NIFTYHUB REST API DOCUMENTATION

1. **Register User**

* **Description**:

Register or add a user in the system by providing related information.

* **Url**:

/users/register

* **Method**:

POST

* **Url Params**:

NONE

* **Data Params**:

{"body" { "users\_username": value, "users\_first\_name": value, "users\_last\_name": value,

"users\_gender": value, "user\_country": value, "user\_city": value, "ser\_state": value,

"user\_zipcode": value, "billing\_address": value, "user\_email": value, "user\_phone": value,

"users\_dob": value, "users\_about": value, "users\_type": value, "users\_status": value, "have\_shop": value, "users\_password": value }}

* **Required Fields**:

users\_username, users\_first\_name and useremail

* **Success Response**:

{ "message": "User registered successfully.", "emailSent": true, "userId": 332 }

* **Error Responses**:

{"err”: {

"success": false,

"message": "users\_first\_name is not string"

} }

OR

{"err”: { "success": false, "message": "User already exist with username: a0zb1q1za1" } }

1. **Get User**

* **Description**:

Return a user object by passing userId as parameter.

* **Url**:

/users/userId

* **Method**:

GET

* **Url Params**:

UserId

* **Data Params**:

{"body" { } }

* **Required Fields**:

userId as parameter

* **Success Response**:

{

"success": true,

"message": "User found with id: 341",

"user": [

{

"users\_username": "a0zb1q111z1a",

"users\_first\_name": "1",

"users\_last\_name": "tasdfest",

"users\_gender": "tesrt",

"user\_country": "asdasdf",

"user\_city": "tasdfesrt",

"ser\_state": "tesrt",

"user\_zipcode": "123",

"billing\_address": "asdf",

"user\_email": "aaqilcs102@gmail.com",

"user\_phone": "03412323987",

"users\_dob": "0000-00-00",

"users\_about": "asdf",

"users\_type": "asdf",

"users\_last\_login\_ip": null,

"users\_joined\_date": "2018-03-22T19:00:00.000Z",

"users\_last\_login\_datetime": null,

"users\_status": 0,

"have\_shop": 1

}

]

}

* **Error Responses**:

{"err”: {

"success": false,

"message": "User not found with the userId: 205"

}}

1. **Update User Status**

* **Description**:

Update user status from active to inactive or inactive to active

* **Url**:

/users/updateuserstatus/231

* **Method**:

PATCH

* **Url Params**:

userId

* **Data Params**:

{“body”: { "status": "inactive" }}

**Required**

“status” is required and should be ‘active’ or ‘inactive’

* **Success Response**:

{

"success": true,

"message": "Status updated successfully"

}

* **Error Response**:

{"err”: {

"success": false,

"message": "Not able to update user status"

}

1. **Upload User Image**

* **Description**:

Upload user image to the server

* **Url**:

users/uploaduseriamge/185

* **Method**:

POST

* **Url Params**:

userId

* **Data Params**:

{“body”: {“avatar”: file,}}

**Required**

“avatar” (sending image file) is required parameter and its value is an image type

* **Success Response**:

{

"success": true,

"message": "Image uploaded successfully"

}

* **Error Response**:

{"err”: {

"message": "Please provide the image to be uploaded"

}}

OR

Only image type allowed

1. **Resend email verification**

* **Description**:

If user wants to resend email verification to verify his or her email address

* **Url**:

users/resendverificationemail

* **Method**:

POST

* **Url Params**:

NONE

* **Data Params**:

{“body”: {“ user\_email”: value,}}

**Required**

“user\_email” is required and its value is an is an email address that is to be verified

* **Success Response**:

{

"message": "Email already verified!"

}

OR:

{

"message": "Verification email sent again.",

"emailSent": true

}

* **Error Response**:

{"err {"success":false,"message":"user\_email is required"}}

OR

{"success":false,"message":"Email not valid"}

1. **Forgot Password**

* **Description**:

If user has forgotten password, to reset password

* **Url**:

users/forgotpassword

* **Method**:

POST

* **Url Params**:

NONE

* **Data Params**:

{“body”: { "useremail": value}}

**Required**

“useremail” is required and its value is an is an email address of the user forgotten password

* **Success Response**:

{

"message": "email sent for password reset",

"emailSent": true,

"userId": 322

}

OR:

{

"success": true,

"message": "Email already sent, please wait for 10 mints for resend"

}

* **Error Response**:

{"err { "success": false, "message": "Please provide the useremail"}}

Or

{ "success": false, "message": "User not found" }

OR

{"success":false,"message":"Email not valid"}

1. **User Login**

* **Description**:

API endpoint for user login

* **Url**:

users/login

* **Method**:

POST

* **Url Params**:

NONE

* **Data Params**:

{“body”: { "username": "aaqilcs102@gmail.com", "password": "asA1!dasdff" }}

**Required**

“username” is required and its value is username of the user trying to login

“password” is required and its value is password of the user trying to login

* **Success Response**:

{

"success": true,

"user": [

{

"users\_id": 343,

"users\_username": "a0zb1q",

"users\_first\_name": "1",

"users\_last\_name": "tasdfest",

"users\_gender": "tesrt",

"user\_country": "asdasdf",

"user\_city": "tasdfesrt",

"ser\_state": "tesrt",

"user\_zipcode": "123",

"billing\_address": "asdf",

"user\_email": "aaqilcs102@gmail.com",

"user\_phone": "03412323987",

"users\_dob": "0000-00-00",

"users\_about": "asdf",

"users\_type": "asdf",

"users\_last\_login\_ip": null,

"users\_joined\_date": "2018-03-22T19:00:00.000Z",

"users\_last\_login\_datetime": "0000-00-00 00:00:00",

"users\_status": 0,

"have\_shop": 1

}

],

"message": "Token generated successfully",

"token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VyIjoiYTB6YjFxIiwiaWF0IjoxNTIxODcyMDc1fQ.aXLlOha74X-Rtc0ueqakV8VxeHHu14ZHkLQ-BH14rnk"

}

* **Error Response**:

{"err {

"success": false,

"message": "Invalid password"

} }

1. **Reset password**

* **Description**:

API endpoint for user to update password through account settings.

