

3. Integrating CTSPaymentLayer

VikasCitruspay edited this page Dec 2, 2015 · 34 revisions

Payment layer lets you do the payments .

PG Payments using CC, DC and NB

You can pay using Credit, Debit, Netbanking see the following code:

Pages 7

Home

1. Integrating CTSAuthLayer

2. Integrating CTSProfileLayer

3. Integrating CTSPaymentLayer

4. Common Errors

5 Migration From V 3.0.x to V 3.1.x

ChangeLog

Clone this wiki locally

https://github.com/citruspay/open-ios

Clone in Desktop

```

//CC, DC payments
CTSElectronicCardUpdate *creditCard = [[CTSElectronicCardUpdate alloc] initWithCard:
creditCard.number = @"4028530052708001";
creditCard.expiryDate = @"03/2020"; //only mm/yyyy format
creditCard.scheme = [CTSUtility fetchCardSchemeForCardNumber:creditCard.number]; //fet
creditCard.ownerName = @"Tester"; // no special characters here
creditCard.cvv = @"123";

CTSPaymentDetailUpdate *paymentInfo = [[CTSPaymentDetailUpdate alloc] init];
[paymentInfo addCard:creditCard];

[CTSUtility requestBillAmount:@"10" billURL:BillUrl callback: ^(CTSBill *bill , NSError
    if(error){
        [UIUtility toastMessageOnScreen:error.localizedDescription];
    }
    else {
        [paymentLayer requestChargePayment:paymentInfo withContact:contactInfo wit
            if(error){
                [UIUtility toastMessageOnScreen:error.localizedDescription];
            }
            else {
                [UIUtility toastMessageOnScreen:[NSString stringWithFormat:@"Payme
            }
        }];
    }
}];

//netbanking payments
CTSPaymentDetailUpdate *paymentInfo = [[CTSPaymentDetailUpdate alloc] init];

// Update bank details for net banking payment.
CTSNetBankingUpdate* netBank = [[CTSNetBankingUpdate alloc] init];
netBank.code = @"CID001"; //you can obtain this from pgSetting service
netBank.name = @"Tester";
[paymentInfo addNetBanking:netBank];

[CTSUtility requestBillAmount:@"10" billURL:BillUrl callback: ^(CTSBill *bill , NSError

    if(error){
        [UIUtility toastMessageOnScreen:error.localizedDescription];
    }
    else {
        [paymentLayer requestChargePayment:paymentInfo withContact:contactInfo wit
            if(error){
                [UIUtility toastMessageOnScreen:error.localizedDescription];
            }
            else {
                [UIUtility toastMessageOnScreen:[NSString stringWithFormat:@"Payme
            }
        }];
    }
}];

```

Saved CC, DC Payments (A.K.A. Tokenized payments)

Obtain users saved cards and use this method to pay using saved card, Citrus server returns a token in each of the saved cards. You only need send this token and CVV(cvv is not saved along with the other card information) for doing the payment, rest of the information will be fetched by the Citrus server from its database using this token.

```

CTSPaymentDetailUpdate *tokenizedCardInfo = [[CTSPaymentDetailUpdate alloc] init];

// Update card for tokenized payment.
CTSElectronicCardUpdate *tokenizedCard = [[CTSElectronicCardUpdate alloc] initWithCreditCard:tokenizedCard];
tokenizedCard.cvv = @"123";
tokenizedCard.token = @"5115669e6129247a1e7a3599ea58e947";
[tokenizedCardInfo addCard:tokenizedCard];

[CTSUtility requestBillAmount:@"10" billURL:BillUrl callback:^(CTSBill *bill, NSError *error){
    if(error){
        [UIUtility toastMessageOnScreen:error.localizedDescription];
    }
    else {
        [paymentLayer requestChargeTokenizedPayment:tokenizedCardInfo withContact:contactInfo callback:^(CTSBill *bill, NSError *error){
            if(error){
                [UIUtility toastMessageOnScreen:error.localizedDescription];
            }
            else {
                [UIUtility toastMessageOnScreen:[NSString stringWithFormat:@"Payment of %d was successful", bill.amount]];
            }
        }];
    }
}];
}
}];

