



Invoicing API Specification (Version 2.3)

Last updated on 18th Sep 2024

RAZER MERCHANT SERVICES	
  	<p>J-39-1, Block J, Persiaran Multimedia, i-City, 40000 Shah Alam, Selangor, Malaysia.</p> <p>+ (603) - 5521 8438</p> <p>support@Fiuu.com</p> <p>Fiuu.com</p> 
	<p>Social Networks</p> <p> https://twitter.com/FiuuPayment https://facebook.com/FiuuPayment https://youtube.com/@FiuuPayment https://instagram.com/FiuuPayment https://linkedin.com/company/FiuuPayment </p>
	<p>Developer Platforms</p> <p> https://t.me/FiuuDeveloperForum (for tech support) https://github.com/FiuuPayment Mobile XDK, seamless and in-page checkout, and many shopping carts payment plugin/module/addon/extension are available </p>
	 

Revision History

Date	Version	Author(s)
2014/05/09	1.0	Tan Tek Jau
2014/05/23	1.1	Khairil
2014/05/27	1.2	Khairil
2016/09/02	1.3	Amalina
2017/07/18	1.4	Clement
2018/04/12	1.5	Hafeez
2018/07/30	1.6	Firdaus
2018/08/02	1.7	Firdaus
2018/08/09	1.8	Firdaus
2021/01/05	1.9	Hafeez
2022/04/13	2.0	Choo Yao Song
2022/07/27	2.1	Choo Yao Song
2023/09/27	2.2	Safwan Hakim
2024/09/18	2.3	Puvaan Raaj

Table of Contents

[Objective](#)

[Overview](#)

[Communication Protocol](#)

[Security & Data Integrity](#)

[1. Private Hash Key \(MsgID\)](#)

[2. Verify Key \(verify_key\)](#)

[3. Checksum String \(checksum\)](#)

[Invoicing APIs](#)

[1. Creating Payment Request](#)

[2. Status Query](#)

[3. Edit Invoice Details](#)

[4. Delete Payment Link](#)

[Error Code & Message](#)

Objective

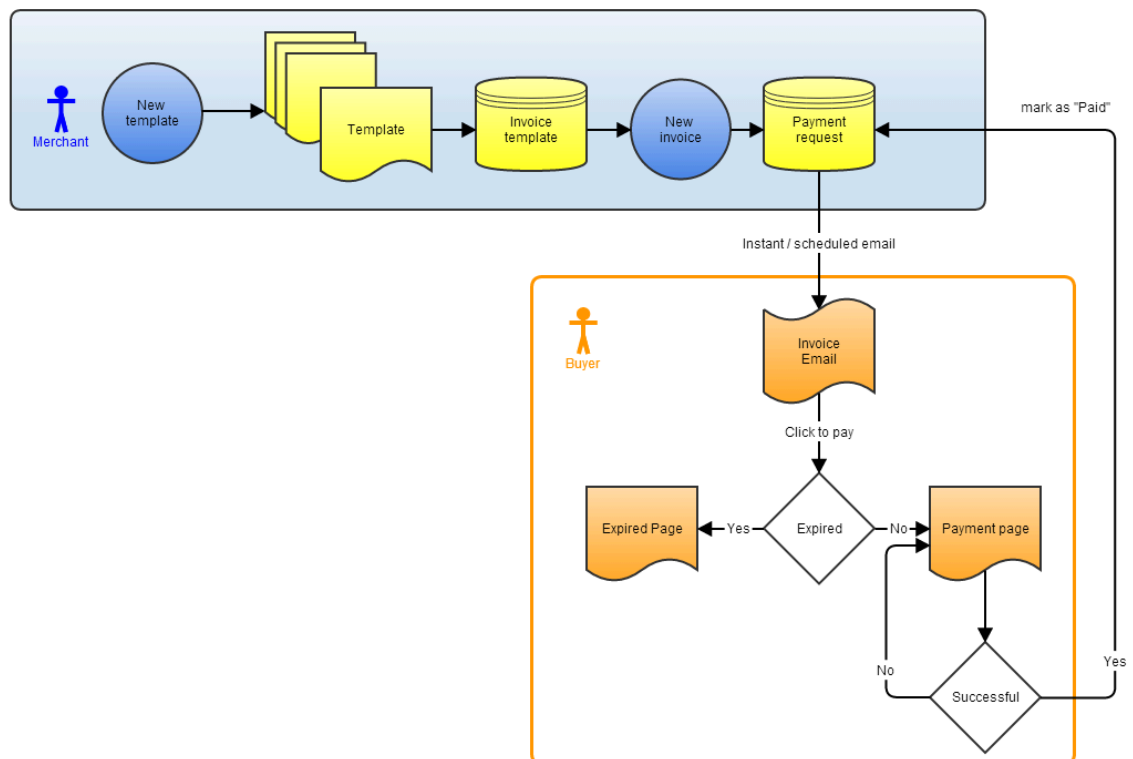
This is a server-to-server APIs for merchant/developer to integrate Fiuu invoicing service and offer payment options by sending payment requests to the merchant's customer.

