1) First request from Merchant to PayU with the required transaction mandatory/ optional parameters. This needs to be a server to server curl call request.

PayU URL Endpoint:

Live Environment: https://secure.payu.in/_payment

Test Environment: https://test.payu.in/_payment

Sr. No	Variable	Description
1)	key **(Mandatory)**	This parameter is the unique Merchant Key provided by PayU for your merchant account. The Merchant Key acts as the unique identifier (primary key) to identify a particular Merchant Account in our database. While posting the data to us, you need to put this Merchant Key value for you merchant account in this parameter. Also, please note that during integration with PayU, you would need to first integrate with our Test Server. PayU would be providing you the necessary Merchant Key for test server. Please do not use your live account's merchant key here. It would not work. Once testing is done, you are ready to move to live server. Here, you would need to replace the test Merchant Key with Live Merchant Key. This is a critical step for successfully moving to live PayU server. Example: C0Ds8q
2)	txnid **(Mandatory)**	This parameter is known as Transaction ID (or Order ID). It is the order reference number generated at your (Merchant's) end. It is an identifier which you (merchant) would use to track a particular order. If a transaction using a particular transaction ID has already been successful at PayU, the usage of same Transaction ID again would fail. Hence, it is essential that you post us a unique transaction ID for every new transaction. (Please make sure that the transaction ID being sent to us hasn't been successful earlier. In case of this duplication, the customer would get an error of 'duplicate Order ID'). Data Type – Varchar Character Limit – 25 characters Example: fd3e847h2
3)	amount **(Mandatory)**	This parameter should contain the payment amount of the particular transaction. Note: Please type-cast the amount to float type Example: 10.00

4)	productinfo **(Mandatory)**	This parameter should contain a brief product description. It should be a string describing the product (The description type is entirely your choice). Data type - Varchar Character Limit – 100 characters Example: tshirt100
5)	firstname **(Mandatory)**	Self-Explanatory (Must contain the first name of the customer) Data Type – Varchar Character Limit – 60 characters Example: Ankit
6)	email **(Mandatory)**	Self-explanatory (Must contain the email of the customer) Data type – Varchar Character Limit – 50 Example: test@gmail.com
7)	phone **(Mandatory)**	Self-explanatory (Must contain the phone number of the customer) Data type – Varchar Character Limit – 50 (numeric value only) Example: 9999999999
8)	surl **(Mandatory)**	Merchant URL on which PayU will post the final transaction response, if the transaction is **success**
9)	furl **(Mandatory)**	Merchant URL on which PayU will post the final transaction response, if the transaction is **failed**
10)	**txn_s2s_flow (Mandatory)**	Please pass its value as 1

12)	hash **(Mandatory)**	Hash is a crucial parameter – used specifically to avoid any tampering during the transaction. There are two different methods to calculate hash. **Please follow method 1 only.** Method 2 is just there for the documentation and is not to be used. Method 1 - This is the simplest way of calculating the hash value. Here, please make sure that the **api_version** parameter is **NOT POSTED** from your end. For hash calculation, you need to generate a string using certain parameters and apply the sha512 algorithm on this string. Please note that you have to use pipe (|) character in between these parameters as mentioned below. The parameter order is mentioned below: **sha512(key|txnid|amount|productinfo|firstname|email|udf1|udf2|udf3|udf4|udf5|||||SALT)** All these parameters (and their descriptions) have already been mentioned earlier in this table. Here, SALT (to be provided by PayU), key, txnid, amount, productinfo, firstname, email are **mandatory** parameters and hence **can't be empty** in hash calculation above. But, udf1-udf5 are **optional** and hence you need to calculate the hash based upon the fact that whether you are posting a particular udf or not. For example, if you are NOT posting udf1. Then, in the hash calculation, udf1 field will be left empty. Following examples will clarify various scenarios of hash calculation: Case 1: If all the udf parameters (udf1-udf5) are posted by the merchant. Then, hash=sha512(**key|txnid|amount|productinfo|firstname|email|udf1|udf2|udf3|udf4|udf5|||||SALT**) Case 2: If only some of the udf parameters are posted and others are not. For example, if udf2 and udf4 are posted and udf1, udf3, udf5 are not. Then, hash=sha512(**key|txnid|amount|productinfo|firstname|email||udf2||udf4|||||SALT**) Case 3: If NONE of the udf parameters (udf1-udf5) are posted. Then, hash=sha512(**key|txnid|amount|productinfo|firstname|email||||||||||SALT**) Example: If key=**C0Dr8m**, txnid=**12345**, amount=**10**, productinfo=**Shopping**, firstname=**Test**, email=**test@test.com**, udf2=**abc**, udf4=**15**, SALT=**3sf0jURk** and udf1, udf3, udf5 are **not** posted. Then, hash would be calculated as Case 2 above:

