
        iv = striv.getBytes("UTF-8");
        byte[] keyBytes = strKey.getBytes("UTF-8");

        SecretKey aesKey = new SecretKeySpec(keyBytes, "TripleDES");

        Cipher cipher = Cipher.getInstance("TripleDES/CBC/Nopadding");
        cipher.init(Cipher.DECRYPT_MODE, aesKey, new IvParameterSpec(iv));

        result = cipher.doFinal(cipherBytes);
    }
```

```

    } catch (Exception e) {
        e.printStackTrace(); }

    return ByteArrayToString(result);
}

```

### **String to Byte Array:**

```

private static byte[] StringToByteArray(String s)
{
    if(s == null) return null;
    int l = s.length();
    if(l%2 == 1) return null;
    byte[] b = new byte[l/2];
    for(int i = 0 ; i < l/2 ;i++) {
        b[i] = (byte)Integer.parseInt(s.substring(i*2,i*2+2),16);
    }
    return b;
}

```

### **Byte Array to String:**

```

private static String ByteArrayToString(byte[] ba)
{
    String hex = new String(ba);
    System.out.println(" hex = " + hex );
    return hex.replace("-", "");
}

```

### **3.1.3 How to decrypt Query String Token:**

```

string str=Convert.ToString(Request.QueryString["str"]);

string RequestDecryStr=SimpleTripleDesDecrypt(str,ClientEncryptKey,ClientEncryptIV);

```

### **3.1.4 Sample format of Decrypted Token:**

```

string RequestDecryStr = UserId|TimeStamp|SessionID|ChecksumValue;

```

## MahaIT: Aaple Sarkar Portal Technical Integration

### 3.1.5 Token Parameters:

Sr. No.	Parameter	Value	Description
1	UserId	59fbcf05-f9f7-47d9-ae90-742122c6a292	32 bit unique number of each user
2	TimeStamp	5575676867789890	Current Date Time
3	SessionID	Fgdfg654656466666rgfg	User Session id
4	ChecksumValue	57789990	Check sum values calculated at Aaple Sarkar Portal end.
5	Authorization token / strServiceCookie	45fbcf05-f9f7-47d9-ae90-742122c6a852	Authorization token calculated at Aaple Sarkar Portal end.

### 3.1.6 Checksum Validation:

Checksum is process and it is a count of the number of bits in a transmission unit that is included with the unit so that the receiver can check to see whether the same number of bits arrived. If the counts match, it's assumed that the complete transmission was received.

A **Cyclic Redundancy Check (CRC)** or Polynomial Code **checksum** is a hash function designed to detect accidental changes to raw computer data, and is commonly used in digital networks and storage devices such as hard disk drives.

#### I. .Net Algorithm Code

##### CRC32 Algorithm:

```
using System;
using System.Collections;
using System.IO;
using System.Security.Cryptography;
using System.Text;

namespace Utility
{
    public class CRC32 : HashAlgorithm
    {
        public const UInt32 DefaultPolynomial = 0xedb88320;
        public const UInt32 DefaultSeed = 0xffffffff;

        private UInt32 hash;
        private UInt32 seed;
        private UInt32[] table;
        private static UInt32[] defaultTable;

        public CRC32()
        {
            table = InitializeTable(DefaultPolynomial);
            seed = DefaultSeed;
        }
    }
}
```

```

        Initialize();
    }

    public CRC32(UInt32 polynomial, UInt32 seed)
    {
        table = InitializeTable(polynomial);
        this.seed = seed;
        Initialize();
    }

    public override void Initialize()
    {
        hash = seed;
    }

    protected override void HashCore(byte[] buffer, int start, int length)
    {
        hash = CalculateHash(table, hash, buffer, start, length);
    }

    protected override byte[] HashFinal()
    {
        byte[] hashBuffer =
            UInt32ToBigEndianBytes(~hash); this.HashValue =
            hashBuffer; return hashBuffer;
    }

    public override int HashSize
    {
        get { return 32; }
    }

    public static UInt32 Compute(byte[] buffer)
    {
        return ~CalculateHash(InitializeTable(DefaultPolynomial),
            DefaultSeed, buffer, 0, buffer.Length);
    }

    public static UInt32 Compute(UInt32 seed, byte[] buffer)
    {
        return ~CalculateHash(InitializeTable(DefaultPolynomial),
            seed, buffer, 0, buffer.Length);
    }

    public static UInt32 Compute(UInt32 polynomial, UInt32 seed, byte[] buffer)
    {
        return ~CalculateHash(InitializeTable(polynomial), seed, buffer,
            0, buffer.Length);
    }

    private static UInt32[] InitializeTable(UInt32 polynomial)

```



## MahaIT: Aaple Sarkar Portal Technical Integration

