I'm not a bot

The following table provides details of some of the test cards that you can use to test your integration in the sandbox (Test environment). As Sandbox is a replica of the production environment, you can push the code in production by just replacing account credentials and URLs.The character X in the card numbers are placeholders that can be substituted with any number (1-9). Any value can be used for the name parameter in Test environment. Payment FlowCard NumberNetworkExpiryCVVOTPPayU/Merchant Hosted Checkout5123456789012346Mastercard05/25123123456PayU/Merchant Hosted Checkout4012001037141112VISA05/25123123456Server-to-Server5497774415170603Mastercard05/25412123456PayU/Merchant Hosted Checkout6082015309577308RUPAY05/25123123456 Card NumberNetworkExpiryCVVOTP5118-7000-0000-0003Mastercard05/251231234564594-5380-5063-9999VISA05/25123123456 Any value can be used for the name parameter in Test environment. Payment Flow Card Number Expiry CVV OTP Mobile (mandatory for EMIs) Kotak Bank DC EMI 4706137805099594Note: Amount range is : 5000.00 to 10000.00 05/25 any random three-digit number 111111 9123412345 AXIS Bank DC EMI 4011510000000007 05/25 any random three-digit number 123456 8884758579 HDFC Bank CC EMI 4453341065876437 05/25 any random three-digit number 123456 9123412345 ICICI Bank CC EMI 4808557848741463 05/25 any random three-digit number 123456 9123412345 Onecard CC EMI 4010636236612108 05/25 any random three-digit number 123456 9123412345 Axis Bank CC EMI 5241784703665106 05/25 123 123456 9123412345 Standard Chartered CC EMI 5404608014083225 05/25 any random three-digit number 123456 9123412345 HSBC Bank CC EMI 4862696278807023 05/25 any random three-digit number 123456 9123412345 Kotak Bank CC EMI 4363888155006621 05/25 any random three-digit number 123456 9123412345 IndusInd Bank CC EMI 4029706572777150 05/25 any random three-digit number 123456 9123412345 Citibank CC EMI 4550387246273400 05/25 any random three-digit number 123456 9123412345 SBI CC EMI 5264686823451576 05/25 any random three-digit number 123456 9123412345 HSBC Bank CC EMI 4862696278807023 05/25 any random three-digit number 123456 9123412345Mob: 9999999999OTP = 123456PAN = XYZPA1234A (for eKYC)DOB = 21-07-1980Aadhaar = 9999 9999 1234Aadhaar OTP = 123456UPI ID - test@upi Any name can be used for the name parameter in Test environment.CurrencyCard NumberExpiryCVVOTPUSD $4755964453587236 12/2024596111111Euro 4020419926566936 12/2024041111111 Any value can be used for the name parameter in Test environment.Card NumberNetworkExpiryCVV5506900480000008Master Card05/2025123489537007734693 7VISA05/2025123 You can use anything@payu or 9999999999@payu as VPA to test your UPI-related integration in the sandbox (Test environment) only for the payment flows involving _payment AP. Use the following credentials for Net Banking in the sandbox:user name: payupassword: payuOTP: 123456The above Net Banking credentials must be used in API Playground. VendorMobile NumberOTPPayTM7777777777 or use card mentioned under Test Cards.888888AmazonYou can test using your original Amazon account details.AirtelYou can use your mobile number. BNPL ProviderbankcodeTenureMobile NumberCredit Card No.LazypayLAZYPAYNA9123412345NAHDFC BankHDFCF1515 days91234123454234567890056334HDFC BankHDFCF3030 days91234123454234567890056334HDFC BankHDFCF6060 days91234123454234567890056334HDFC BankHDFCF9090 days91234123454234567890056334Use Paytm Debit Card to make your everyday transactions simpler and more convenient. With its easy-to-use interface and seamless integration with the Paytm app, this debit card offers a range of benefits that can help you save time and money. Whether you want to shop online or pay bills, the Paytm Debit Card has got you covered. In this blog post, we will explore the various features and advantages of using the Paytm Debit Card, and why you should consider getting one for yourself. Step 1: Open paytm app - Ensure that you have the latest version of the Paytm app installed on your device. Open the Paytm app and log in to your account. Step 2: Select paytm wallet - Once you are logged in, select the Paytm Wallet option from the home screen. If you do not have a Paytm account, you will need to sign up for one first. Step 3: Add money to wallet - To use your Paytm debit card, you will need to add money to your Paytm wallet first. Click on the "Add Money" option and enter the amount you wish to add to your wallet. You can add money using your debit card, credit card, UPI, net banking or other payment methods. Step 4: Select 'prepaid, debit & credit cards' option - After adding money to your Paytm wallet, select the "Prepaid, Debit & Credit Cards" option from the home screen. This option allows you to view and manage all your linked debit and credit cards. Step 5: Enter paytm debit card details & pay - Click on the "Add New Card" option and enter your Paytm debit card details, including the card number, expiry date, CVV, and cardholder name. Once you have entered the details, click on the "Add Card" button to complete the process. You may be asked to enter an OTP or provide other authentication details before your Paytm debit card is added to your account. Step 1: Open paytm app - Open the Paytm app on your smartphone. If you haven't already, log in to your Paytm account. Step 2: Go to mobile recharge - Once you're logged in, tap on the "Mobile Recharge" icon. Choose the mobile operator and enter your mobile number. Enter the amount you want