13)	pg (**Mandatory**)	It must be set as the payment category. Please set its value to '**NB'** for Net Banking , '**CC'** for Credit/Debit Card, '**CASH'** for Cash Card and '**EMI'** for EMI
14)	bankcode (**Mandatory**)	Please set it as '**CC/DC'** for CC/DC transactions. For NB, please refer to bankcode list for specific bank.
15)	ccnum (**Mandatory**)	This parameter must contain the card (credit/debit) number entered by the customer for the transaction.
16)	ccname (**Mandatory**)	This parameter must contain the name on card – as entered by the customer for the transaction.
17)	ccvv (**Mandatory**)	This parameter must contain the cvv number of the card – as entered by the customer for the transaction.
18)	ccexpmon (**Mandatory**)	This parameter must contain the card's expiry month - as entered by the customer for the transaction. Please make sure that this is always in 2 digits. For months 1-9, this parameter must be appended with 0 – like 01, 02…09. For months 10-12, this parameter must not be appended – It should be 10, 11 and 12 respectively.
19)	ccexpyr (**Mandatory**)	The customer must contain the card's expiry year – as entered by the customer for the transaction. It must be of 4 digits. For example - 2017, 2029 etc.
20)	lastname	Self-Explanatory (only alphabets a-z are allowed) Data Type – Varchar Character Limit – 20 characters Example: Verma
21)	address1	Self-Explanatory Data Type – Varchar Character Limit – 100 Characters allowed : A to Z, a to z, 0 to 9, @, - (Minus), _ (Underscore), / (Backslash), (Space), (Dot)
22)	address2	Self-explanatory Data Type – Varchar Character Limit – 100 (Allowed characters are same as for address1 parameter)
23)	city	Self-explanatory Data type – Varchar Character Limit – 50 (Allowed characters are same as for address1 parameter)

24)	state	Self-explanatory Data type – Varchar Character Limit – 50 (Allowed characters are same as in address parameter)
25)	country	Self-explanatory Data type – Varchar Character Limit – 50 (Allowed characters are same as in address parameter)
26)	zipcode	Self-explanatory Data type – Varchar Character Limit – 20 (Only numeric value allowed)
27)	udf1	User defined field 1 – This parameter has been made for you to keep any information corresponding to the transaction, which may be useful for you to keep in the database. UDF1-UDF5 fields are for this purpose only. It's completely for your usage and you can post any string value in this parameter. udf1-udf5 are optional parameters and you may use them only if needed Data type – Varchar Character Limit – 255
28)	udf2	User defined field 2 – Same description as UDF1 Data type – Varchar Character Limit – 255
29)	udf3	User defined field 3 – Same description as UDF1 Data type - Varchar Character Limit – 255
30)	udf4	User defined field 4 – Same description as UDF1 Data type – Varchar Character Limit – 255
31)	udf5	User defined field 5 – Same description as UDF1 Data type – Varchar Character Limit – 255
32)	offer_key	This parameter is useful when the merchant wants to give the customer a discount offer on certain transactions based upon a pre-defined combination. This combination can be based upon payment options/bins etc. For each new offer created, a unique **offer_key** is generated. At the time of a transaction, this offer_key needs to be posted by the merchant.
33)	s2s_client_ip **(mandatory)**	This parameter must have the source IP of the user
34)	s2s_device_info (**mandatory**)	This parameter must have the user agent of device