```

{
    if (polynomial == DefaultPolynomial && defaultTable != null)
        return defaultTable;

    UInt32[] createTable = new UInt32[256];
    for (int i = 0; i < 256; i++)
    {
        UInt32 entry = (UInt32)i;
        for (int j = 0; j < 8; j++)
            if ((entry & 1) == 1)
                entry = (entry >> 1) ^ polynomial;
            else
                entry = entry >> 1;
        createTable[i] = entry;
    }

    if (polynomial == DefaultPolynomial)
        defaultTable = createTable;

    return createTable;
}

private static UInt32 CalculateHash(UInt32[] table, UInt32 seed, byte[]
buffer, int start, int size)
{
    UInt32 crc = seed;
    for (int i = start; i < size; i++)
        unchecked
        {
            crc = (crc >> 8) ^ table[buffer[i] ^ crc & 0xff];
        }
    return crc;
}

private byte[] UInt32ToBigEndianBytes(UInt32 x)
{
    return new byte[] {
        (byte)((x >> 24) & 0xff),
        (byte)((x >> 16) & 0xff),
        (byte)((x >> 8) & 0xff),
        (byte)(x & 0xff)
    };
}
}
}

```

## II. JAVA Algorithm Code

### **CRC32 Algorithm:**

```
package org.com;

import java.util.zip.Checksum;

public class CRC32 implements Checksum
{
    /** The crc data checksum so far. */
    private int crc = 0;

    /** The fast CRC table. Computed once when the CRC32 class is
    loaded. */ private static int[] crc_table = make_crc_table();

    /** Make the table for a fast CRC. */
    private static int[] make_crc_table ()
    {
        int[] crc_table = new int[256];
        for (int n = 0; n < 256; n++)
        {
            int c = n;
            for (int k = 8; --k >= 0; )
            {
                if ((c & 1) != 0)
                    c = 0xedb88320 ^ (c >>> 1);
                else
                    c = c >>> 1;
            }
            crc_table[n] = c;
        }
        return crc_table;
    }

    /**
     * Returns the CRC32 data checksum computed so
     far. */
    public long getValue ()
    {
        return (long) crc & 0xffffffffL;
    }

    /**
     * Resets the CRC32 data checksum as if no update was
     ever called. */
    public void reset () { crc = 0; }
```

## MahaIT: Aaple Sarkar Portal Technical Integration

---

```
/**
 * Updates the checksum with the int bval.

 * @param bval (the byte is taken as the lower 8 bits of bval)
 */

public void update (int bval)
*{
    int c = ~crc;
    c = crc_table[(c ^ bval) & 0xff] ^ (c >>>
8); crc = ~c;
}

/**
 * Adds the byte array to the data checksum.

 * @param buf the buffer which contains the data
 * @param off the offset in the buffer where the data starts
 * @param len the length of the data
 */
public void update (byte[] buf, int off, int len)
{
    int c = ~crc; while
    (--len >= 0)
        c = crc_table[(c ^ buf[off++]) & 0xff] ^ (c >>>
8); crc = ~c;
} }

/**
 * Adds the complete byte array to the data checksum.
 */
public void update (byte[] buf) { update(buf, 0, buf.length); }
```

**3.1.7 Steps to calculate CheckSum Value and Validation:****I. .Net Sample Code**

```
string[] param = RequestDecryStr.Split('|');
if (param != null && param.Length > 0)
{
    string _UsrId = string.Empty;
    string _UsrTimeStamp = string.Empty;
    string _UsrSession = string.Empty;
    string _ClientCheckSumValue = string.Empty;
    string _trackid = string.Empty;
    string _strServiceCookie = string.Empty; -- New Addition

    string _ChkValueRawData = string.Empty;
    _UsrId = param[0];
    _UsrTimeStamp = param[1];
    _UsrSession = param[2];
    _ClientCheckSumValue = param[3];
    _strServiceCookie = param[4]; -- New Addition