Merchant or developer is required to read the security section, validate the data properly to ensure integrity to avoid MITM (man-in-the-middle) attack.

DO NOT trust the result, especially the payment status, if any of the parameters has been modified, out of range, or not in the proper format as defined in this document.

All credential information should be encrypted or hashed or encoded properly before hardcoding in the application or passing through the internet.

Overview



Communication Protocol

Protocol : JSON over HTTPS(SSL) using GET & POST method

API URL : <https://api.fiuu.com/RMS/API/Invoice/index.php>

API URL (sandbox) : <https://sandbox.fiuu.com/RMS/API/Invoice/index.php>

Security Hash : SHA1 & MD5

Request Method : URL?op=[func]

** GET only allow the "op" parameter and case sensitive (all other attempt will trigger the alarm)*

POST variables

Variable name	Data Type	Attribute / Example	Description
params	JSON encoded string	{"var1":val1_int,"var2","val2_str",..}	JSON encoded string of all the functions variable
checksum	char(32)	md5(\$params + sha1(md5(\$verify_key)))	MD5 & SHA1 hashing checksum

Result variables (Output as JSON encoded plaintext)

Variable name	Data Type	Attribute / Example	Description
results	JSON encoded string	{"var1":val1_int,"var2","val2_str",..}	JSON encoded string of all the result variables
checksum	char(32)	md5(\$results + sha1(md5(\$verify_key)) + \$datetime)	MD5 & SHA1 hashing checksum
datetime	Datetime	YYYY-MM-DD HH:mm:ss	date & time that send out the request

Security & Data Integrity

1. Private Hash Key (*MsgID*)

Embedded in the *params* variable. This can be treated as a message id or token for each request, which RMS will echo back to the merchant in the result for each request, or vice versa.

DO NOT disclose how to generate this to any party. This is a merchant self-known method and only merchants know how to verify the validity.

2. Verify Key (*verify_key*)

This is the checksum password that is known by both parties, **merchant & RMS** only, and used in generating and verifying checksum string for both sides. **DO NOT disclose this to 3rd parties.**

If this key is compromised, a new key must be generated immediately by RMS and the merchant needs to change the code accordingly to resume the payment service.

3. Checksum String (*checksum*)

MD5 & SHA1 hashstring formed by different combinations of all parameters (*params*), verify key (*verify_key*) and *datetime*. Merchant is required to verify the checksum and validate all incoming data before using it. This checksum is to ensure the data integrity and the source to avoid man-in-the-middle attack. **DO NOT disclose the checksum formula to the public.**

Invoicing APIs

1. Creating Payment Request

This is a request sent by a merchant to the RMS system to create an invoice or payment request to a specific customer/user/buyer. The customer will receive an preformatted invoice email with payment link attached. He/she can just click to proceed to the payment page without any hassle.