2) Response from PayU to Merchant. It would be an S2S call response.

a) **If there is an error in request processed**, below error will be sent to merchant:

{"result":null,"status":"failed","error":"E1616","message":"Non-seamless not allowed in S2S Flow"}

b) **If the request is correctly processed**, then a curl response back to merchant in response to step (1).

Sr. No	Variable	Description
1)	post_data	base64 encoded html page which needs to be pushed on user's browser, it will auto submit to bank's page.
2)	post_uri	Bank URL where data needs to be posted to, by the merchant

post_data is a base64 encoded string. The merchant needs to decode **post_data**, which is an html form with auto submit, which then needs to be shown on the customer's browser. The html being auto submit, it will take the customer to the bank page for required credentials.

Example of base 64 encoded and its decoded response:

Response (Example):

eyJzdGF0dXMiOiJzdWNjZXNzIiwicmVzdWx0Ijp7Im1paHBheWlkIjoiNzAwMDEwMDA0MzY2MDE
iLCJtb2RlIjoiREMiLCJzdGF0dXMiOiJmYWlsdXJlIiwia2V5Ijoib2ZTeTFEIiwidHhuaWQiOiJjMGUy
ZTA4ZDc1MTNmNjlkY2M1OSIsImFtb3VudCI6IjEuMjAiLCJhZGRlZG9uIjoiMjAxNy0wMi0yMSAxO
DozMTozNCIsInByb2R1Y3RpbmZvIjoiUHJvZHVjdCBJbmZvIiwiZmlyc3RuYW1lIjoiUGF5dS1BZG1
pbiIsImxhc3RuYW1lIjoiIiwiYWRkcmVzczEiOiIiLCJhZGRyZXNzMiI6IiIsImNpdHkiOiIiLCJzdGF0ZS
I6IiIsImNvdW50cnkiOiIiLCJ6aXBjb2RlIjoiIiwiZW1haWwiOiJ0ZXN0QGV4YW1wbGUuY29tIiwicG
hvbmUiOiIxMjM0NTY3ODkwIiwidWRmMSI6IiIsInVkZjIiOiIiLCJ1ZGYzIjoiIiwidWRmNCI6IiIsInVkZ
jUiOiIiLCJ1ZGY2IjoiIiwidWRmNyI6IiIsInVkZjgiOiIiLCJ1ZGY5IjoiIiwidWRmMTAiOiIiLCJjYXJkX3
Rva2VuIjoiIiwiY2FyZF9ubyI6IjQ1OTE1MFhYWFhYWDUyNzkiLCJmaWVsZDAiOiIiLCJmaWVsZDEi
OiIxMTcwMjIxOTQwMzI0NjgiLCJmaWVsZDIiOiIiLCJmaWVsZDMiOiIiLCJmaWVsZDQiOiIiLCJma
WVsZDUiOiIyIiwiZmllbGQ2IjoiQXV0aGVudGljYXRpb24gZmFpbGVkLlRyYW5zYWN0aW9uIGNhb
m5vdCBiZSBhdXRob3JpemVkIiwiZmllbGQ3IjoiRW1wdHkgdmFsdWUgcmVjZWl2ZWQsIHZhbHV
lcyBhcmUgcnJuID0gLHRyYW5zX2lkID0gMTE3MDIyMTk0MDMyNDY4LGF1dGhfY29kID0gIiwiZmll
bGQ4IjoiIiwiZmllbGQ5IjoiQXV0aGVudGljYXRpb24gZmFpbGVkLlRyYW5zYWN0aW9uIGNhbm5v
dCBiZSBhdXRob3JpemVkIiwicGF5bWVudF9zb3VyY2UiOiJwYXI1IiwiUEdfVFlQRSI6IlNCSVBHIiwi
ZXJyb3IiOiJFMzAzIiwiZXJyb3JfTWVzc2FnZSI6IkNhcmQgYXV0aGVudGljYXRpb24gZmFpbHVyZS
IsIm5IdF9hbW91bnRfZGViaXQiOiIwIiwidW5tYXBwZWRzdGF0dXMiOiJmYWlsZWQiLCJoYXNoIjo
iMjAzZTgxMTVlM2FmNGZlN2UyMDEyNDlkNmZiZDA5MzU4YzBlNTA1ODgxZGMxODI2ZDk0ZDAz
ODczMTk3ZDFkZjQ1OTUzMGE5NGI5ZWZiNWQyMWZmZjI4MWI0MWEwMzA0YTcxNjY3MjExM2E
zMTY2M2FlMGViMThmYzM4Njg5ODciLCJiYW5rX3JlZl9ubyI6IiIsImJhbmtfcmVmX251bSI6IiIsIm
Jhbmtjb2RlIjoiVklTQSIsInN1cmwiOiJodHRwczpcL1wvYWRtaW4ucGF5dS5pblwvdGVzdF9yZXN
wb25zZSIsImN1cmwiOiJodHRwczpcL1wvYWRtaW4ucGF5dS5pblwvdGVzdF9yZXNwb25zZSIsI
mZ1cmwiOiJodHRwczpcL1wvYWRtaW4ucGF5dS5pblwvdGVzdF9yZXNwb25zZSIsImNhcmRfaG
FzaCI6ImYzNmM3NDhkNjRhZmUxODNiYzNiNWQwOTJlNDI1MjA5ZDk3MGUxNmM3NmIzMzBiO
DEzNDc3YWQzNWQ1NTIhN2YifX0=