* **Url**:

users/resetpassword/userId

* **Method**:

POST

* **Url Params**:

userId

* **Data Params**:

{"body" : { "updatedPassword": "fals!11AAAe",

"token": "5bd4838127f7a45fc95b5e4440edb70fe46ce67"}}

* **Success Response**:

{

"success": true,

"message": "Password updated successfully."

}

* **Error Responses**:

{"err”: {

"success": false,

"message": "Please provide the token for update password."

}}

1. **Update Order Delivery Status:**

* **Description**:

API endpoint to update status of delivery of a particular order.

* **Url**:

/order/updatedeliverystatus/orderId

* **Method**:

PATCH

* **Url Params**:

orderId

* **Data Params**:

{"body": {"delivery\_status": "complete","token": token} }

* **Required Fields**:

delivery, token,

* **Success Response**:

{"success":true,"message":"Order delivery status in Order Details update successfully"}

* **Error Responses**:

{"err”: {"success":false,"message":"delivery status invalid, it should be 'pending','in process','shipped','delivered'"}}

OR

{"err”: {"success":false,"message":"order not found with orderId: 31"}}

1. **Add Admin:**

* **Description**:

API endpoint to add an admin, will work only for once.

* **Url**:

/admin/register

* **Method**:

POST

* **Url Params**:

NONE

* **Data Params**:

{"body": { "users\_username": "a0z1z1n", "users\_first\_name": "1", "users\_last\_name": value,

"users\_gender": "tesrt", "user\_country": "asdasdf", "user\_city": "tasdfesrt", "ser\_state": "tesrt",

"user\_zipcode": "123", "billing\_address": "asdf", "user\_email": "aaqil1c1s102@gmail.com", "user\_phone": value, "users\_dob": "22/03/2015", "users\_about": "asdf", "users\_password": "asA1!dasdff" }}

* **Required Fields**:

users\_username, user\_email, users\_password

* **Success Response**:

{"success":true,"message":"Admin added successfully", adminId: 1}

* **Error Responses**:

{"err {"success":false,"message":"Admin already exists"} }

1. **Update Admin:**

* **Description**:

API endpoint to update commission rate.

* **Url**:

/admin/addcommission

* **Method**:

POST

* **Url Params**:

NONE

* **Data Params**:

{"body": { "flat\_comission\_rate": 12, "token": token }}

* **Required Fields**:

flat\_comission\_rate, token,

* **Success Response**:

{"success":true,"message":"commission added successfully", commissionId: 1}

* **Error Responses**:

{"err {"success":false,"message":"access denied"} }

{"err {"success":false,"message":"flat\_comission\_rate is required"} }`

1. **Add Commission Rate:**

* **Description**:

API endpoint to add commission rate.

* **Url**:

/admin/addcommission

* **Method**:

POST

* **Url Params**:

NONE

* **Data Params**:

{"body": { "flat\_comission\_rate": 12, "token": token }}

* **Required Fields**:

flat\_comission\_rate, token,

* **Success Response**:

{"success":true,"message":"commission added successfully", commissionId: 1}

* **Error Responses**:

{"err {"success":false,"message":"access denied"} }

{"err {"success":false,"message":"flat\_comission\_rate is required"} }`

1. **Add Commission Record:**

* **Description**:

API endpoint to add a commission record in the system.

* **Url**:

/vendor/addcommissionrecord/vendorId

* **Method**:

POST

* **Url Params**:

vendorId

* **Data Params**:

{"body": { "date": "23/03/2018", "invoice\_number": 1, "fk\_tbl\_order\_order\_number": 23,

"amount": 1,"commission\_deducted": 1, "token": token } }

* **Required Fields**:

“date” is required of type with format ‘year/month/day’

"invoice\_number": is required of type number ,

"fk\_tbl\_order\_order\_number": is required of type number, it’s a foreign key of Order table,

"amount": is required of type number ,

"commission\_deducted": is required of type number,

"token": is required and is jwt token could be sent through body, parameter or headers

* **Success Response**:

{"success":true,"message":"Commission Record added successfully","commissionRecordId":2}

* **Error Responses**:

{"err: {"success":false,"message":"Failed to authenticate token."} }

{"err {"success":false,"message":"no order found with Id: 231111"} }`

1. **Get a Commission Record:**

* **Description**:

API endpoint to get a commission record of a particular customer.

* **Url**:

/vendor/commissionrecords/vendorId

* **Method**:

GET

* **Url Params**:

vendorId

* **Data Params**:

{"body": }

* **Required Fields**:
* **Success Response**:

{"success":true,"message":"Commission record(s) found ","commissionRecords":[{"commission\_record\_id":2,"date":"0000-00-00","invoice\_number":1,"fk\_vendor\_id":400,"fk\_tbl\_order\_order\_number":23,"amount":1,"commission\_deducted":1},{"commission\_record\_id":3,"date":"0000-00-00","invoice\_number":1,"fk\_vendor\_id":400,"fk\_tbl\_order\_order\_number":23,"amount":1,"commission\_deducted":1},{"commission\_record\_id":4,"date":"0000-00-00","invoice\_number":1,"fk\_vendor\_id":400,"fk\_tbl\_order\_order\_number":23,"amount":1,"commission\_deducted":1},{"commission\_record\_id":5,"date":"0000-00-00","invoice\_number":1,"fk\_vendor\_id":400,"fk\_tbl\_order\_order\_number":23,"amount":1,"commission\_deducted":1}]}

* **Error Responses**:

{"err: {"success":false,"message":"Failed to authenticate token."} }

{"err: {"success":false,"message":"No Commission record(s) found with vendorId: 4001"} }`

1. **ADD PRODUCT**

* **Description**:

Add a particular product against a shop.