Function: AddInvoice

Parameters in POST Variables

Name	Type	Mandatory	Description
<i>merchant_id</i>	varchar(32)	Yes	RMS merchant ID
<i>orderid</i>	varchar(32)	Yes	Order ID or Invoice number
<i>bill_name</i>	varchar(128)	Optional	Customer Name
<i>bill_email</i>	varchar(128)	Optional	Customer Email
<i>amount</i>	Double	Yes	Total amount to be paid
<i>remark</i>	Text	Yes	Remark for this invoice that will be displayed in payment email
<i>currency</i>	varchar(3)	Yes	Currency in the invoice, please use IDR as default
<i>email_subject</i>	varchar(128)	Yes	Email subject or the invoice title
<i>expire_date</i>	Date/Time YYYY-MM-DD HH:mm:ss	Optional	Empty indicates never expire
<i>MsgID</i>	varchar(32)	Yes	Merchant data integrity checksum or message ID
<i>mobile_number</i>	varchar(32)	Optional	Mobile number to send SMS or Whatsapp with country code E.g. +60123456789
<i>tcctype</i>	varchar(4)	Optional	Available value is: SALS = Capture Transaction (Default) AUTH = Authorize Transaction (Please inform support@fiuu.com before starting using pre-auth)

<i>send_type</i>	integer(1)	Optional	If empty, default value will be '1' to send SMS, if the value is 2, will send to Whatsapp
<i>block_paid</i>	integer(1)	Optional	By default, this will be set to 0 to allow receiving multiple payments using the same payment link. If you want to block receiving payments using the same link, set the value to 1.
<i>version</i>	integer(1)	Optional	By default, the value will be set to 1, which means the MD5 hashing algorithm will be used. If you want to use a different hashing algorithm that matches the one used for the vcode in payment request, you can pass the value 2.

*** Array of invoice detail (Size limitation: 0 - 5)**

Name	Type	Mandatory	Description
<i>item[0][descr]</i>	varchar(255)	Optional	Itemized descriptions, name of goods/service
<i>item[0][totalcost]</i>	Double	Optional	Itemized subtotal * will not sum up automatically * must match the number above

Results array

Key	Value	Detail
<i>invoice_id</i>	int(10)	Invoice ID if invoice is created successfully. Otherwise will be empty if error occurs.
<i>pymt_link</i>	Text	Payment link if invoice is created successfully. Otherwise will be empty if error occurs.
<i>sms_send_status</i>	Text	Indicator if the SMS successfully send or not.
<i>ws_send_status</i>	Text	Indicator if the Whatsapp successfully send or not
<i>error_code</i>	varchar(10)	Error code (if any)
<i>error_desc</i>	Text	Error message (if any)
<i>MsgID</i>	varchar(32)	Echo of MsgID

2. Status Query

This is a request initiated by a merchant to the RMS system to check on the payment status of a specific invoice. This function can be called manually (e.g. click a button) or scheduled task (e.g. cron job) within a reasonable interval. Merchants are only allowed to send 1 request every 5 - 15 minutes when necessary. Massive access activities will be blocked automatically without prior notice. We recommend large merchants to implement a “*CallBack*” status update function in order to get notified by the RMS system once there is any alteration on payment status.

Function: QueryStatus

Parameters in POST Variables

Name	Type	Mandatory	Description
<i>merchant_id</i>	varchar(32)	Yes	RMS merchant ID
<i>amount</i>	Double	Yes	Amount to be paid
<i>currency</i>	varchar(3)	Yes	Currency in the invoice, please use IDR as default
<i>invoice_id</i>	int(10)	Yes	Invoice ID or order ID
<i>MsgID</i>	varchar(32)	Yes	Merchant data integrity checksum or message ID

Result array

Key	Value	Detail
<i>invoice_id</i>	int(10)	Invoice ID or order ID
<i>sent</i>	“Y” or “N”	“Y” if invoice has been sent
<i>paid</i>	“TRUE” or “FALSE” or “PENDING”	TRUE = Success FALSE = Failed PENDING = Pending
<i>deleted</i>	“TRUE” or “FALSE”	TRUE = Payment link deleted FALSE = Payment link is available
<i>error_code</i>	varchar(10)	Error code (if any)
<i>error_desc</i>	Text	Error message (if any)
<i>MsgID</i>	varchar(32)	Echo of MsgID
<i>pymt_link</i>	Text	Payment link

3. Edit Invoice Details

This is a request sent by a merchant to the RMS system to edit the invoice details. For example, orderID, name, amount, currency, expiry date, email, email subject and mobile number. Details of invoice will be updated based on the value passed.