Decoded response below (Example):

```
<html><body><form name="payment_post" id="payment_post" action="http://
axisnetbanking.com" method="post"><input type="hidden" name="RU" value="https://
zomato.com/resulthandling?
payu_response=aHR0cHM6Ly9zZWN1cmUucGF5dS5pbi9BWEITX05CX1JFU1BPTlNFLnBocA==&
reference=1e281da1d50a5ae1d3217570d3761dcf81db4e7c2c6c60f0ef601df5f8647202"><inp
ut type="hidden" name="qs"
value="PRN~12d457876d30e900a9e0$PID~000000002438$MD~P$ITC~70000000097771$CRN~I
NR$AMT~10.00$RESPONSE~AUTO$CG~Y"></form><script type='text/javascript'>

        window.onload=function(){

            document.forms['payment_post'].submit();

        }

    </script></body></html>
```

3. Once the customer is redirected to bank's page where customer authentication is done, bank gives back the redirect response to PayU and PayU gives the final transaction response to merchant via a redirect. Below parameters will be sent in this response. The merchant must refer to '**status'** parameter mentioned in the table below for processing/ discarding the order.

Sr.No	Variable Name	Description
1	mihpayid	It is a unique reference number created for each transaction at PayU's end. For every new transaction request that hits PayU's server (coming from any of our merchants), a unique reference ID is created and it is known as **mihpayid** (or PayU ID)

2	mode	This parameter describes the payment category by which the transaction was completed/attempted by the customer. The values are mentioned below: **Category used by Customer Value of Mode Parameter** Credit Card CC Debit Card DC NetBanking NB Cash Card CASH EMI EMI IVR IVR Cash On Delivery COD
3	status	This parameter gives the status of the transaction. Hence, the value of this parameter depends on whether the transaction was successful or not. You must map the order status using this parameter only. The values are as below: If the transaction is successful, the value of 'status' parameter would be '**success'**. The value of 'status' as '**failure'** or '**pending'** must be treated as a failed transaction only.
4	key	This parameter would contain the merchant key for the merchant's account at PayU. It would be the same as the key used while the transaction request is being posted from merchant's end to PayU.
5	txnid	This parameter would contain the transaction ID value posted by the merchant during the transaction request.
6	amount	This parameter would contain the original amount which was sent in the transaction request by the merchant.
7	discount	This parameter would contain the discount given to user - based on the type of offer applied by the merchant.
8	offer	This parameter would contain the offer key which was sent in the transaction request by the merchant.
9	productinfo	This parameter would contain the same value of **productinfo** which was sent in the transaction request from merchant's end to PayU

