**Addis Ababa City Government Revenues Bureau**

**CUSTOR COMPUTING PLC, Tel 0115519397**

**Addis Ababa City Government Revenues Bureau.**

**API Documentation, Version 1.0**

Contents

[**Introduction** 1](#_Toc24536801)

[**Major Tasks** 2](#_Toc24536802)

[**Network Connection** 3](#_Toc24536803)

[**Deployment** 4](#_Toc24536804)

[**Authentication and Authorization** 5](#_Toc24536805)

[**Collecting Payment** 7](#_Toc24536806)

[**Voiding a Payment (Payment Reversal)** 15](#_Toc24536807)

[**Displaying Payment List** 17](#_Toc24536808)

# **Introduction**

This document documents the major web service functions that can be used to implement an integrated cash collection system.

Web Services provide a mechanism that allows different developers access the functionalities of a system without knowing the development platform as well as the database back end system. Web services are labelled as the “**Create Once, Consume Everywhere**” technology since they can be consumed by any programming language and are supported by desktops and mobile applications.

The major advantages of web services can be summarized as follows:

* Web services allow software and services from different companies and locations to be combined easily to provide an integrated service in a secured manner.
* Web services provide interoperability between various software applications running on disparate platforms/operating systems.
* Web services use open standards and protocols. Protocols and data formats are normally text-based, making it easy for developers to comprehend.
* Web services allow the reuse of services and components within an infrastructure. Web services are loosely coupled thereby facilitating a distributed approach to application integration.
* Services should support **secure communication**

# **Major Tasks**

*The major tasks required for fully implementing integrated payment management system are the following.*

|  |  |  |
| --- | --- | --- |
| ***No*** | ***Major Task*** | ***Activities*** |
|  | ***Create a secured VPN between financial centers and Addis Ababa city government revenues bureau*** | * *Create secured infrastructure on both parties.* * *Create Secured VPN connection on top of the secured infrastructure* |
|  | ***Handling payments/reconciliations*** | * *Search payment order information* * *Generate receipts* * *Prepare reconciliation reports* |

# **Network Connection**

For any Payment services, the city government revenues bureau eTax Payment system will link with any external system (i.e. different Financial institutes). eTax Payment system will use same APIs to connect with different 3rd financial institutes.

When a customer initiates a payment either on online or on windows service, the financial institutes will first identified by “Username”, “Password’. The authorized institution queries the payment information based on invoice (Order number) inputted for the customer. Finally the finical institute reserve funds from the customer account and then sends a conformation message to the eTax API. The transaction will only be successful when the third party conformation is passed, otherwise it will be cancelled.

# **Deployment**

The Web Service application is deployed at data center prepared by the Addis Ababa city government revenues bureau as shown below. 

# **Authentication and Authorization**

Each user accessing the system is identified by “Username”, “Password’ and “TaxCenterId”. The system also uses token based authorization technique where an authenticated user will be allowed to access resources for a particular period of time (session).

Users first login to a central service, which generates a token for a particular session. This token will be used to access other services without the need for supplying credentials for every service call. Client applications are required to save this value locally for each session and pass it with each API call in the Header part of the web service call.

The first step is to call the following API:

http://domainname/eTax/api/AuthenticateEx/

and then sending the **Username** and **Password** in the API header.

If the Username and Password is valid, then the system returns a “Token” that is used for accessing other API calls during a specified session. Otherwise, the API returns a **null** value.