Function: EditInvoice

Parameters in POST Variables

Name	Type	Mandatory	Description
<i>merchant_id</i>	varchar(32)	Yes	RMS merchant ID
<i>amount</i>	Double	Optional	Amount to be paid
<i>currency</i>	varchar(3)	Optional	Currency in the invoice, please use IDR as default
<i>invoice_id</i>	int(10)	Yes	Invoice ID or order ID
<i>MsgID</i>	varchar(32)	Yes	Merchant data integrity checksum or message ID
<i>orderid</i>	varchar(32)	Optional	Order ID or Invoice number
<i>bill_name</i>	varchar(128)	Optional	Customer Name
<i>bill_email</i>	varchar(128)	Optional	Customer Email
<i>email_subject</i>	varchar(128)	Optional	Email subject or the invoice title
<i>expire_date</i>	Date/Time YYYY-MM-DD HH:mm:ss	Optional	Empty indicates never expire

Result array

Key	Value	Detail
<i>invoice_id</i>	int(10)	Invoice ID or order ID
<i>error_code</i>	varchar(10)	Error code (if any)
<i>error_desc</i>	Text	Error message (if any)
<i>MsgID</i>	varchar(32)	Echo of MsgID
<i>response</i>	Text	Response message

4. Delete Payment Link

This is a request sent by a merchant to the RMS system to delete the payment link of the invoice. The specific payment link will be unusable after deletion.

Function: DeletePaymentLink

Parameters in POST Variables

Name	Type	Mandatory	Description
<i>merchant_id</i>	varchar(32)	Yes	RMS merchant ID
<i>amount</i>	Double	Yes	Amount to be paid
<i>currency</i>	varchar(3)	Yes	Currency in the invoice, please use IDR as default
<i>invoice_id</i>	int(10)	Yes	Invoice ID or order ID
<i>MsgID</i>	varchar(32)	Yes	Merchant data integrity checksum or message ID

Result array

Key	Value	Detail
<i>invoice_id</i>	int(10)	Invoice ID or order ID
<i>error_code</i>	varchar(10)	Error code (if any)
<i>error_desc</i>	Text	Error message (if any)
<i>MsgID</i>	varchar(32)	Echo of MsgID
<i>response</i>	Text	Response message

Example of calling AddInvoice API using C#

```
public const SslProtocols _Tls12 = (SslProtocols)0x00000C00;
public const SecurityProtocolType Tls12 = (SecurityProtocolType)_Tls12;

public static void Main(string[] args)
{
    using (var client = new HttpClient())
    {
        string merchantID = "[MERCHANTID]";
        string vkey = "[VERIFYKEY]";
        string md5_vkey = GetMD5HashData(vkey);

        string parameters = "";
        string[] values = null;

        dynamic data = new JObject();
        data.merchant_id = "[MERCHANTID]";
        data.orderid = "demoINV01";
        data.bill_name = "demo";
        data.bill_email = "demo@rms.com";
        data.amount = "2.10";
        data.remark = "Product 1";
        data.currency = "MYR";
        data.email_subject = "Payment to RMS";
        data.MsgID = "534523424";
        data.mobile_number = "+60123456789";
        data.item = new JArray() as dynamic;

        dynamic items = new JObject();
        items.descr = "ITEM 1";
        items.totalcost = "1.10";
        data.item.Add(items);

        dynamic items2 = new JObject();
        items2.descr = "ITEM 2";
        items2.totalcost = "1.10";
        data.item.Add(items2);

        string data_parms = Convert.ToString(data);

        string checksum =
        GetMD5HashData(data_parms+GetSHA1HashData(md5_vkey));

        NameValueCollection parms = new NameValueCollection();
        parms.Add("params", data_parms);
        parms.Add("checksum", checksum);

        foreach (string key in parms.Keys)
        {
            values = parms.GetValues(key);
            foreach (string value in values)
            {
```

```

        parameters = parameters + key + "=" + value + "&";
    }
}

parameters = parameters.Remove(parameters.Length - 1);

ServicePointManager.SecurityProtocol = Tls12;
string url =
"https://api.fiuu.com/RMS/API/Invoice/index.php?op=AddInvoice";

WebClient wc = new WebClient();
wc.Headers[HttpRequestHeader.ContentType] =
"application/x-www-form-urlencoded";
string HtmlResult = wc.UploadString(url, parameters);

Console.WriteLine(HtmlResult);
}
}