* **Url**:

product/add

* **Method**:

POST

* **Url Params**:

NONE

* **Data Params**:

{"body" : { "product\_title": value, "product\_code": value, "product\_description": value, "fk\_tbl\_product\_cetegory\_id": value, "product\_price": value, "product\_stock": value, "fk\_tbl\_reseller\_shop\_id": value, "width": value, "depth": value, "height": value, "fk\_tbl\_reseller\_shipment\_mode": value } }

* **Success Response**:

{

"success": true,

"message": "Product added successfully",

"productId": 5

}

* **Error Responses**:

{"err”: {

"success": false,

"message": "Shipment mode does not exists"

}}

OR

{"err”: {

"success": false,

"message": "Reseller shop does not exists"

}}

OR

{"err”: {

"success": false,

"message": "Product category does not exists"

}}

1. **Upload Product Image**

* **Description**:

Upload product image to the server

* **Url**:

product/uploadImage/productId

* **Method**:

POST

* **Url Params**:

productId

* **Data Params**:

{“body”: {“avatar”: file, isDefault”: value}}

**Required**

“avatar” (sending image file) is required parameter and its value is an image type

* **Success Response**:

{

"success": true,

"message": "Image uploaded successfully"

}

* **Error Response**:

{"err”: {

"success": false,

"message": "product does not exist"

}}

OR

{

"success": false,

"message": "Please provide the image"

}

OR

Only image type allowed

1. **Update Product**

* **Description**:

Update a product in the database and return success or failure response

* **Url**:

product/updateproduct/productId

* **Method**:

PUT

* **Url Params**:

ProductId

* **Data Params**:

{“payload”: {

"product\_title": value,

"product\_code": value",

"product\_description": value,

"fk\_tbl\_product\_cetegory\_id": value,

"product\_price": value,

"product\_stock": value,

"width": value,

"depth": value,

"height": value

}}

**Required**

* **Success Response**:

{"success":true,"message":"Product updated successfully"}

* **Error Response**:

{"err”: {"success":false,"message":"Product not found with id: 31"}}

OR

{"err”: {"success":false,"message":"Product category does not exist"}}

1. **Add Product Category**

* **Description**:

Add a product category

* **Url**:

/product/addproductcategory

* **Method**:

POST

* **Url Params**:

NONE

* **Data Params**:

{"payload": {

"slug": value,

"parent\_category\_id": value,

"category\_title": value,

"catgory\_description": value,

"category\_image": value,

"status": value

}}

**Required**

**Optional**

* **Success Response**:

{

"success": true,

"message": "Product category added successfully"

}

* **Error Response**:

{ "err”: { "success": false, "message": "body cannot be empty" }}

1. **Add Shop**

* **Description**:

Add a shop against a user

* **Url**:

/shop/add/userId

* **Method**:

POST

* **Url Params**:

userId

* **Data Params**:

{“payload”: { "shop\_name": value, "short\_description": value, "shop\_address": value,

"shop\_zipcode": value , "business\_hours": value, "business\_days": value, "about\_shop": value,

"lattitude": value, "longitude": value

}}

**Required**

userId is required as parameter to add a shop

* **Success Response**:

{ "success": true, "message": "shop added successfully"}

* **Error Response**:

{"err”: { "success": false, "message": "User not found" }}

1. **Update Product Status:**

* **Description**:

Update status of a product

* **Url**:

product/updateproduct/productId

* **Method**:

Patch

* **Url Params**:

productId

* **Data Params**:

{"status": value}

**Required**

status is a required and its value can be as “active” or “inactive”.

* **Success Response**:

{ "success": true, "message": "Product status updated successfully"}

* **Error Response**:

{"err”: { "success": false, "message": "body cannot be empty" }}

OR

{ "success": false, "message": "Product not found with id: 12" }

1. **Add Order**

* **Description**:

Add an order against a particular shop

* **Url**:

order/add/userId

* **Method**:

POST

* **Url Params**:

userId

* **Data Params**:

{"payload”: { "fk\_tbl\_user\_shipment\_address": value, "invoice\_amount": value, "shipment\_amount": value, "customer\_ip": value} }

**Required**

* **Success Response**:

{ "success": true, "message": "order added successfully", "orderId": 16 }

* **Error Response**:

{"err”: { "success": false, "message": "Shipment address not found"} }

OR

{ "success": false, "message": "User not found" }

1. **Add Order Details:**

* **Description**:

API endpoint to add order details

* **Url**:

/order/addorderdetails/orderId

* **Method**:

POST

* **Url Params**:

orderId

* **Data Params**:

{"payload" : { "fk\_product\_id": value, "fk\_reseller\_shop\_id": value,"product\_price": value, "product\_quantity": value, "fk\_tbl\_product\_shipment\_mode": value, "product\_shipment\_charges": value, "product\_design": value, "product\_color": value,

"product\_width": value, "product\_depth": value, "product\_height": value, "product\_delivery\_status": value, "product\_delivery\_date": "value"

} }

* **Success Response**:

{ "success": true, "message": "Order details added successfully", "orderDetailsId": 4 }

* **Error Response**:

{"err”: { "success": false, "message": "Product not found"} }

OR

{ "success": false, "message": "Order not found"}

Or

{

"success": false,

"message": "Shipment Mode not found"

}

1. **Add Product Review:**

* **Description**:

API endpoint to add a product review.

* **Url**:

/review/productId

* **Method**:

POST

* **Url Params**:

productId

* **Data Params**:

{"body": { "fk\_tbl\_user\_id": 417, "product\_rating": 1, "review\_title": "1","review\_description": "1",

"review\_name":"1" }}

* **Required Fields**:

fk\_tbl\_user\_id: is a required field as a number, it is a foreign key of a user,

product\_rating: is a required field as a number, it is the rating of a product from 1 to 5,

review\_title: is a required field as a string, it is the title of a review

review\_description: is a required field as a string, it is the descript of a review

review\_name: is a required field as a string, it is the name of a review

----------

status is set true (1) by default

* **Success Response**:

{"success":true,"message":"Review Added successfully.","reviewId":3}

* **Error Responses**:

{"err {"success":false,"message":"product\_rating is required."} }

{"err {"success":false,"message":"user not found with userId: 411117"} }`

1. **Get Product Review:**

* **Description**:

API endpoint to get a product review.