    _ChkValueRawData = String.Format("{0}|{1}|{2}|{3}|{4}",
    _UsrId, _UsrTimeStamp, _UsrSession, _ ClientCheckSumValue,
    _strServiceCookie); ---New Addition

    string _CaluculatedCheckSumValue =
    GenerateCheckSumValue(_ChkValueRawData);

    if (_ClientCheckSumValue.Equals(_CaluculatedCheckSumValue))
    {
        Then here call the web service
    }
    else
    {
        "Invalid checksum value";
    }
}
else
{
    "Invalid Request param";
}
```

```
public static string GenerateChecksumValue(string reqStr)
{
    try
    {
        System.Text.AsciiEncoding AsciiEncoding = new System.Text.AsciiEncoding();
        UInt32 checksumvalue = CRC32.Compute(AsciiEncoding.GetBytes(reqStr)); // String to
bytes        return checksumvalue.ToString();

    }
    catch (Exception)
    {
        return string.Empty;
    }
    finally
    {
    }
}
```

## II. JAVA Sample Code

```
String ReturnURL = "ClientRegistrationReturnURL";
String ClientCode = "ClientCode";
String ClientEncryptKey = "ClientEncryptKey";
String ClientEncryptIV = "ClientEncryptIV";

String _Checksumkey = "Checksum";

String CheckSumRequestValue = "", CheckSumRequestString =
"", CalCheckSumValue = "", culture = "";

String param[] = decryptData.split("\\|");
String userId = param[0];
String userTimeStamp = param[1];
String userSession = param[2];
String clientCheckSumValue = param[3];
String strServiceCookie = param[4]
String checksumkey = "GMahwp8v3G7M";

String checkValueRowData =
userId+"|"+userTimeStamp+"|"+userSession+"|"+ checksumkey+"|"+
strServiceCookie;

ValidateChecksum vchkSum = new ValidateChecksum();

String caluculatedChecksumValue =
String.valueOf(vchkSum.CalculateChecksum(checkValueRowData));

if (clientCheckSumValue.equals(caluculatedChecksumValue))
{
    Then here call the web service
}
else
{
    "Invalid checksum value";
}
```

## MahaIT: Aaple Sarkar Portal Technical Integration

```

    }
} catch (Exception e)
{
    e.printStackTrace();
}

```

### 3.1.8 Push Web-service calling:

**Testing URL:** [http://testcitizenservices.MahaITgov.in/Dept\\_Authentication.asmx?WSDL](http://testcitizenservices.MahaITgov.in/Dept_Authentication.asmx?WSDL)

**Web method:**

String xmlResponse = **GetParameterNew** (str,ClientCode); -- New Addition

String ResponseXML = SimpleTripleDesDecrypt(xmlResponse, EncryptKey, EncryptIV);

#### Request Parameters:

Sr	Parameter	Value	Description
1	Str	63458359jmfkklflf8499090	Token received from Aaple Sarkar Portal in query string.
2	ClientCode	*****	Provided by MahaIT

#### Response Parameters:

Sr No	Parameter Name	Value	Description
1	Response	Success/failed/Error	Status of request
2	UserID	"NNNN"	User Identification Number
3	UsertypeName	User	Type of User
4	Password	F91E15DBEC69FC40F81F0876E7009648	Password in MD5 encrypted Format
5	PasswordChanged	True/false	Is Password changed?
6	IsActive	True/ false	Is Active?
7	VerifyStatus	Verified/Not Verified	Is Status Verified or Not?
8	EmailID	<a href="mailto:suresh.rasal@MahaIT.gov.in">suresh.rasal@MahaIT.gov.in</a>	Email Id of User
9	MobileNo	9888898888	Mobile No of User
10	Salutation	MR/MISS/MRS	Salutation of User

## MahaIT: Aaple Sarkar Portal Technical Integration

11	FullName	Xxxxxxx	Full Name of User
12	FullName_mr	Xxxxxxx	Full Name in Marathi
13	Age	30	Age of User
14	Gender	M/F/T	Gender of User
15	UIDNO	127890865443	Aadhaar No of User
16	PANNo	ASXh4789N	PAN No of User
17	DOB	14-12-1984	Date of Birth of User
18	AddrCareOf	Xxxxxx	Address Care of User
19	AddrCareOf_LL	Xxxx	Address Care of User in Marathi
20	AddrBuilding	Xxxxxx	Building Address of User
21	AddrBuilding_LL	Xxxxxxx	Building Address of User in Marathi
22	AddrStreet	Xxxxxxx	Street Address of User
23	AddrStreet_LL	Xxxx	Street Address of User in Marathi
24	AddrLandmark	Xxx	Address Landmark
25	AddrLandmark_LL	Xxx	Address Landmark in Marathi
26	AddrLocality	Xxxxxx	Address Locality
27	AddrLocality_LL	Xxxxxx	Address Locality in Marathi
28	PinCode	445656	Pin Code of User
29	DistrictID	345	District Code
30	TalukaID	2345	Taluka Code
31	VillageID	23456	Village Code
32	FatherFullName	Xxxxxx	Father Full Name
33	FatherFullName_mr	Xxxx	Father Full Name in Marathi
34	Father_Salutation	MR/MISS/MRS	Father Salutation
35	TrackId	160115001100000001	MahaIT Track ID