```

Sample of AddInvoice API response

```

{
    "results": {
        "invoice_id": 19312,
        "pymt_link":
"https://pay.fiuu.com/RMS/qpay.php?k=c330419c69a951f4fed0e7ce225d38fc5235",
        "sms_send_status": "SEND",
        "error_code": "",
        "error_desc": "",
        "MsgID": "1976"
    },
    "checksum": "c92a10324374fac681719d63979d00fe",
    "datetime": "2018-07-25 10:47:29"
}

```

Example of calling QueryStatus API using C#

```
public const SslProtocols _Tls12 = (SslProtocols)0x00000C00;
public const SecurityProtocolType Tls12 = (SecurityProtocolType)_Tls12;

public static void Main(string[] args)
{
    using (var client = new HttpClient())
    {
        string merchantID = "[MERCHANTID]";
        string vkey = "[VERIFYKEY]";

        string md5_vkey = GetMD5HashData(vkey);

        string parameters = "";
        string[] values = null;

        dynamic data = new JObject();
        data.merchant_id = "[MERCHANTID]";
        data.invoice_id = "18330";
        data.amount = "2.10";
        data.currency = "MYR";
        data.MsgID = "534523424";

        string data_parms = Convert.ToString(data);

        //Console.WriteLine(data_parms);

        string checksum =
        GetMD5HashData(data_parms+GetSHA1HashData(md5_vkey));

        NameValueCollection parms = new NameValueCollection();
        parms.Add("params", data_parms);
        parms.Add("checksum", checksum);

        foreach (string key in parms.Keys)
        {
            values = parms.GetValues(key);
            foreach (string value in values)
            {
                parameters = parameters + key + "=" + value + "&";
            }
        }

        parameters = parameters.Remove(parameters.Length - 1);

        ServicePointManager.SecurityProtocol = Tls12;
        string url =
        "https://api.fiuu.com/RMS/API/Invoice/index.php?op=QueryStatus";

        WebClient wc = new WebClient();
        wc.Headers[HttpRequestHeader.ContentType] =
        "application/x-www-form-urlencoded";
```



```
        string HtmlResult = wc.UploadString(url, parameters);

        Console.WriteLine(HtmlResult);
    }
}
```

Sample of QueryStatus API response

```
{
    "results": {
        "invoice_id": "18330",
        "sent": "TRUE",
        "paid": "PENDING",
        "error_code": null,
        "error_desc": "",
        "MsgID": "534523424",
        "pymt_link":
        "https://www.onlinepayment.com.my/RMS/qpay.php?k=2739fb201388f450a449ce0f61f3739a8111"
    },
    "checksum": "c92a10324374fac681719d63979d00fe",
    "datetime": "2018-04-12 18:46:15"
}
```

Example of calling EditInvoice API using C#

```
public const SslProtocols _Tls12 = (SslProtocols)0x00000C00;
public const SecurityProtocolType Tls12 = (SecurityProtocolType)_Tls12;

public static void Main(string[] args)
{
    using (var client = new HttpClient())
    {
        string merchantID = "[MERCHANTID]";
        string vkey = "[VERIFYKEY]";

        string md5_vkey = GetMD5HashData(vkey);

        string parameters = "";
        string[] values = null;

        dynamic data = new JObject();
        data.merchant_id = "[MERCHANTID]";
        data.invoice_id = "19340";
        data.amount = "2.10";
        data.currency = "MYR";
        data.MsgID = "57344";
        data.orderid = "demoINV01";
        data.bill_name = "demo";
        data.bill_email = "demo@rms.com";
        data.email_subject = "Payment to RMS";
        data.mobile_number = "+60123456789";

        string data_parms = Convert.ToString(data);

        //Console.WriteLine(data_parms);

        string checksum =
        GetMD5HashData(data_parms+GetSHA1HashData(md5_vkey));

        NameValueCollection parms = new NameValueCollection();
        parms.Add("params", data_parms);
        parms.Add("checksum", checksum);

        foreach (string key in parms.Keys)
        {
            values = parms.GetValues(key);
            foreach (string value in values)
            {
                parameters = parameters + key + "=" + value + "&";
            }
        }

        parameters = parameters.Remove(parameters.Length - 1);

        ServicePointManager.SecurityProtocol = Tls12;
        string url =
```

```
"https://api.fiuu.com/RMS/API/Invoice/index.php?op=EditInvoice";

WebClient wc = new WebClient();
wc.Headers[HttpRequestHeader.ContentType] =
"application/x-www-form-urlencoded";
string HtmlResult = wc.UploadString(url, parameters);

Console.WriteLine(HtmlResult);
}
}
```