* **Url**:

/review/productId

* **Method**:

GET

* **Url Params**:

productId

* **Data Params**: NONE
* **Required Fields**: NONE
* **Success Response**:

{"success":true,"message":"Review found","review":[{"review\_id":7,"fk\_tbl\_product\_id":16,"fk\_tbl\_user\_id":417,"product\_rating":"5","review\_title":"1","review\_description":"1","review\_name":"1","review\_date":"2018-03-26T19:00:00.000Z","status":1}]}

* **Error Responses**:

{"err {"success":false,"message":"Review not found with reviewId: 71"} }

1. **Get All Reviews of a Product:**

* **Description**:

API endpoint to get all reviews of a single product.

* **Url**:

/review/all/productId

* **Method**:

GET

* **Url Params**:

productId

* **Data Params**: NONE
* **Required Fields**: NONE
* **Success Response**:

{"success":true,"message":"Reviews found","review":[{"review\_id":2,"fk\_tbl\_product\_id":16,"fk\_tbl\_user\_id":417,"product\_rating":"1","review\_title":"1","review\_description":"1","review\_name":"1","review\_date":"2018-03-26T19:00:00.000Z","status":0},{"review\_id":3,"fk\_tbl\_product\_id":16,"fk\_tbl\_user\_id":417,"product\_rating":"5","review\_title":"1","review\_description":"1","review\_name":"1","review\_date":"2018-03-26T19:00:00.000Z","status":0]}

* **Error Responses**:

{"err {"success":false,"message":"Review not found for productId: 161"} }

1. **Get All Reviews of a Product:**

* **Description**:

API endpoint to get all reviews of a single product.

* **Url**:

/review/update/reviewId

* **Method**:

PUT

* **Url Params**:

reviewId

* **Data Params**:

{ "product\_rating": 1,

"review\_title": "1",

"review\_description": "1",

"review\_name":"1" }

* **Required Fields**:
  + product\_rating is required field of type number, it must be in the range of 1 to 5
  + review\_title is required field of type string
  + review\_description is required field of type string
  + review\_name is required field of type string
* **Success Response**:

{

"success": true,

"message": "review updated successfully"

}

* **Error Responses**:

{ "success": false, "message": "product\_rating must be 1 to 5"}

1. **Get All Reviews of a Product:**

* **Description**:

API endpoint to get all reviews of a single product.

* **Url**:

/review/update/reviewId

* **Method**:

PUT

* **Url Params**:

reviewId

* **Data Params**:

{ "product\_rating": 1,

"review\_title": "1",

"review\_description": "1",

"review\_name":"1" }

* **Required Fields**:
  + product\_rating is required field of type number, it must be in the range of 1 to 5
  + review\_title is required field of type string
  + review\_description is required field of type string
  + review\_name is required field of type string
* **Success Response**:

{

"success": true,

"message": "review updated successfully"

}

* **Error Responses**:

{ "success": false, "message": "product\_rating must be 1 to 5"}

1. **Update shop preferences**

* **Description**:

API endpoint to Update or change shop preferences

* **Url**:

/shop/updateshoppreference/userId

* **Method**:

PUT

* **Url Params**:

userId

* **Data Params**:

{“body”: { "shop\_address": "asdf", "shop\_zipcode": "asdf" , "lattitude": "1", "longitude": "1" }}

**Required**

“shop\_address” is required field of type string

“shop\_zipcode” is required field of type string [ZIP code can be string, for example Canada has zip codes with characters]

“latitude": is required field as latitude of type number

“longitude” is a required field as longitude of type number

* **Success Response**:

{ "response": true, "message": "shop preferences updated successfully" }

* **Error Response**:

{"err”: { {"response":false,"message":"longitude must must be a number"} }}

1. **Update Email address:**

* **Description**:

API endpoint to Update user email address

* **Url**:

/users/updateemail/userId

* **Method**:

PATCH

* **Url Params**:

userId

* **Data Params**:

{“body”: { "userEmail": "aa1qilcs102@gmail.com", "newEmail": "test@gmail.com"}}

* **Required**

“userEmail” is required field as a valid email address

“newEmail” is required field as a valid email address

* **Success Response**:

{"response":true,"message":"email address updated successfully"}

* **Error Response**:

{"response":false,"message":"Please provide the new email address"}

{"response":false,"message":"user not found with email address: aa1qilcs102@gmailasdasd.com"}

1. **Adding Payment Method:**

* **Description**:

API endpoint to add a payment method against a shop

* **Url**:

/shop/paymentmethod/shopId

* **Method**:

POST

* **Url Params**:

shopId

* **Data Params**:

{“body”: {"payment\_method\_name": "paypal"}}

* **Required**

“payment\_method\_name” is required field as a string

* **Success Response**:

{"response":true,"message":"Payment Method added successfully. ","paymentMethodId":8}

* **Error Response**:

{"response":false,"message":"Payment method is not valid."}

1. **Get All Payment Methods of a Shop:**

* **Description**:

API endpoint to get a list of payment methods that a shop have

* **Url**:

/shop/paymentmethod/shopId

* **Method**:

GET

* **Url Params**:

shopId

* **Data Params**:

NONE

* **Required:**

NONE

* **Success Response**:

{ "response": true, "message": "found payment method(s) for shopId: 54",

"paymentMethods": [

{ "payment\_method\_id": 1, "payment\_method\_name": "paypal" },

{ "payment\_method\_id": 2, "payment\_method\_name": "paypal" }

] }

* **Error Response**:

{"response":false,"message":"Payment method is not valid."}

1. **Remove a Payment Method:**

* **Description**:

API endpoint to remove payment method by providing it’s id

* **Url**:

/shop/paymentmethod/paymentMethodId

* **Method**:

DELETE

* **Url Params**:

paymentMethodId

* **Data Params**:

NONE

* **Required:**

NONE

* **Success Response**:

{ "response": true, "message": "Payment method removed with Id: 4"}

* **Error Response**:

{ "response": false, "message": "no payment method found for id: 4" }

1. **Get all invoices:**

* **Description**:

API endpoint to get all the invoices of a particular vendor

* **Url**:

/invoice/all/shopId

* **Method**:

GET

* **Url Params**:

shopId

* **Data Params**:

NONE

* **Required:**

NONE

* **Success Response**:

{

"response": true,

"message": "got order details: ",

"invoices": [

{

"orderId": 27,

"date": "2018-04-04T10:50:42.000Z",

"invoiceNumber": "5",

"amount": 100,

"customerId": 417,

"customerName": "Muhammad Aaqil",

"dueDate": "2018-04-10T19:00:00.000Z",

"orderStatus": "complete"

},

{

"orderId": 28,

"date": "2018-04-04T10:58:34.000Z",

"invoiceNumber": "6",

"amount": 11111,

"customerId": 417,

"customerName": "Muhammad Aaqil",

"dueDate": "2018-04-29T19:00:00.000Z",

"orderStatus": "complete"

}

]

}

* **Error Response**:

{ "response": false, "message": "No order found with shopId: 541"}

1. **Get a single invoice record:**

* **Description**:

API endpoint to get a single invoice record detail by providing invoice number

* **Url**:

localhost:3000/invoice/single/invoiceNumber

* **Method**:

GET

* **Url Params**:

invoiceNumber

* **Data Params**:

NONE

* **Required:**

NONE

* **Success Response**:

{

"response": true,

"message": "Invoice record: ",

"invoiceRecord": {

"orderId": 27,

"customerId": 417,

"invoiceNumber": "5",

"shipmentAddressId": 12,

"orderDate": "2018-04-04T10:50:42.000Z",

"orderStatus": "complete",

"items": [

{

"productId": 91,

"cost": 12,

"quantity": 2,

"title": "product\_title",

"description": "product\_description"

},

{

"productId": 92,

"cost": 121,

"quantity": 2,

"title": "product\_title",

"description": "product\_description"

}

],

"shippingAddress": {

"fk\_tbl\_users\_users\_id": 404,

"user\_address": "asdf",

"user\_country": "asdf",

"user\_city": "asdf",

"user\_zipcode": "asdf",

"status": 1

},

"billingDetails": {

"userCountry": "asdasdf",

"userCity": "tasdfesrt",

"userState": "tesrt",

"userZipcode": "123",

"userBillingAddress": "asdf"

} }}

* **Headers**:

x-access-token: token

* **Error Response**:

{ "response": false, "message": "No order found with invoice number: 51" }

**GET CUSTOMER BY USERID**

* **Description**:

Returns the customer info linked to the registered user.

* **Url**:

/orm/usermanagement/getcustbyuserid

* **Method**:

POST

* **Url Params**:

NONE

* **Data Params**:

{"payload”: {"userId”: value}}

**Required**

“userId” is a required parameter and its value is number e.g. 1326

* **Success Response**:

{"err”: null, "payload”: {"result”: [{"customerId":1348907,"firstName":"Amer","MIDDLENAME”: null,

"lastName":"Ahmed","FULLNAME":"ALI AHMED","tel":null,"mobile":"9652927226","email":null,"address":null,"postCode":null,"buildingNo":null,"city":null,"state":null,"nationalityId":null,"placeOfBirth":null,"dateOfBirth":null,"occupationId":null,"sourceOfFundId":null,"isActive":"1","countryId":null,"customerTypeId":null,"companyRegistrationNumber":null,"nationalId":null,"recipients":[{"receiverId": 1348964, "senderId": 1348963, "receiverName": "receiver one", "receiverCity": null, "receiverMobile": "252617335859"},{"receiverId": 1348964, "senderId": 1348963, "receiverName": "receiver two", "receiverCity": null, "receiverMobile": "252617335877"}]}]}}

* **Error Response**:

{"err”: ["Fail to find request data"],"payload”: {"result”: null}}

**GET USER TOKEN(DEPRECATED)**

* **Url**:

/orm/usermanagement/getusertoken

**ADD REMITTANCE**

* **Description**:

API is used to add a new remittance

* **Url**:

/orm/remittancemanagement/addremittance

* **Method**:

POST

* **Url Params**:

NONE

* **Data Params**:

{"payload":{"guid":"6fc64d0a-54cb-ae76-512f-c8e083e77770","senderIsSanctioned":false,"senderTypeId":null,"senderId":1348907,"senderFirstName":"Ahmed","senderMiddleName":null,"senderLastName":"Ali","senderMobile":"9652927226","senderTel":null,"senderCompanyRegistrationNumber":null,"senderAddress":null,"senderPostCode":null,"senderBuildingNumber":null,"senderState":null,"senderCity":null,"senderCountryId":null,"senderDateOfBirth":null,"senderPlaceOfBirth":null,"senderNationalityId":null,"senderNationalId":null,"senderOccupationId":null,"senderSourceOfFundId":null,"senderConductorId":null,"senderRelationshipId":null,"senderRelationship":null,"senderConductorName":null,"senderDocuments":[],"remittances":[{"receiverEMoneyAccountVerified":false,"receiverIsSanctioned":false,"remittanceTypeId":1,"pBranchId":null,"payingBranchCode":null,"pBranchCountryId":129,"pBranchCountry":"Kenya","pBranchCountryIso2":"KE","pBranchAddress":null,"pBranchCityId":null,"pBranchCity":null,"receiverId":null,"receiverEMoneyAccount":null,"receiverFullName":"Luqman Ahmed","receiverMobile":"252617335853","receiverTel":null,"sDescription":null,"remittancePurposeId":2,"customerRelationshipId":null,"amount":44,"comm":5,"isDiscounted":false,"sCurrencyRate":1,"sendingAccountTypeId":1,"sGlAccountId":7948,"sGlAccountTitle":"Cash in hand USD SWTST","transferAmount":44,"serviceCharge":5,"sCurrencyCode":"USD","receivingAmount":"44","pCurrencyCode":"USD","pCurrencyId":271,"payment":[{"type":1,"amount":44,"cheque":"","dueDate":"","bankName":"","chequeComments":"","paymentMethod":"Visa"}],"remittanceSubTypeId":null}]}}