Method: **POST**

Request Body:

|  |  |
| --- | --- |
| **Parameter** | **Description** |
| Username | The UserID of the user accessing the API |
| Password | The Password of the user accessing the API |

**Return Value**: The function returns a **UserProfile** object that contains the following parameters:

|  |  |
| --- | --- |
| **Parameter** | **Description** |
| FullName | The full name of the user whose UserID was passed on to the API call |
| Roles | The list of roles assigned for the user |
| PasswordExpiryRemainingDays | The number of days that remain before the user is notified to change his password. |
| MinNumbersLength | The minimum number of numbers that need to be contained in the password |
| LastPasswordChangeDate | The date at which the user changed his password |
| MinPasswordLength | Minimum Password Length |
| MinUpperCaseLength | Minimum number of Upper Case Characters in the password |
| MinSpecialCharactersLength | Minimum number of Special Characters in the password (! ,& etc) |
| MaxPasswordLength | Maximum Password Length |
| MobileNo | Mobile Number of the User |
| LocationCode | Location Code of the User |
| LocationName | Name of the User’s location |

**Error code**

|  |  |
| --- | --- |
| **Error Code** | **Message** |
| 401 | Unauthorized User. |

# **Collecting Payment**

The following steps need to be used to process the payment.

1. Search the existing record
   1. Search existing record by supplying the Payment Order Number (Invoice Number) that was generated while the service requested by the user is approved by the officer. Use the following API for this purpose.

***http://domainname/eTax/api/***[***PaymentEx***](http://localhost/MotiPay/swagger/ui/index#!/PaymentEx/PaymentEx_Put)***/GetPaymentDetailByInvoiceNoEx /{InvoiceNo}***

**Method**: GET

**Input**: InvoiceNo - the Payment Order Number generated when the payment order record was registered

**Return Value**: An object with the following attributes

|  |  |  |  |
| --- | --- | --- | --- |
| **Parameter** | **Description** | **Data Type** | **Length** |
| OrderID | An id that uniquely identifies each record | Integer |  |
| OrderedDate | The Date at which the payment order is prepare using UTC format | Date Time |  |
| TotalAmount | The Total amount to be paid for the particular service | Decimal |  |
| IsVoid | Tells weather the transaction is void or not | bit |  |
| IsPaid | Tells weather the transaction is Paid or not | bit |  |
| CustomerName | The business name of the customer | String | 80 |
| Tin | Tax payer identification number | String | 10 |
| ServiceTaken | Service description for receipt. | String | 120 |
| ErrorCode | Error code if any | Integer |  |
| AccountNumber | Account Number in which the money is deposited | String | 30 |
| ErrorMessage | Error message if any | String | 150 |

**Error codes**

|  |  |
| --- | --- |
| **Error Code** | **Message** |
| **403** | The Payment is already made by this payment order number |
| **101** | Invoice No. doesn't exist |
| 401 | Unauthorized User. |
| 502 | Error! Payment Order No longer vailed |
| 306 | There is no specific Account no to pay for this transaction |

* 1. Search existing record by supplying the Payment Order Number (Invoice Number) that was generated while the service requested by the user and request Id which is generated by ET-switch. Use the following API for this purpose.

***http://domainname/eTax/api/***[***PaymentEx***](http://localhost/MotiPay/swagger/ui/index#!/PaymentEx/PaymentEx_Put)***/GetPaymentDetailByInvoiceNoEx /{InvoiceNo}/{ETSRequestID}***

**Method**: GET

**Input**: ***InvoiceNo -*** the Payment Order Number generated when the payment order record was registered

***ETSRequestID*** - the Request Number generated when the payment order record was registered

**Return Value**: An object with the following attributes

|  |  |  |  |
| --- | --- | --- | --- |
| **Parameter** | **Description** | **Data Type** | **Length** |
| OrderID | An id that uniquely identifies each record | Integer |  |
| ETSRequestID | The Request Number generated by the 3rd party | Integer |  |
| OrderedDate | The Date at which the payment order is prepare using UTC format | Date Time |  |
| TotalAmount | The Total amount to be paid for the particular service | Decimal |  |
| IsVoid | Tells weather the transaction is void or not | bit |  |
| IsPaid | Tells weather the transaction is Paid or not | bit |  |
| CustomerName | The business name of the customer | String | 80 |
| Tin | Tax payer identification number | String | 10 |
| ServiceTaken | Service description for receipt. | String | 120 |
| ErrorCode | Error code if any | Integer |  |
| AccountNumber | Account Number in which the money is deposited | String | 30 |
| ErrorMessage | Error message if any | String | 150 |