Sample of Edit Invoice API response

```
{
  "results": {
    "invoice_id": "19340",
    "error_code": "",
    "error_desc": "",
    "MsgID": "57344",
    "response": "Success edited Invoice"
  },
  "checksum": "bf424cb7b0dea050a42b9739eb261a3a",
  "datetime": "2022-04-13 15:22:38"
}
```

Example of calling DeletePaymentLink API using C#

```
public const SslProtocols _Tls12 = (SslProtocols)0x00000C00;
public const SecurityProtocolType Tls12 = (SecurityProtocolType)_Tls12;

public static void Main(string[] args)
{
    using (var client = new HttpClient())
    {
        string merchantID = "[MERCHANTID]";
        string vkey = "[VERIFYKEY]";

        string md5_vkey = GetMD5HashData(vkey);

        string parameters = "";
        string[] values = null;

        dynamic data = new JObject();
        data.merchant_id = "[MERCHANTID]";
        data.invoice_id = "19340";
        data.amount = "2.10";
        data.currency = "MYR";
        data.MsgID = "57344";

        string data_parms = Convert.ToString(data);

        //Console.WriteLine(data_parms);

        string checksum =
        GetMD5HashData(data_parms+GetSHA1HashData(md5_vkey));

        NameValueCollection parms = new NameValueCollection();
        parms.Add("params", data_parms);
        parms.Add("checksum", checksum);

        foreach (string key in parms.Keys)
        {
            values = parms.GetValues(key);
            foreach (string value in values)
            {
                parameters = parameters + key + "=" + value + "&";
            }
        }

        parameters = parameters.Remove(parameters.Length - 1);

        ServicePointManager.SecurityProtocol = Tls12;
        string url =
        "https://api.fiuu.com/RMS/API/Invoice/index.php?op=DeletePaymentLink";

        WebClient wc = new WebClient();
        wc.Headers[HttpRequestHeader.ContentType] =
        "application/x-www-form-urlencoded";
```

```
        string HtmlResult = wc.UploadString(url, parameters);

        Console.WriteLine(HtmlResult);
    }
}
```

Sample of Delete Payment Link API response

```
{
  "results": {
    "invoice_id": "19340",
    "error_code": "",
    "error_desc": "",
    "MsgID": "57344",
    "response": "Success deleted payment link"
  },
  "checksum": "bf424cb7b0dea050a42b9739eb261a3a",
  "datetime": "2022-04-13 15:25:46"
}
```

MD5 and SHA1 using C#

```
public static string GetMD5HashData(string data)
{
    MD5 md5 = System.Security.Cryptography.MD5.Create();
    byte[] inputBytes = System.Text.Encoding.ASCII.GetBytes(data);
    byte[] hash = md5.ComputeHash(inputBytes);

    StringBuilder sb = new StringBuilder();
    for (int i = 0; i < hash.Length; i++)
    {
        sb.Append(hash[i].ToString("x2"));
    }

    return sb.ToString();
}

public static string GetSHA1HashData(string data)
{
    SHA1 sha = new SHA1CryptoServiceProvider();
    byte[] temp;

    temp = sha.ComputeHash(Encoding.UTF8.GetBytes(data));

    StringBuilder sb = new StringBuilder();
    for (int i = 0; i < temp.Length; i++)
    {
        sb.Append(temp[i].ToString("x2"));
    }

    return sb.ToString();
}
```

Error Code & Message

Error Code	Error Message	Description
102	Invalid merchant ID	You must provide a valid and active RMS merchant ID
103	Empty orderid	Order ID cannot be empty
106	Empty amount	Valid amount is required
107	Empty remark	Remark cannot be empty
108	Empty currency	Currency is required
109	Empty mode	Mode is required
110	Empty inv_tpl	Template ID is required
111	Empty MsgID	Message ID is required
113	Empty email subject	Email subject cannot be empty
114	SQL save invoice problem	Please retry after a while
119	Invalid checksum	Please check the formula or raw data (avoid invisible character)

- End of Doc -