1. Download the aar from github and import it as a module dependency in an Android Studio Project:
https://github.com/payu-intrepos/Android-Custom-Browser/releases

2. For making payment using CC/DC flow

2.1 Create a form in android with following fields:

a) CC/DC number
b) Month and Year
c) Name on card
d) Expiry
e) Store card for one click payment (check box)

2.2 Webpage to be loaded in Merchant's webview

URL: https://secure.payu.in/_payment;
METHOD: *POST*
DATA:

pg=CC&device_type=1&key=obScKz&txnid=1446200455592&amount=10.0&productinfo=myproduct&firstname=firstname&email=me@itsmeonly.com&surl=https%3A%2F%2Fpayu.herokuapp.com%2Fsuccess&furl=https%3A%2F%2Fpayu.herokuapp.com%2Ffailure&hash=aad895dd35e61ae04ce8da9bea286e20c8d0fe31e3d11a5cfc0d48e300c3fca1157bc7d27fc736dac1002c22c86e981cfd4b44630db590a97fe2c00ad80b18a0&udf1=udf1&udf2=udf2&udf3=udf3&udf4=udf4&udf5=udf5&bankcode=CC&ccnum=5123456789012346&ccvv=123&ccexpyr=2019&ccexpmon=12&ccname=gshs&card_name=gshs&user_credentials=obScKz:payutest@payu.in&store_card=1&one_click_checkout=1

2.3 Description of post parameters:

a) **pg**=CC (Refer Appendix for possible pg values)
b) **key**=obScKz (Merchant key)
c) **txnid**=1446200455592 (ID for this transaction, given and generated by merchant)
d) **amount**=10.0
e) **productinfo**=myproduct
f) **firstname**=firstname (Username/Firstname)
g) **email**=me@itsmeonly.com (Email address of user)
h) **surl**=https%3A%2F%2Fpayu.herokuapp.com%2Fsuccess
(Success url on which the user will be redirected upon successful payment)
i) **furl**=https%3A%2F%2Fpayu.herokuapp.com%2Ffailure
(Failure url on which the user will be redirected upon payment failure)
j) **hash**=
aad895dd35e61ae04ce8da9bea286e20c8d0fe31e3d11a5cfc0d48e300c3fca1157bc7d27fc736dac1002c22c86e981cfd4b44630db590a97fe2c00ad80b18a0 (Hash, unique identifier which represents this particular transaction - mainly useful to avoid data forgery during payments, Refer Appendix for hash generation code)

k) **udf1**=udf1 (User defined params - can be empty string if nothing needs to be passed)

l) **udf2**=udf2 (User defined params - can be empty string if nothing needs to be passed)
m) **udf3**=udf3 (User defined params - can be empty string if nothing needs to be passed)
n) **udf4**=udf4 (User defined params - can be empty string if nothing needs to be passed)
o) **udf5**=udf5 (User defined params - can be empty string if nothing needs to be passed)
p) **bankcode**=CC (Refer Appendix for possible bankcode values)
q) **ccnum**=5123456789012346 (Card number of user, as taken from the form in step 2.1)
r) **ccvv**=123 (Card cvv, as taken from the form in step 2.1)
s) **ccexpyr**=2019 (Card expiry year, as taken from the form in step 2.1)
t) **ccexpmon**=12 (Card expiry month, as taken from the form in step 2.1)
u) **ccname**=gshs (Name on card, as taken from the form in step 2.1)
v) **card_name**=gshs (Name with which the card needs to be stored, as taken from the form in step 2.1)
w) **user_credentials**=obScKz:payutest@payu.in (Credential which identifies user at merchant's end. Can possibly use - merchantKey:email_id OR anything with which a merchant can identify this user)
x) **store_card**=1 (Set this parameter as 1)
y) **one_click_checkout**=1 (Set this parameter as 1)

2.4 Callback after successful transaction

- Merchant will get One click enabled stored card details in onActivityResult method, code would be as follows:

SAMPLE CODE:

```
@Override
protected void onActivityResult(int requestCode, int resultCode, final Intent data) {
if (requestCode == 100) {
        if(data != null ) {
                if(data.hasExtra("merchant_hash")){ // we have merchant hash, lets store it on merchant server.
                        storeMerchantHash(data.getStringExtra("merchant_hash"));
                }
        }else{
                Toast.makeText(this, "Could not receive data", Toast.LENGTH_LONG).show();
        }
}
}
```