**Error codes**

|  |  |
| --- | --- |
| **Error Code** | **Message** |
| **403** | The Payment is already made by this payment order number |
| **101** | Invoice No. doesn't exist |
| 401 | Unauthorized User. |
| 502 | Error! Payment Order No longer vailed |
| 306 | There is no specific Account no to pay for this transaction |

1. Making payment record using the following API
   1. Collect payment

[**http://domainname/*eTax/*api/paymentEx/{PaymentOrderNo}**](http://domainname/eTax/api/paymentEx/%7bPaymentOrderNo%7d)

Method: **PUT**

The API expects an object with the following parameters:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Datatype** | **Length** | **Is Mandatory** |
| PaymentOrderNo | The Payment Order Number obtained in the previous call | Integer |  |  |
| CashierName | The Name of the cashier (for Banks Bank code is enough) | string | 15 |  |
| ReceiptNo | A number that appears in a pre-printed receipt | String | 15 |  |
| CheckNo | The Check No, if payment is made by cheque | String | 15 |  |
| BankCode | The Financial center code that uniquely identifies it from other Financial centers. | String | 6 |  |
| BankName | The Name of the Financial center. | String | 6 |  |
| PaymentMethod | The payment method. | String | 12 |  |
| UserID | The User name of the Cashier (in case of banks they can pass the bank code as the username) | String | 15 |  |
| LocationCode | Optional filed used to identify the location of the tax payer (in case of banks they can pass 0 as default value) | int |  |  |

Sample Request Body:

{

  "PaymentOrderNo": 6,

  "CashierName": "CBE",

  "ReceiptNo": "0004654",

  "CheckNo": "",

  "BankCode": "CBE",

  "BankName": "Commercial Bank of Ethiopia",

  "PaymentMethod": "online",

  "UserID": "CBE",

  "LocationCode": 0

}

Note:

1. Payment Amount is internally saved by the officer at Addis Ababa city government revenues bureau.
2. Payment Date is saved as system date.

**Return Value**: IsSaved – True if operation is successful; false otherwise

* ErrorMessage – Error Reason, if any
* ErrorCode – 200 if successful

**Error response**

|  |  |
| --- | --- |
| **Error Code** | **Message** |
| 302 | Order number in path parameter is different form request body |
| 400 | Receipt Number cannot be empty |
| 403 | The Payment is already made by this Invoice number |
| 426 | Error! The Receipt Number has already been used |
| 101 | Error! Invoice No. doesn't exist |
| 500 | internal server error |
| 401 | Unauthorized User. |
| 502 | Payment no Longer valid |

* 1. This API is used to collect payment like the API above but returns Object containing receipt number instead of Boolean.

[**http://domainname/eTax/api/paymentET/{PaymentOrderNo}**](http://domainname/eTax/api/paymentET/%7bPaymentOrderNo%7d)

Method: **PUT**

The API expects an object with the following parameters:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Datatype** | **Length** | **Is Mandatory** |
| InvoceNo | The Payment Order Number obtained in the previous call | Integer |  |  |
| ReceiptNo | A number that appears in a pre-printed receipt | String | 15 |  |
| CheckNo | The Check No, if payment is made by cheque | String | 15 |  |
| BankCode | The Financial center code that uniquely identifies it from other Financial centers. | String | 6 |  |
| PaymentMethod | The payment method. | String | 12 |  |

Sample Request Body:

{

"InvoiceNo": 4546,

"ReceiptNo": "00345634",

"CheckNo": "",

"BankCode": "CBE",

"BankName": "CBE",

"PaymentMethod": "Online”

}

Note:

1. Payment Amount is internally saved by the officer at Addis Ababa city government revenues bureau.
2. Payment Date is saved as system date.