```

<ResMessage>
  <Response>Success</Response> <UserID>59fbcf05-f9f7-
47d9-ae90-742122c6a292</UserID>
  <UsertypeName>Super</UsertypeName>
  <Username>sadmin</Username>
  <Password>F91E15DBEC69FC40F81F0876E7009648</Password>
  <PasswordChanged>true</PasswordChanged>
  <IsActive>true</IsActive>
  <VerifyStatus>Verified</VerifyStatus>
  <EmailID>suresh.rasal@MahaIT.gov.in</EmailID>
  <MobileNo>9888898888</MobileNo>
  <Salutation>CA</Salutation>
  <FullName>Suresh Rasal</FullName>
  <FullName_mr>                </FullName_mr>
  <Age>29</Age>
  <Gender>M</Gender>

```



```
<UIDNO>NA</UIDNO>
<PANNo>NA</PANNo>
<DOB>01/01/1988</DOB>
<AddrCareOf>Bandra Mumbai</AddrCareOf>
<AddrCareOf_LL>          </AddrCareOf_LL>
<AddrBuilding>godrej</AddrBuilding>
<AddrBuilding_LL>        </AddrBuilding_LL>
<AddrStreet>godrej</AddrStreet>
<AddrStreet_LL>          </AddrStreet_LL>
<AddrLandmark>godrej</AddrLandmark>
<AddrLandmark_LL>        </AddrLandmark_LL>
<AddrLocality>godrej</AddrLocality>
<AddrLocality_LL>        </AddrLocality_LL>
<PinCode>400089</PinCode>
<DistrictID>530</DistrictID>
<TalukaID>4284</TalukaID>
<VillageID>567154</VillageID>
<FatherFullName>Ashok Rao</FatherFullName>
<FatherFullName_mr>      </FatherFullName_mr>
<Father_Salutation>Shri</Father_Salutation>
<TrackId>160115001100000001</TrackId>
</ResMessage>
```

### 3.2 Step 2: Call pullWebService() web-service from Department

**Testing URL :-** [http://testcitizenservices.mahaitgov.in/Dept\\_Authentication.asmx?WSDL](http://testcitizenservices.mahaitgov.in/Dept_Authentication.asmx?WSDL)

**Request String Parameters:**

Sr No	Parameter	Value	Description
1	Track ID	16011500100000000001	MahaIT Track ID (Generated in first request)
2	Client Code	XXXXXX	MahaIT provided client code
3	User ID	59fbcf05-f9f7-47d9-ae90-742122c6a292	User Id provided in First Request
4	ServiceID	XXXX	4 digit service id as per given at client registration time
5	ApplicationID	XXXXXXXXXXXXXXXXXXXX	Your Unique Order ID/Application ID/Application No.
6	Payment Status	Y/N	Y = Yes , N = No
7	Payment Date	YYYY-MM-DD	Provide the date in "YYYY-MM-DD" format only NA if not available
8	DigitalSign Status	Y/N	Y = Yes , N = No
9	DigitalSign Date	YYYY-MM-DD	Provide the date in "YYYY-MM-DD" format only or NA if not available
10	Estimated ServiceDays	7	only Integer Number or 0
11	Estimated Service Date	YYYY-MM-DD	Provide the date in "YYYY-MM-DD" format only NA if not available.
12	Amount	100.56	Amount in decimal format.
13	Request Flag	0 or 1 or 2	0 = to update both status, 1 = to update payment status, 2 = to update digital signature.
14	ApplicationStatus	1,2,3,4,5	1 = Document Pending 2 = Payment Pending 3 = Under Scrutiny 4 = Application Approved

## Maha: Aaple Sarkar Portal Technical Integration

			5 = Application Rejected
15	Remark	Alphanumeric	Status of message
16	UD1(Compulsory for UD Department)	123 /Int	MahaIT ULB id(Generated in first request)
17	UD2(Compulsory for UD Department)	123 /Int	MahaIT ULBDistrict(Generated in first request)
18	UD3	NA	User Defined Field 3
19	UD4	NA	User Defined Field 4
20	UD5	NA	User Defined Field 5
21	CheckSum	648349349	Check Sum Value calculated at department end.

### 3.2.1 CheckSum value generation process:

#### Raw string to generate checksum value:

```
String stringbeforechecksum = Track ID|Client Code|User
ID|ServiceID|ApplicationID|Payment Status|Payment Date|DigitalSign
Status|DigitalSign Date|Estimated ServiceDays|Estimated Service Date
|Amount|Request Flag| ApplicationStatus | Remark |UD1|UD2|UD3|UD4|UD5|ChecksumKey
```

```
String checksumvalue = GenerateChecksumValue(stringbeforechecksum);
```

#### Final string with checksum value:

```
String finalstring = Track ID|Client Code|User
ID|ServiceID|ApplicationID|Payment Status|Payment Date|DigitalSign
Status|DigitalSign Date|Estimated ServiceDays|Estimated Service Date
|Amount|Request Flag| ApplicationStatus | Remark |UD1|UD2|UD3|UD4|UD5|Checksumvalue
```

### 3.2.2 Web method Call (update status on Application):

1. Encrypt the final string generated in above point 3.2.1 and then call the web service to update the status of application on Aaple Sarkar Portal.
2. String EncyKey = SimpleTripleDes(Response, EncryptKey, EncryptIV);

#### Web method Parameters:

Parameter	Value	Description
EncyKey	Hdfgkdfgkkk859595mfjkfkklflf	Encrypted string generated in point no 3.2.2

## MahaIT: Aaple Sarkar Portal Technical Integration

---

DeptCode    XXXXXX

MahaIT provided client code

---

String Response= SetAppStatus (EncyKey, DeptCode);

**If updated successfully response will be:**

<ResMessage> <status>Success/Fail</status> </ResMessage>

**If any error in request parameter then response will be:**

<ResMessage> <error>Error Message</error> </ResMessage>

**If any error in request parameter then response will be:**

<ResMessage> <error>Error Message</error> </ResMessage>

### 3.3 STEP 3: PAYMENT POST DATA FROM DEPARTMENT

This in process we are going to take payment from department.

First API which we are going sent request is **ValidateRequest** API in this part we are going to send below parameters to API.

#### 3.2.1 Request String Parameters:

Sr No	Parameter	Value	Description
1	ClientCode	XXXXXX	MahaIT provided client code
2	Checksum	648349349	Check Sum Value calculated at department end.
3	ServiceID	XXXX	4 digit service id as per given at client registration Time
4	ApplicationID	XXXXXXXXXXXXXXXXXXXX	Your Unique Order ID/Application ID/Application No.
5	Districtid	XXXX	MahaIT provided District code.
6	ApplicationDate	YYYY-MM-DD	Provide the date in "YYYY-MM-DD" format only.
7	TrackID	16011500100000000001	MahaIT Track ID (Generated in first request)
8	User ID	XXXXXXXXXXXXXXXXXXXX	AS User Identification Number.
9	MobileNo	9888898888	Mobile No of User
10	Name	Xxxxxxx	Full Name of User
11	Returnurl	XXXXXXXXXXXXXXXXXXXX	After AS payment where the page land with payment data.
12	UD1	NA	User Defined Field 1
13	UD2	NA	User Defined Field 2
14	UD3	NA	User Defined Field 3
15	UD4	NA	User Defined Field 4
16	UD5	NA	User Defined Field 5

### Response Key and Errors

If the request is valid then you will receive Key and that time Error will be empty else if the request is not valid then you will receive Errors with an empty key.

## Creating Request for API

In this part we are going to Concat all parameters which we are going to send in below format with "|" separated.

### Format

String **finalstring** =

ClientCode|Checksum|ServiceID|ApplicationID|Districtid|ApplicationDate|TrackID|APUserId|MobileNo|Name|returnurl|UD1|UD2|UD3|UD4|UD5;

After Concating Parameters we are next going to set this parameter to a Class property.

### Format

```
public class WebString
{
    public string webstr { get; set; }
    public string deptcode { get; set;}
}
```

Next we are going to assign finalstring and Department code to WebString class as show below.