**Required**

"guid" is a required parameter and it’s a unique number, length can be max 32

"senderFirstName" is a required parameter and it’s a string

" senderLastName " is a required parameter and it’s a string

" senderMobile " is a required parameter and it’s a string

"remittanceTypeId" is a required parameter and it’s a number

" receiverFullName " is a required parameter and it’s a string

" receiverMobile " is a required parameter and it’s a string

" remittancePurposeId " is a required parameter and it’s a number

" amount " is a required parameter and it’s a number

" comm" is a required parameter and it’s a number

" sCurrencyRate" is a required parameter and it’s a number

" sendingAccountTypeId " is a required parameter and it’s a number (use default value as 1)

" sGlAccountId " is a required parameter and it’s a number (

use: "branchDefaultCashAccountId” from login response)

" sGlAccountTitle " is a required parameter and it’s a string (use: "branchDefaultCashAccountName" from login response)

"transferAmount" is a required parameter and it’s a number

"serviceCharge" is a required parameter and it’s a number

"sCurrencyCode" is a required parameter and it’s a string

"receivingAmount" is a required parameter and it’s a number

"pCurrencyCode" is a required parameter and it’s a string

"pCurrencyId" is a required parameter and it’s a number

"payment" is a required parameter and it’s an array

[

type: paymentMethodId (use: “paymentMethodId” from getpaymentmethods response e.g. 5),

amount: required and it’s a number,

cheque: N/A,

dueDate: N/A,

bankName: N/A,

chequeComments: N/A,

paymentMethod: paymentMethod (use: “description” from getpaymentmethods response e.g. "Visa")

]

* **Success Response**:

{"err":null,"payload":{"result":[{"status":1,"remittances":[{"remittanceId":1028517,"remittanceCode":"1711211028517","senderVerificationCode":"","remittanceTypeId":1,"receiverMobile":"252617335853","pBranchCity":null,"amount":11,"comm":5,"receiverFullName":"Irfan Ahmed","receivingAmount":"11","pCurrencyCode":"USD","transferAmount":11,"sCurrencyCode":"USD","serviceCharge":5}],"guid":"e7c8c0c7-4816-e5c7-385e-1fa75db8a8db"}]}}

* **Error Response**:

{"err”: ["Fail to find request data"],"payload”: {"result”: null}}

**GET CUSTOMER TRANSACTIONS**

* **Description**:

Returns the transaction history for the current user.

* **Url**:

/orm/remittancemanagement/getcusttransactions

* **Method**:

POST

* **Url Params**:

NONE

* **Data Params**:

{"payload”: {"customerId”: value, "numberOfDays":30}}

**Required**

“customerId” is a required parameter and its value is number e.g. 1326, a user needs to make at least one successful transaction to get registered as a customer.

“numberOfDays” is a required parameter and its value is number e.g. 30 (default)

* **Success Response**:

{"err":null,"payload":{"result":[{"remittanceId":1028626,"sBranchId":759,"pBranchId":395,"sCustomerId":1348963,"rCustomerId":1348964,"amount":22,"comm":5,"sDate":"2017-12-07T14:16:46.771Z","pDate":null,"sDescription":null,"pDescription":null,"isBranchApproved":"0","isParentApproved":"1","isRegionalApproved":"1","isCpoApproved":"0","remittanceStateId":1,"remittanceTypeId":1,"remittancePurposeId":1,"sGlAccountId":7948,"pGlAccountId":null,"sCurrencyRate":1,"pCurrencyRate":null,"isFrozen":"0","receiverFirstName":"rece","receiverMiddleName":null,"receiverLastName":"first","receiverMobile":"252617335859","senderFirstName":"Ahmed","senderMiddleName":null,"senderLastName":"Ali","senderMobile":"252617335800","sendingBranchCode":"SWTST","payingBranchCode":"FSTPCH","remittanceCode":"1712071028626"},{"remittanceId":1028625,"sBranchId":759,"pBranchId":395,"sCustomerId":1348963,"rCustomerId":1348964,"amount":35,"comm":5,"sDate":"2017-12-07T14:07:33.858Z","pDate":null,"sDescription":null,"pDescription":null,"isBranchApproved":"0","isParentApproved":"1","isRegionalApproved":"1","isCpoApproved":"0","remittanceStateId":1,"remittanceTypeId":1,"remittancePurposeId":1,"sGlAccountId":7948,"pGlAccountId":null,"sCurrencyRate":1,"pCurrencyRate":null,"isFrozen":"0","receiverFirstName":"receiver","receiverMiddleName":null,"receiverLastName":"two","receiverMobile":"252617335859","senderFirstName":"Ahmed","senderMiddleName":null,"senderLastName":"Ali","senderMobile":"252617335800","sendingBranchCode":"SWTST","payingBranchCode":"FSTPCH","remittanceCode":"1712071028625"}]}}

**Error Response**:

{"err”: ["Fail to find request data"],"payload”: {"result”: null}}

**GET ADD REMITTANCE VIEW**

* **Description**:

API is used to get the remittance configuration for a given sending country. The configuration may vary from the send form fields (isMandatory, isVisible, isActive,) to other various configurations.

