used create_payment from website model

- used create_payment from website model
- `def create_payment(self, new_ebs_no=None, ob_balance=None,mobile_ebs_no=False,mobile_pay_reference=False):`
- `where we will pass the mobile reference and ebs no and call the outstanding balance function which will create the entry in ebs.online.payment with the same way in portal for the api quick_payment`
- `we will update the amount paid for the corresponding account and the once payment done if there is value inside amount pay ebs.gas.payment will be created`
- `quick_payment_link for this api we will recieve the id for the ebs online reference payment and check if the amount is correct if so will create the link and update the reference and id from transaction once the payment done and save it in ebs online payment reference`

Please find the below 2 api's you can you for both Login (where values will be passed on header)and Gas bill payment (values will be passed on body)
http://40.120.88.200:8269/api_v3/ebs/mobile_payment_multi_gas_account
http://40.120.88.200:8269/api_v3/ebs/mobile_multi_gas_account_payment_link

IF THE SINGLE GAS ACCOUNT HAS BALANCE WILL PASS IT INTO THE SAME

1. **http://40.120.88.200:8269/api_v3/ebs/mobile_payment_multi_gas_account**

INPUT:
{
"customer_number" :"L222363"
}

OUTPUT:
"status": true,
"customer_number": "L222363",
"outstanding_balance": 1225.87,
"ebs_online_payment_ref": 348,
"accounts_info": [
{
"gas_account_number": "1502040ON1PJ001",
"gas_account_id": 127059,

```
"customer_name": "Dynasty Properties International FZ-LLC",
"premise": "PJ-The One",
"unit": "0204",
"due_amount": 497.46
},
{
"gas_account_number": "1502050ON1PJ001",
"gas_account_id": 130789,
"customer_name": "Rayvella Properties International Limited",
"premise": "PJ-The One",
"unit": "0205",
"due_amount": 236.13
},
{
"gas_account_number": "1517040ON1PJ001",
"gas_account_id": 131214,
"customer_name": "Dash Properties International FZ-LLC",
"premise": "PJ-The One",
"unit": "1704",
"due_amount": 237.29
},
{
"gas_account_number": "1503030ON1PJ001",
"gas_account_id": 131215,
"customer_name": "Dynasty Properties International FZ-LLC",
"premise": "PJ-The One",
"unit": "0303",
"due_amount": 254.99
}
],
"message": "Success"
}
```

2. http://40.120.88.200:8269/api_v3/ebs/mobile_multi_gas_account_payment_link
   (will create ebs online payment here due to concurrent issue)
   INPUT:

```
{
"customer_number" :"L222363",
"res_values":[
{
"gas_account_id": 127059,
"amount": 100
},
{
"gas_account_id": 130789,
"amount": 100
}
],
"mobile":"0565459140",
"amount":"200",
"ebs_online_payment_ref":"348"
}
```

OUTPUT:

```
{
"status": true,
"link":
"https://secure.paytabs.com/payment/wr/5EB4084C82E4D03872FFFE6D5158B850747A1168DC5FFD54379CF6C0"
}
```

app_payment_link--> login payment
# this will be shown by him on mobile we limited the length to gas account length to display the values in mobile with gas account/premise/unit

LPAY (APAY) Multi

1. **http://40.120.88.200:8269/api_v3/ebs/mobile_payment_multi_gas_account**
   request.httprequest.headers.get('user_id')
   this is he will pass the values to the header id so we use this to get the value in the body he will not pass anything
2. **http://40.120.88.200:8269/api_v3/ebs/mobile_multi_gas_account_payment_link**

INPUT:

```
{
"customer_number" :"L222363",
"res_values":[
{
"gas_account_id": 127059,
"amount": 100
},
{
"gas_account_id": 130789,
"amount": 100
}
],
"mobile":"0565459140",
"amount":"200",
"ebs_online_payment_ref":"348"
}
```

OUTPUT:

```
{
"status": true,
"link":
"https://secure.paytabs.com/payment/wr/5EB4084C82E4D03872FFFE6D5158B850747A1168DC5FFD54379CF6C0"
}
```

calling This fn **paytabs_payment_generate_values** from
cust_method_name = '%s_payment_generate_values' % (acquirer_id.provider)

will give
'payment_url':
'https://secure.paytabs.com/payment/wr/5EA1CD2B82E4108007929469ACD1B349583770EBC555DB94A2C44EBD', 'tran_ref': 'TST2312301576925'}

added now message if we get the postman code of 406 where the limit exceeds

2. then he will click this below api to get the below output and payment will be generated and from here he will generate the receipt.
   http://40.120.88.200:8269/api_v3/ebs/quick_payment_success_link

o/p:

```
{
"status": true,
"receipt": "http://40.120.88.200:8269/my/payment/print/2728036",
"file_name": "TST2231401378043",
"code": "GPAY-20221110174252",
"amount": 200.0,
"message": "Payment Processed sucessfully"
}
```

quick_payment_success_link

here we are giving the filename to download

LPAY (APAY) Single

1. get_customer_balance

2. app_payment_link (will create ebs online payment here due to concurrent issue)
3. quick_payment_success_link

OPAY added advance payment

1. /init_payment_details (to add advance payment inside dropdown in notification.mobile.master)
2. other_pay_payment_link
3. advance_payment_ebs_no (field which will store either gas/account or ebs number) https://eu-gateway.mastercard.com/api/documentation/integrationGuidelines/supportedFeatures/testAndGoLive.html?locale=en_US

Other Payment

1. When pay now is clicked, the "**other_pay_payment_link**" api will send the payment url and the transaction reference.
2. To produce a draft payment transaction, we shall call the "**mobile_payment_vals**" function.
3. The "**paytabs_payment_generate_values**" function posts requests to the 'https://secure.paytabs.com/payment/request' api with the authorisation and by giving parameters with the amount, the return URL, the callback URL, the cancel URL, and the customer information. This function returns the payment url and the transaction reference.
4. A mobile payment page will open, containing a form for entering credit card information.
5. After submission, the mobile developer will have already saved the reference(from 1st api) that he would provide over the 2nd api "**quick_payment_success_link**"
6. After that, we will use this reference to call the function "**paytabs_mobile_payment_return**," which will parse the reference (since the url was previously sent) and get the paytab payment status response by passing this reference as input to postman to the api "https://secure.paytabs.com/payment/query." From there, we will update the txn state and values by calling "**form_feedback**" and "**_paytabs_form_validate**,"
7. Consequently, we will send the payment_status success/failed/pending based on the response and the txn state, and the matching dialog box will be displayed inside the mobile with a printout and with a color code.

app_payment_link
for pdr payment