```

Loading Money into Users Citrus Prepaid Account

Money can be loaded into users citrus account using following method, this can be done using direct payments as well as saved cards

```

CTSPaymentDetailUpdate *creditCardInfo = [[CTSPaymentDetailUpdate alloc] init];

CTSElectronicCardUpdate *creditCard = [[CTSElectronicCardUpdate alloc] initWithCreditCard:creditCard];
creditCard.number = @"4028530052708001";
creditCard.expiryDate = @"03/2020"; //only mm/yyyy format
creditCard.scheme = [CTSUtility fetchCardSchemeForCardNumber:creditCard.number]; //fetch scheme
creditCard.ownerName = @"Tester"; // no special characters here
creditCard.cvv = @"123";

[creditCardInfo addCard:creditCard];

[paymentLayer requestLoadMoneyInCitrusPay:creditCardInfo withContact:contactInfo withPaymentMethod:PaymentMethodDirect callback:^(CTSBill *bill, NSError *error){
    if(error){
        //handle error
    }
    else{
        //handle success
    }
}];
}
}];

```

Paying via Prepaid account/Citrus Cash

Once there is some money in user's prepaid account it can be used to do payments. for that see the following method

```
[CTSUtility requestBillAmount:@"10" billURL:BillUrl callback: ^(CTSBill *bill , NSError
    if(error){
        [UIUtility toastMessageOnScreen:error.localizedDescription];
    }
    else {
        [paymentLayer requestChargeCitrusWalletWithContact:contactInfo address:addr
            if(error){
                //handle error
            }
            else{
                //handle success
            }
        }
    }
    }];
    }
    }];
    }];
```

Initiate Cashout Process into users Account from Citrus prepaid account

User can request to put money from Citrus cash account back into their bank account

```
CTSCashoutBankAccount *bankAccount = [[CTSCashoutBankAccount alloc] init];
bankAccount.owner = @"Tester";
bankAccount.branch = @"HSBC0000123";
bankAccount.number = @"123456789987654";

[paymentLayer requestCashoutToBank:bankAccount amount:@"5" completionHandler:^(CTSCash
    if(error){
        //handle error
    }
    else{
        //handle success
    }
}];
```

Fetch Available Schemes and Banks for the Merchant

Not all banks and card types are enabled for every merchant, they can be added, removed by contacting your sales contact at Citrus. To fetch what all is available for you use following method

```
[paymentLayer requestMerchantPgSettings:VanityUrl withCompletionHandler:^(CTSPgSetting
    if(error){
        //handle error
    }
    else {
        LogTrace(@" pgSettings %@", pgSettings);
        for (NSString* val in pgSettings.creditCard) {
            LogTrace(@"CC %@", val);
        }
        for (NSString* val in pgSettings.debitCard) {
            LogTrace(@"DC %@", val);
        }
        for (NSDictionary* arr in pgSettings.netBanking) {
            LogTrace(@"bankName %@", [arr valueForKey:@"bankName"]);
            LogTrace(@"issuerCode %@", [arr valueForKey:@"issuerCode"]);
        }
    }
}];
```

Fetch the PG Health

You can fetch the Health of the PG using following method

```
[paymentLayer requestGetPGHealthWithCompletionHandler:^(CTSPGHealthRes* pgHealthRes, NSError* error){
    if(error){
        //handle error
    }
    else{
        //handle success
    }
}];
```

Send money to another Citrus User

Money can be sent from one user's Citrus Wallet to another users wallet

```
[paymentLayer requestTransferMoneyTo:@"9800000000" amount:@"20" message:@"Here is Some Money" completion:^(CTSPGHealthRes* pgHealthRes, NSError* error){
    if(error){
        //handle error
    }
    else{
        //handle success
    }
}];
```

Dynamic Pricing, offer coupons, surcharge.

Using dynamic pricing you can apply surcharge or discount or even issue a coupon to the user.

Dynamic pricing happens in 2 stages first applying the rule and second is processing the payment

There are three ways in which rule can be applied

SearchAndApply

Search and apply will search for the possible rule depending upon the input data. The rule can be applied on the payment mode or card bin range or user details etc. The system will check for matching rule using the input data and apply the rule and return the altered amount if proper rule was found. Once the request is successful the user can make transaction for the altered amount. The input parameters will be transaction amount, payment details, userDetails , valid bill url .

```

//Payment details
CTSElectronicCardUpdate *creditCard = [[CTSElectronicCardUpdate alloc] initWithCreditCard:];
creditCard.number = @"4028530052708001";
creditCard.expiryDate = @"03/2020"; //only mm/yyyy format
creditCard.scheme = [CTSUtility fetchCardSchemeForCardNumber:creditCard.number]; //fet
creditCard.ownerName = @"Tester"; // no special characters here
creditCard.cvv = @"123";

CTSRuleInfo *ruleInfo = [[CTSRuleInfo alloc] init];
ruleInfo.originalAmount = @"100";
ruleInfo.operationType = DPRequestTypeSearchAndApply;

CTSPaymentDetailUpdate *paymentInfo = [[CTSPaymentDetailUpdate alloc] init];
[paymentInfo addCard:instrument];

CTSUser *user = [[CTSUser alloc] init];
user.email = @"test@gmail.com";
user.mobile = @"9700000000";

[paymentLayer requestPerformDynamicPricingRule:ruleInfo paymentInfo:paymentInfo billUr
    if(error){
        //handle error
    }
    else {
        //handle success
        [UIUtility toastMessageOnScreen: [NSString stringWithFormat:@"Request Status:
    }
}];