- Now, storeMerchantHash is the method which will be implemented by merchant and will be responsible for storing one click payment hash on their server

SAMPLE CODE:

```
private void storeMerchantHash(String merchantHashData){
        JSONObject merchantHashObject = new JSONObject(merchantHashData);
        String merchantHash = merchantHashObject.getString(PayuConstants.MERCHANT_HASH);
        String cardToken = merchantHashObject.getString(PayuConstants.CARD_TOKEN);

        // MAKE AN API CALL HERE TO MERCHANT SERVER
        // STORE merchantHash and cardToken
}
```

3. For making payment through one click payment flow

3.1 Make API call to PayU server (No need to load in WebView) to get one click enabled card list

URL: https://info.payu.in/merchant/postservice.php?form=2

METHOD: POST

DATA:
key=obScKz&hash=62ca4a7ff8d456e799d70a2fa2fd1d3b863ec884610a332623c420194c0bf21261142ffc6a8987117e17c371b45c8f0097ec8365185887432efc1182af7e9b50&command=get_user_cards&var1=obScKz:payutest@payu.in

Params description:

key: Merchant key

hash: Webservice hash, refer appendix to understand how to generate it

command: API name, get_user_cards in this case

var1: User credentials which uniquely identity user for that merchant, possibly merchant_key:email_id

RETURN DATA: {"msg":"Cards fetched Successfully","status":1,"user_cards":{"8f6051ab2252fa25c7e91e19fb3dde195b4c0f3e":{"card_brand":"MASTERCARD","isDomestic":"Y","card_type":"MASTCC","card_token":"8f6051ab2252fa25c7e91e19fb3dde195b4c0f3e","expiry_year":"2018","is_expired":0,"card_no":"512345XXXXXX2346","card_cvv":1,"card_bin":"512345","card_mode":"CC","name_on_card":"gsgs","card_name":"gsgs","expiry_month":"12"}}}

3.2 Merchant has to make an API call to their server which will return all the one click payment enabled card details - card_token and merchant_hash as stored in #2.4 point number 2.

SAMPLE DATA TO BE RETURNED FROM MERCHANT SERVER:

[

{“card_token”:"f09a1ff70c55bccc6422ce5305afd0e2835e7fbc",”merchant_hash”:"zDWFJlS_tNsA"},

{“card_token”:"f09a1fe70c55bccc6422ce5305afd0e2835e7abc",”merchant_hash”:"zDWFJlS_tshf"},

]

3.3 Merchant has to populate a listview in android which will have option to choose payment via any of the one click enabled cards. This list will have intersection of cards returned by PayU server and Merchant server.

3.4 User can press the Pay button now and initiate the one click payment by loading following url in a webview with the params prescribed as follows

URL: https://secure.payu.in/_payment

METHOD: POST

DATA:

pg=CC&device_type=1&key=obScKz&txnid=1446201095154&amount=10.0&productinfo=myproduct&firstname=firstname&email=me@itsmeonly.com&surl=https%3A%2F%2Fpayu.herokuapp.com%2Fsuccess&furl=https%3A%2F%2Fpayu.herokuapp.com%2Ffailure&hash=e6e37912cd4b5e8d881db3620d0785b9f3b01da04cac20873685369969c55aa2585955d9ef2c0356ab59c4b58118f12fad11c9e1db916354ba921c81cd8f5f50&udf1=udf1&udf2=udf2&udf3=udf3&udf4=udf4&udf5=udf5&bankcode=CC&user_credentials=obScKz:payutest@payu.in&**store_card_token**=693813ffe60fc7e55e2c4529568fdd72ae3b35ac&**card_merchant_param**=lzDWFJlS_tNsA&ccexpmon=12&ccexpyr=2019&ccname=inmobile

*** Params marked in **bold** are the only ones which are new here*

Params Description:

store_card_token : Card token obtained from merchant server

card_merchant_param: Merchant hash obtained from merchant server

APPENDIX

1) **Payment hash generation code**
   sha512(key|txnid|amount|productinfo|firstname|email|udf1|udf2|udf3|udf 4|udf5|||||SALT)

2) **Webservice hash generation code**
   sha512(key|command|var1|salt)

3) **Possible bankcode values -** For CC/DC and One click payment options just pass **CC**

4) **Possible pg values -** For CC/DC and One click payment flow just pass **CC**