to recharge for. Step 3: Recharge the mobile - Tap on "Proceed to Recharge." Verify the recharge details and tap on "Pay Now." Step 4: Select the debit card option - On the payment page, select the "Debit Card" option. Choose the Paytm Debit Card as your payment option. Step 5: Enter paytm debit card details & pay - Enter the card number, expiry date, CVV, and the name on the card. Confirm the payment by entering the OTP received on your registered mobile number. Step 1: Open paytm app - Download the Paytm App from the App Store or Google Play Store. Log in to your Paytm account or create a new account if you don't have one already. Make sure you have an active Paytm Debit Card. Step 2: Go to movie tickets - Open the app and click on the "Movie Tickets" icon. Browse the available movies, select the showtime and cinema, and select the seats you want to book. Check the available cashback or discounts on movie tickets before booking. Make sure to select the correct movie, showtime, and cinema before proceeding with the booking. Step 3: Book the movie ticket - Enter the required details, including the number of tickets and the seat selection. Proceed to the payment options screen. Double-check the information you have entered before moving on to the payment page. Choose a showtime that fits your schedule and preferences. Step 4: Select debit card for payment - On the payment options screen, select the Paytm Debit Card option. Enter the required card details, including the card number, expiry date, and CVV code. Make sure your Paytm Debit Card has sufficient balance to cover the cost of the movie tickets. Check if your Paytm Debit Card is eligible for any discounts or cashback offers. Step 5: Enter paytm debit card details & pay - Double-check the card details you have entered and click on the "Pay" button. You will receive a confirmation message for your movie ticket booking. Keep a record of your ticket booking confirmation for future reference. Contact Paytm customer support if you encounter any issues with the payment process. Step 1: Open paytm app - Make sure you have the Paytm app downloaded on your smartphone. Open the app and log in to your account using your credentials. If you do not have a Paytm account, create one by following the instructions. Step 2: Go to electricity bills - On the homepage of the app, scroll down to the Electricity section. Click on it to open the bill payment page. Step 3: Enter bill details - Enter the details of the electricity bill you want to pay. Make sure to enter the correct consumer number and the amount to avoid any discrepancies. Step 4: Select 'postpaid, debit & credit cards' option - On the payment page, select the Postpaid, Debit & Credit Cards option. This will take you to a page where you can enter your card details. Step 5: Enter details & pay - Enter your Paytm Debit Card details, including the card number, expiry date, CVV, and name. Verify the details and click on Pay to complete the transaction. Using your Paytm Debit Card to pay your bills is a quick and hassle-free process. By following the above steps, you can easily make payments and avoid the hassle of standing in long queues. So, go ahead and enjoy the convenience of paying your bills with just a few taps on your smartphone! Make sure to check your account balance before making any payments to avoid any insufficient balance issues. Double-check the bill details before proceeding with the payment to ensure there are no errors. Keep your Paytm Debit Card details secure and do not share them with anyone to avoid fraudulent activities. Can I use my Paytm Debit Card for other bill payments as well? Yes, you can use your Paytm Debit Card for various bill payments such as mobile recharges, DTH, water bills, gas bills, etc. Is it safe to use my Paytm Debit Card for payments? Yes, it is safe to use your Paytm Debit Card for payments as Paytm uses advanced security measures to protect your card details. Can I save my Paytm Debit Card details for future payments? Yes, you can save your card details for future payments to save time. What should I do if I encounter any issues while making a payment? In case of any issues, you can contact Paytm customer support for assistance. Is there any fee charged for using the Paytm Debit Card? No, there are no additional fees charged for using the Paytm Debit Card for payments. Can I use my Paytm Debit Card for international payments? No, the Paytm Debit Card can only be used for domestic transactions. How long does it take for the payment to reflect in my account? The payment is usually reflected in your account within a few minutes of completing the transaction. Every bank account holder receives a debit card as part of their welcome kit, provided by the bank. However, before using the debit card, it must be activated. There are three activation methods available, and we will guide you through each one, providing the necessary steps. By following these methods, you can easily add and activate your debit card, ensuring seamless transactions quickly. A debit card can be activated in 3 ways. You may select any of the methods based on your needs and the resources available. It is important to note that, while the steps to be taken in each of these methods may differ depending on the banks policies and the process to activate a debit card, a few common steps remain the same. Step 1: Gently open the sealed envelope to obtain your four-digit PIN, which was provided to you