

Gas Status Change Request**

** 1. Gas Status Change Request**

http://40.120.88.200:8269/api_v3/status/change_request_account_info

**<!-- create the values in portal for the submitted data and will send the selected gas account in list and also send to show either the refund page or payment page

- first will check if the gas account has refund bank details
- if yes then will check the OB

-->**

http://40.120.88.200:8269/api_v3/status/gas_status_change_request

**

**

http://40.120.88.200:8269/api_v3/status/refund_bank_master_data

**

**

http://40.120.88.200:8269/api_v3/status/bank_account_update

2. APAY

- 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 receive 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 use 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

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

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 (will give the print out and do the payment)

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)

LOGIN

1. api_v3/notify/get_notify_token
 2. String model = 'ebs.gas.invoice';
String method = 'get_customer_balance';
 3. String model = 'ebs.customer.gas.account';
String method = 'user_account_info';
-