* **Url**:

/orm/remittancemanagement/getaddremittanceview

* **Method**:

POST

* **Url Params**:

NONE

* **Data Params**:

{"payload”: {"sCountryId”: value}}

**Required**

“sCountryId” is a required parameter and its value is the number e.g. 245 for UK

* **Success Response**:

{"err":null,"payload": {"result":[{"remittanceConfigs":[{"remittanceConfigId":8,"description":"Send remittance sender date of birth ALL","countryId":null,"countryName":null,"remittanceFormFieldId":8,"remittanceFormId":1,"fieldName":"send\_remittance\_sender\_dateofbirth","sortOrder":1,"isVisible":"1","isEnabled":"1","isMandatoryField":"0","isComplianceField":"1","labelValue":"Date of birth","isActive":"1"},{"remittanceConfigId":7,"description":"Send remittance sender country ALL","countryId":null,"countryName":null,"remittanceFormFieldId":7,"remittanceFormId":1,"fieldName":"send\_remittance\_sender\_country","sortOrder":1,"isVisible":"1","isEnabled":"1","isMandatoryField":"0","isComplianceField":"1","labelValue":"Country","isActive":"1"},{"remittanceConfigId":5,"description":"Send remittance sender city ALL","countryId":null,"countryName":null,"remittanceFormFieldId":5,"remittanceFormId":1,"fieldName":"send\_remittance\_sender\_city","sortOrder":1,"isVisible":"1","isEnabled":"1","isMandatoryField":"0","isComplianceField":"1","labelValue":"City","isActive":"1"},{"remittanceConfigId":4,"description":"Send remittance sender street ALL","countryId":null,"countryName":null,"remittanceFormFieldId":4,"remittanceFormId":1,"fieldName":"send\_remittance\_sender\_address","sortOrder":1,"isVisible":"1","isEnabled":"1","isMandatoryField":"0","isComplianceField":"1","labelValue":"Street","isActive":"1"},{"remittanceConfigId":9,"description":"Send remittance sender place of birth ALL","countryId":null,"countryName":null,"remittanceFormFieldId":9,"remittanceFormId":1,"fieldName":"send\_remittance\_sender\_placeofbirth","sortOrder":1,"isVisible":"1","isEnabled":"1","isMandatoryField":"0","isComplianceField":"1","labelValue":"Place of birth","isActive":"1"},{"remittanceConfigId":10,"description":"Send remittance sender nationality ALL","countryId":null,"countryName":null,"remittanceFormFieldId":10,"remittanceFormId":1,"fieldName":"send\_remittance\_sender\_nationality","sortOrder":1,"isVisible":"1","isEnabled":"1","isMandatoryField":"0","isComplianceField":"1","labelValue":"Nationality","isActive":"1"},{"remittanceConfigId":11,"description":"Send remittance sender occupation ALL","countryId":null,"countryName":null,"remittanceFormFieldId":11,"remittanceFormId":1,"fieldName":"send\_remittance\_sender\_occupation","sortOrder":1,"isVisible":"1","isEnabled":"1","isMandatoryField":"0","isComplianceField":"1","labelValue":"Occupation","isActive":"1"},{"remittanceConfigId":12,"description":"Send remittance sender source of funds ALL","countryId":null,"countryName":null,"remittanceFormFieldId":12,"remittanceFormId":1,"fieldName":"send\_remittance\_sender\_sourceoffunds","sortOrder":1,"isVisible":"1","isEnabled":"1","isMandatoryField":"0","isComplianceField":"1","labelValue":"Source of funds","isActive":"1"},{"remittanceConfigId":13,"description":"Send remittance sender primary document ALL","countryId":null,"countryName":null,"remittanceFormFieldId":13,"remittanceFormId":1,"fieldName":"send\_remittance\_sender\_primary\_document","sortOrder":1,"isVisible":"1","isEnabled":"1","isMandatoryField":"0","isComplianceField":"1","labelValue":"Primary document","isActive":"1"},{"remittanceConfigId":1,"description":"Send remittance sender tel ALL","countryId":null,"countryName":null,"remittanceFormFieldId":1,"remittanceFormId":1,"fieldName":"send\_remittance\_sender\_tel","sortOrder":1,"isVisible":"1","isEnabled":"1","isMandatoryField":"0","isComplianceField":"0","labelValue":"Tel","isActive":"1"},{"remittanceConfigId":33,"description":"Send remittance sender relationship ALL","countryId":null,"countryName":null,"remittanceFormFieldId":31,"remittanceFormId":1,"fieldName":"send\_remittance\_sender\_receiver\_relation","sortOrder":1,"isVisible":"1","isEnabled":"1","isMandatoryField":"0","isComplianceField":"0","labelValue":"Relationship","isActive":"1"}]}]}}

**Error Response**:

{"err”: ["Fail to find request data"],"payload”: {"result”: null}}

**GET PAYMENT METHODS**

* **Description**:

Returns the list of available payment methods for the given portal (web, mobile).

* **Url**:

/orm/remittancemanagement/getpaymentmethods

* **Method**:

POST

* **Url Params**:

NONE

* **Data Params**:

{"payload”: {" paymentMethodTypeId”: value}}

**Required**

“paymentMethodTypeId” is a required parameter and its value is number e.g. (for Mobile App send 2)

* **Success Response**:

{"err”: null, "payload”: {"result”: [ {“paymentMethodId”: 5, “description”: "Visa"}, {“paymentMethodId”: 6, “description”: "MasterCard"}]}}

* **Error Response**:

{"err”: ["Fail to find request data"],"payload”: {"result”: null}}

**REGISTER PAYMENT WITH PAYMENT GATEWAY**

* **Description**:

API is used to register the online payment transaction with the payment gateway

* **Url**:

/orm/remittancemanagement/registeronlinepayment

* **Method**:

POST

* **Url Params**:

NONE

* **Data Params**:

{"payload":{"guid":"6fc64d0a23232322322","senderIsSanctioned":false,"senderTypeId":null,"senderId":1348907,"senderFirstName":"Ahmed","senderMiddleName":null,"senderLastName":"Ali","senderMobile":"9652927226","senderTel":null,"senderCompanyRegistrationNumber":null,"senderAddress":null,"senderPostCode":null,"senderBuildingNumber":null,"senderState":null,"senderCity":null,"senderCountryId":null,"senderDateOfBirth":null,"senderPlaceOfBirth":null,"senderNationalityId":null,"senderNationalId":null,"senderOccupationId":null,"senderSourceOfFundId":null,"senderConductorId":null,"senderRelationshipId":null,"senderRelationship":null,"senderConductorName":null,"senderDocuments":[],"remittances":[{"receiverEMoneyAccountVerified":false,"receiverIsSanctioned":false,"remittanceTypeId":1,"pBranchId":null,"payingBranchCode":null,"pBranchCountryId":129,"pBranchCountry":"Kenya","pBranchCountryIso2":"KE","pBranchAddress":null,"pBranchCityId":null,"pBranchCity":null,"receiverId":null,"receiverEMoneyAccount":null,"receiverFullName":"Luqman Ahmed","receiverMobile":"252617335853","receiverTel":null,"sDescription":null,"remittancePurposeId":2,"customerRelationshipId":null,"amount":44,"comm":5,"isDiscounted":false,"sCurrencyRate":1,"sendingAccountTypeId":1,"sGlAccountId":7948,"sGlAccountTitle":"Cash in hand USD SWTST","transferAmount":44,"serviceCharge":5,"sCurrencyCode":"USD","receivingAmount":"44","pCurrencyCode":"USD","pCurrencyId":271,"payment":[{"type":5,"amount":44,"cheque":"","dueDate":"","bankName":"","chequeComments":"","paymentMethod":"Visa"}],"remittanceSubTypeId":null}]}}

**Required**

"guid" is a required parameter and it’s a unique number, length can be max 32

"senderFirstName" is a required parameter and it’s a string

" senderLastName " is a required parameter and it’s a string

" senderMobile " is a required parameter and it’s a string

"remittanceTypeId" is a required parameter and it’s a number

" receiverFullName " is a required parameter and it’s a string

" receiverMobile " is a required parameter and it’s a string

" remittancePurposeId " is a required parameter and it’s a number

" amount " is a required parameter and it’s a number

" comm" is a required parameter and it’s a number

" sCurrencyRate" is a required parameter and it’s a number

" sendingAccountTypeId " is a required parameter and it’s a number (use default value as 1)

" sGlAccountId " is a required parameter and it’s a number (

use: "branchDefaultCashAccountId” from login response)

" sGlAccountTitle " is a required parameter and it’s a string (use: "branchDefaultCashAccountName" from login response)

"transferAmount" is a required parameter and it’s a number

"serviceCharge" is a required parameter and it’s a number

"sCurrencyCode" is a required parameter and it’s a string

"receivingAmount" is a required parameter and it’s a number

"pCurrencyCode" is a required parameter and it’s a string

"pCurrencyId" is a required parameter and it’s a number

"payment" is a required parameter and it’s an array

[

“type”: paymentMethodId (use: “paymentMethodId” from getpaymentmethods response e.g. 5),

“amount”: required and it’s a number,

“cheque”: N/A,

“dueDate”: N/A,

“bankName”: N/A,

“chequeComments”: N/A,

“paymentMethod”: paymentMethod (use: “description” from getpaymentmethods response e.g. "Visa")

]

* **Success Response**:

{"err”: null, "payload”: {"result”: [{"transactionId": “VMPH554R4XT8H543NX45”,

"terminalUrl”: “<https://test.epayment.nets.eu/terminal/default.aspx?merchantId=647767&transactionId=VMPH554R4XT8H543NX45>”, “orderNumber”:” R79S825GH6”}]}}

* **Error Response**:

{"err”: ["Fail to find request data"],"payload”: {"result”: null}}

**PROCESS PAYMENT GATEWAY TERMINAL RESPONSE**

* **Description**:

API is used to process the actual payment or capture the amount from the customer account after payment validation is “OK” with payment gateway. (Response code “OK” is send to the redirect URL after payment is validated in the payment gateway terminal).

* **Url**:

/orm/remittancemanagement/processonlinepayment

* **Method**:

POST

* **Url Params**:

NONE

* **Data Params**:

{"payload”: {"guid":" M42R4624FP8J9P794ZW4","operation":” sale”}}

**Required**

"guid" is a required parameter and it’s a unique number, length can be max 32

"operation" is a required parameter and it’s a string (use “sale” for processing the payment)

* **Success Response**:

{"err”: null, "payload”: {"result”: [{"transactionId": “M42R4624FP8J9P794ZW4”, "responseCode":” OK”}]}}

* **Error Response**:

{"err”: ["Fail to find request data"],"payload”: {"result”: null}}

**GET RECEIVER LAST REMITTANCE DETAILS**

* **Description**:

API is used to get the last remittance info for the selected receiver or recipient and it contains the receiver country, amount info.

* **Url**:

/orm/remittancemanagement/getreceiverlastremittancedetails

* **Method**:

POST

* **Url Params**:

NONE

* **Data Params**:

{"payload”: {"senderId":1326,"receiverId":5565}}

**Required**

“senderId” is the remittance sender id and is a required parameter and its value is number e.g. 1326, a user needs to make at least one successful transaction to get registered as a customer.

“receiverId” is the remittance receiver id and is a required parameter and its value is number e.g. 5565.

* **Success Response**:

{"err”: null, "payload”: {"result”: [{"remittanceTypeId":1,"receiverCustomerName":"ahmed ali","receiverCustomerMobile":"252634335853","receiverCustomerTel”: null, "receiverCustomerAddress”: null, "receiverCustomerCity”: null,"receiverCustomerCountryId":211,"payingBranchCode":"FSTPCH","remittancePurposeId":1,"amount":25,"customerRelationshipId”: null, "remittanceSubTypeId”: null}]}}

* **Error Response**:

{"err”: ["Fail to find request data"],"payload”: {"result”: null}}

**GET REMITTANCE PURPOSES**

* **Description**:

API is used to get remittance purposes or remittance sending reasons

* **Url**:

/orm/remittancemanagement/getremittancepurposes

* **Method**:

POST

* **Url Params**:

NONE

* **Data Params**:

{"payload”: {}}

* **Success Response**:
* {"err”: null, "payload”: {"result”: [ {“remittancePurposeId”: 1, “description”: "Family Support"}, {“remittancePurposeId”: 2, “description”: "Personal Needs"}]}}
* **Error Response**:

{"err”: ["Fail to find request data"],"payload”: {"result”: null}}