10	firstname	This parameter would contain the same value of **firstname** which was sent in the transaction request from merchant's end to PayU
11	lastname	This parameter would contain the same value of **lastname** which was sent in the transaction request from merchant's end to PayU
12	address1	This parameter would contain the same value of **address1** which was sent in the transaction request from merchant's end to PayU
13	address2	This parameter would contain the same value of **address2** which was sent in the transaction request from merchant's end to PayU
14	city	This parameter would contain the same value of **city** which was sent in the transaction request from merchant's end to PayU
15	state	This parameter would contain the same value of **state** which was sent in the transaction request from merchant's end to PayU
16	country	This parameter would contain the same value of **country** which was sent in the transaction request from merchant's end to PayU
17	zipcode	This parameter would contain the same value of **zipcode** which was sent in the transaction request from merchant's end to PayU
18	email	This parameter would contain the same value of **email** which was sent in the transaction request from merchant's end to PayU
19	phone	This parameter would contain the same value of **phone** which was sent in the transaction request from merchant's end to PayU
20	udf1	This parameter would contain the same value of **udf1** which was sent in the transaction request from merchant's end to PayU
21	udf2	This parameter would contain the same value of **udf2** which was sent in the transaction request from merchant's end to PayU
22	udf3	This parameter would contain the same value of **udf3** which was sent in the transaction request from merchant's end to PayU
23	udf4	This parameter would contain the same value of **udf4** which was sent in the transaction request from merchant's end to PayU

24	udf5	This parameter would contain the same value of **udf5** which was sent in the transaction request from merchant's end to PayU
25	hash	This parameter is absolutely crucial and is similar to the hash parameter used in the transaction request send by the merchant to PayU. PayU calculates the hash using a string of other parameters and returns to the merchant. The merchant must verify the hash and then only mark a transaction as success/failure. This is to make sure that the transaction hasn't been tampered with. The calculation is as below: **sha512(SALT|status|||||udf5|udf4|udf3|udf2|udf1|email|firstname|productinfo|amount|txnid|key)** The handling of udf1 – udf5 parameters remains similar to the hash calculation when the merchant sends it in the transaction request to PayU. If any of the udf (udf1-udf5) was posted in the transaction request, it must be taken in hash calculation also. If none of the udf parameters were posted in the transaction request, they should be left empty in the hash calculation too.
26	error	For the failed transactions, this parameter provides the reason of failure. Please note that the reason of failure depends upon the error codes provided by different banks and hence the detailing of error reason may differ from one transaction to another. The merchant can use this parameter to retrieve the reason of failure for a particular transaction.
27	bankcode	This parameter would contain the code indicating the payment option used for the transaction. For example, in Debit Card mode, there are different options like Visa Debit Card, Mastercard, Maestro etc. For each option, a unique bankcode exists. It would be returned in this bankcode parameter. For example, Visa Debit Card – **VISA,** Master Debit Card – **MAST.**
28	PG_TYPE	This parameter gives information on the payment gateway used for the transaction. For example, if SBI PG was used, it would contain the value **SBIPG.** If SBI Netbanking was used for the transaction, the value of PG_TYPE would be **SBINB**. Similarly, it would have a unique value for all different type of payment gateways.
29	bank_ref_num	For each **successful** transaction – this parameter would contain the **bank reference number** generated by the bank.