```

CalculatePricing

Calculate pricing will search for the given rule name and apply the rule, if rule with given name is found, it will return the altered amount. Once the request is successful the user can make transaction for altered amount. The input parameters will be transaction amount, payment details, user details and rule name, valid bill url

```

//payment details
CTSElectronicCardUpdate *creditCard = [[CTSElectronicCardUpdate alloc] initWithCreditCard:];
creditCard.number = @"4028530052708001";
creditCard.expiryDate = @"03/2020"; //only mm/yyyy format
creditCard.scheme = [CTSUtility fetchCardSchemeForCardNumber:creditCard.number]; //fet
creditCard.ownerName = @"Tester"; // no special characters here
creditCard.cvv = @"123";

CTSRuleInfo *ruleInfo = [[CTSRuleInfo alloc] init];
ruleInfo.ruleName = @"MonsoonSale";
ruleInfo.originalAmount = @"100";
ruleInfo.operationType = DPRequestTypeCalculate;

CTSPaymentDetailUpdate *paymentInfo = [[CTSPaymentDetailUpdate alloc] init];
[paymentInfo addCard:instrument];

CTSUser *user = [[CTSUser alloc] init];
user.email = @"test@gmail.com";
user.mobile = @"9700000000";

[paymentLayer requestPerformDynamicPricingRule:ruleInfo paymentInfo:paymentInfo billUr
    if(error){
        //handle error
    }
    else {
        //handle success
        [UIUtility toastMessageOnScreen: [NSString stringWithFormat:@"Request Status:
    }
}];

```

ValidateRule

Validate rule will be used to validate the whether the rule is properly applied or not. The input values will be transaction amount, payment details, user details, rule name and altered amount. System will search for the given rule and apply the rule and check whether the given altered amount matches with the generated altered amount. Once the rule is validated you can proceed for the transaction with altered amount.

```

CTSElectronicCardUpdate *creditCard = [[CTSElectronicCardUpdate alloc] initWithCreditCard:
creditCard.number = @"4028530052708001";
creditCard.expiryDate = @"03/2020"; //only mm/yyyy format
creditCard.scheme = [CTSUtility fetchCardSchemeForCardNumber:creditCard.number]; //fet
creditCard.ownerName = @"Tester"; // no special characters here
creditCard.cvv = @"123";

CTSRuleInfo *ruleInfo = [[CTSRuleInfo alloc] init];
ruleInfo.ruleName = @"50PrecentOff";
ruleInfo.alteredAmount = @"50";
ruleInfo.originalAmount = @"100";
ruleInfo.operationType = DPRequestTypeValidate;

CTSPaymentDetailUpdate *paymentInfo = [[CTSPaymentDetailUpdate alloc] init];
[paymentInfo addCard:instrument];

CTSUser *user = [[CTSUser alloc] init];
user.email = @"test@gmail.com";
user.mobile = @"9700000000";

[paymentLayer requestPerformDynamicPricingRule:ruleInfo paymentInfo:paymentInfo billUr
    if(error){
        //handle error
    }
    else {
        //handle success
        [UIUtility toastMessageOnScreen: [NSString stringWithFormat:@"Request Status:
    }
}];

```

□

Processing the payments for dynamic pricing

only after one of the above services is applied user can proceed with the actual payments.to do that use the following method.

```

contactInfo = [[CTSContactUpdate alloc] init];
contactInfo.firstName = @"first";
contactInfo.lastName = @"last";
contactInfo.email = @"Ytest@gmail.com";
contactInfo.mobile = @"9700000000";

addressInfo = [[CTSUserAddress alloc] init];
addressInfo.city = @"Mumbai";
addressInfo.country = @"India";
addressInfo.state = @"Maharashtra";
addressInfo.street1 = @"Golden Road";
addressInfo.street2 = @"Pink City";
addressInfo.zip = @"401209";

[paymentLayer requestChargeDynamicPricingContact:contactInfo withAddress:addressInfo c
    if(error){
        //handle error
    }
    else {
        //handle success
    }
}];

```