**Return Value**:

A Result object with the following parameters:

* IsSaved – True if operation is successful; false otherwise
* ReceiptNo- Receipt No in which the payment is done
* AccountNo-Account Number in which the money is deposited ( when isSaved-true)
* ErrorMessage – Error Reason, if any
* ErrorCode – 200 if successful

**Error response**

|  |  |
| --- | --- |
| **Error Code** | **Message** |
| 302 | Order number in path parameter is different form request body |
| 400 | Receipt Number cannot be empty |
| 403 | The Payment is already made by this invoice number |
| 426 | Error! The Receipt Number has already been used |
| 101 | Error! Invoice No. doesn't exist |
| 500 | internal server error |
| 401 | Unauthorized User. |
| 502 | Payment no Longer valid |

# **Voiding a Payment (Payment Reversal)**

To tag a payment record as “Void” the following api call is used:

1. [**http://domainname/eTax/api/VoidPayment/VoidByReceiptNo**](http://domainname/eTax/api/VoidPayment/VoidByReceiptNo)**/{ReceiptNo}**

Method: PUT

Request Body: **ReceiptNo** is Mandatory.

Return Value: Result object with the following fields:

**IsSaved**: Boolean

**ErrorCode: Int**

**ErrorMessage**: Message returned if there is error in the operation, empty string otherwise.

|  |  |
| --- | --- |
| **Error Code** | **Message** |
| 401 | Unauthorized User. |
| 402 | User has no right to access this resource |
| 101 | Receipt No. doesn't exist |
| 303 | Sorry, Only paid transactions can be voided |
| 503 | Sorry, You are not allowed to void transaction prepared by another Organization |
| 504 | Sorry, The transaction is already void |
| 500 | Internal server error |

1. [**http://domainname**/**eTax/api/VoidPayment/VoidPaymentByPaymentOrderNo**](http://domainname/eTax/api/VoidPayment/VoidPaymentByPaymentOrderNo)**/{InvoiceNo}**

Method: PUT

Request Body: The API expects **InvoiceNo** as request body.

Return Value: Result object with the following fields:

**IsSaved**: Boolean

**ErrorCode: Int**

**ErrorMessage**: Message returned if there is error in the operation, empty string otherwise.

|  |  |
| --- | --- |
| **Error Code** | **Message** |
| 401 | Unauthorized User. |
| 402 | User has no right to access this resource |
| 101 | Invoice No. doesn't exist |
| 303 | Sorry, Only paid transactions can be voided |
| 503 | Sorry, You are not allowed to void transaction prepared by another Organization |
| 504 | Sorry, The transaction is already void |
| 500 | Internal server error |

# 

# **Displaying Payment List**

To display list of payment records registered during a given period of time, the following **api** is used.

**http://domainname/eTax/api/paymentlist/{DateFrom}/{DateTo}/{IsVoid}/{UserID}**

**Method: GET**

**Input Parameters**:

|  |  |
| --- | --- |
| **Parameter** | **Description** |
| **DateFrom** | Date in UTC form indicting the beginning of the payment date range |
| **DateTo** | Date UTC form indicting the end of the payment date range |
| **IsVoid** | If this parameter is true, then the list will show records that are tagged “Void” only. Otherwise, the list will show records that are flag paid excluding void payments. |
| **UserID** | User ID of the user that created the payment record |

**Return Value**: List object in JSON format.

[

{

"PaymentOrderNo": 0,

"CashierName": "string",

"ReceiptNo": "string",

"CustomerName": "string",

"Tin": "string",

"PaidAmount": 0,

"PaymentDate": "2019-11-12T14:14:07.971Z",

"ChequeNo": "string"

}

]

|  |  |
| --- | --- |
| **Return Field** | **Description** |
| PaymentOrderNo | The Payment Order Number that identifies each payment record |
| CashierName | The name of the cashier that collected the payment |
| ReceiptNo | The number that appears on the electronic receipt |
| CustomerName | The name of the Customer |
| Tin | The Tax payer identification number of the customer |
| PaidAmount | The amount paid by the customer |
| PaymentDate | The date at which the payment was settled |
| ChequeNo | The Cheque Number, if payment is settled by Checque |