```
String EncyKey = iasObj.SimpleTripleDes(finalstring, EncryptKey, EncryptIV);
WebString objstr = new WebString();
objstr.webstr = EncyKey;
objstr.deptcode = ClientCode;
return objstr;
```

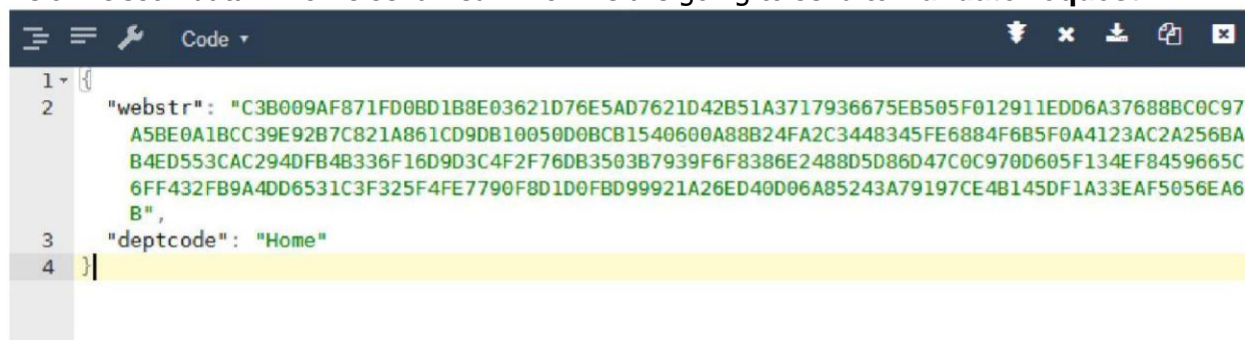
After assigning values next we are going to Serialize **webstring** class and post data using **WebClient** but before posting data we must set Headers which we given below.

1. **Content-Type:application/json**
2. **Accept:application/json**

## Code Snippet For Send Request

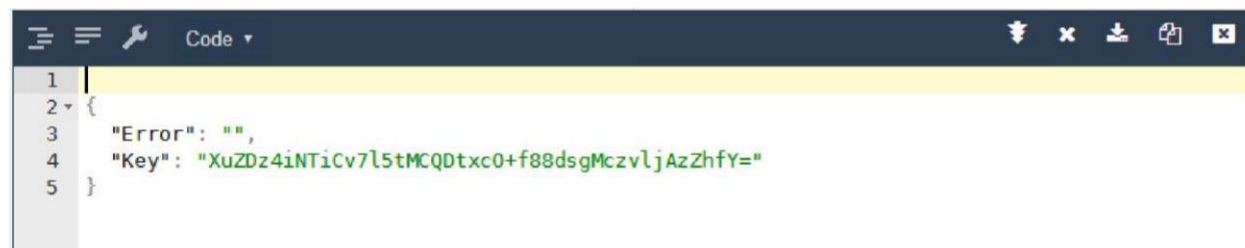
```
using (var client = new WebClient())
{
    client.Headers.Add("Content-Type:application/json");
    client.Headers.Add("Accept:application/json");
    var serializeddata = JsonConvert.SerializeObject(objstr);
    message =
client.UploadString("http://testcitizenservices.MahaITgov.in/en/OutPayment/ValidateRequest",
serializeddata);
}
```

Below is Json data which is serialized which we are going to send to **ValidateRequest** API.



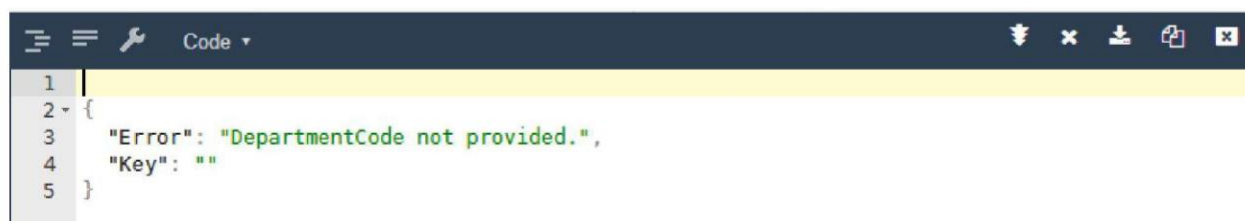
```
1 {
2   "webstr": "C3B009AF871FD0BD1B8E03621D76E5AD7621D42B51A3717936675EB505F012911EDD6A37688BC0C97
   A5BE0A1BCC39E92B7C821A861CD9DB10050D0BCB1540600A88B24FA2C3448345FE6884F6B5F0A4123AC2A256BA
   B4ED553CAC294DFB4B336F16D9D3C4F2F76DB3503B7939F6F8386E2488D5D86D47C0C970D605F134EF8459665C
   6FF432FB9A4DD6531C3F325F4FE7790F8D1D0FBD99921A26ED40D06A85243A79197CE4B145DF1A33EAF5056EA6
   B",
3   "deptcode": "Home"
4 }
```

After Posting Request if request is **valid** then you will get **Key** as response in following format.



```
1 {
2   "Error": "",
3   "Key": "XuZDz4iNTiCv7l5tMCQDtxc0+f88dsgMczvIjAzZhfy="
4 }
```

After Posting Request if request is **Invalid** then you will get **Error** as response in following format.



```
1 {
2   "Error": "DepartmentCode not provided.",
3   "Key": ""
4 }
```

After you Get Key Next we are going to redirect request to payment URL with parameters in querystring as show in below code snippet.

### Code Snippet for Send Request

```
return Redirect(http://testcitizenservices.MahaITgov.in/en/OutPayment/Pay
+ "? webstr=" + "webstring" + "&DeptCode=" + "Department Code" + "&Authentication" + "Key");
```

After redirecting you will see below screen along with payment data.



The screenshot shows the MahaIT portal interface. At the top, there is a header with the MahaIT logo, a user profile section (MahaIT शासन, सुस्वागतम्, अश्विनी), and a language selector (English). Below the header, there is a navigation bar with the text 'आपले सरकार' and 'महाराष्ट्र लोकसेवा हक्क अधिनियम'. The main content area displays a table with application details and payment information.

Application ID: / ॲप्लिकेशन क्र.	१८५१८९००३३२२००१८F०३F
Service Name: / दाखल्याचे नाव	License for amplified sound system
सेवा विभागाचे नाव	Home
Applicant Name / अर्जदाराचे नाव	ASHWINI
सेवेसाठी कालमर्यादा	
सरकारी शुल्क	०.००
सेवा शुल्क	२०.००
राज्य वस्तु आणि सेवा कर (९%)	१.८०
केंद्र वस्तु आणि सेवा कर (९%)	१.८०
एकूण रक्कम	२३.००

Below the table, there is a green 'Confirm' button. At the bottom, a note states: 'Use the Application Id to track the current status of your application at section 'Track your Application' on home page of'.

### **3.3.2 GET Data PROCESS:**

After Payment department will get response on their provided return URL. Following are parameters are to be sent in encrypted format querystring.



Sr No	Parameter	Value	Description
1	ClientCode	XXXXXX	MahaIT provided client code
2	ServiceID	XXXX	4 digit service id as per given at client registration time
3	ApplicationID	XXXXXXXXXXXXXXXXXXXX	Your Unique Order ID/Application ID/Application No.
4	PaymentTransactionid	16011500100000000001	Payment Transaction ID
5	BankRefID	XXXX	Bank Reference ID
6	BankRefNo	XXXX	Bank Reference Number
7	BankID	XXXX	Bank Id
8	PaymentDate	XXXXXXXXXXXXXXXXXXXX	Online Payment date
9	PaymentStatus	True/False	Payment Status
10	TotalAmount	XXXX	Total Paid amount

Finalstring

=Home|3220|1851890033220018F03F|180601125100471865|100001196046|1000011960461|Ato  
m Bank|01/06/2018 00:00:00|True|23.60

String EncyKey = iasObj.SimpleTripleDes(finalstring, EncryptKey, EncryptIV);

Post\_Url = . Department provided return URL + "?str=" + EncyKey;

~~~~~ Thanks ~~~~~