by the bank along with your debit card. Step 2: Insert your debit card into the ATM. Step 3: Enter the debit card number and the ATM PIN. Step 4: You will then be prompted to create a new ATM PIN. Step 5: Follow the instructions on the machine to enter your new ATM PIN, and your card will be activated. Step 1: Log in to your banks internet banking portal.Step 2: Navigate to the Debit Card section of the portal.Step 3: Look for an option that says generate PIN or create PIN.Step 4: Select this option, and the portal will provide you with instructions on how to activate your debit card.Step 5: Follow the steps provided to activate your debit card online You must have received a phone banking PIN or other information that will assist you in phone banking conversations Dial the phone number provided to you and speak with an operational member to activate your debit card Once connected, follow all of the instructions given to you during the phone call to have your debit card activatedNote: Some banks send OTP to the customers registered mobile number for verification and activation of a debit card for added security. As a result, when activating your debit card, you must have your phone with the registered mobile number ready. Also Read: 5 Awesome Things to Know About Debit Cards Yes, it is necessary to generate a new PIN for your new debit card. It is important to destroy your old debit card before using the new one to prevent any potential fraud. There can be various reasons for a declined debit card transaction online. Some possible causes include entering incorrect information, insufficient funds in the account, multiple incorrect PIN entries, an expired card, or reaching the daily withdrawal limit. If you are facing issues, it is recommended to contact your bank for further assistance. While there is no specific time frame for activating a debit card, it is generally advisable to activate it as soon as you receive it to ensure its security and usability. If you have lost the PIN provided with your welcome kit, you should contact your bank or financial institution for guidance on how to retrieve or set a new PIN. If you did not receive a debit card when you opened your account, it is recommended to contact your bank or financial institution to inquire about the process for applying and obtaining a debit card. If you are experiencing difficulties using your debit card for online transactions, it is advisable to contact your bank or financial institution for assistance. They can help diagnose the issue and provide a solution. You will typically receive a confirmation message or email once your debit card is activated. Additionally, you can try making a small purchase or check your account balance and transaction history to confirm if your debit card is activated. If you have any doubts or need assistance with card activation, it is best to contact your bank or financial institution. Debit cards are not automatically activated. You will need to activate your card either through an ATM or by contacting the customer care service of your bank. Yes, you need to activate your new debit card before you can start using it for transactions. Failing to activate your debit card will prevent you from using it for various transactions such as ATM withdrawals or making purchases that require debit card information. Additionally, an unactivated card can pose a security risk if it is lost or stolen. Each set is randomly generated to simulate real data.Test data is actually the input given to a software program. It represents data that affects or is affected by the execution of the specific module. Some data may be used for positive testing, typically to verify that a given set of input to a given function produces an expected result. Other data may be used for negative testing to test the ability of the program to handle unusual, extreme, exceptional, or unexpected input. Poorly designed testing data may not test all possible test scenarios which will hamper the quality of the software.Testing is an iterative part of the development process that it performed to ensure the quality of the code. During the development process you will need fake data similar to real data for testing purposes.Generate Random Data AttributesThe following list of data will be auto generated:Credit Card Details, IBAN, Swift Bic Number, Account Number.Generate Visa card Master Card, American Express card, JCB card Discover card, Diners card, Voyager card, enRoute card, and credit card number quickly.Fake Payment Data Content ExamplescreditCardType : Generate a credit card type.// 'MasterCard', 'Visa'creditCardNumber : Generate a credit card number with a given type. Supported types are 'Visa', ' MasterCard', 'American Express', and 'Discover'.// '4556817762319090', '5151791946409422'// '4539710900519030', '4929494068680706'creditCardExpirationDate: Generate a credit card expiration date (DateTime).// DateTime: between now and +36 monthscreditCardExpirationDateString: Generate a credit card expiration date (string). By default, only valid dates are generated. Potentially invalid dates can be generated by using false as input. The string is formatted using m/y// '09/23', '06/21'// '01/18', '09/21'creditCardDetails : Generate an array with credit card details. By default, only valid expiration dates will be generated.// ['type' => 'Visa', 'number' => '4961616159985979', 'name' => 'Mr. Charley Greenfelder II', 'expirationDate' => '01/23']// ['type' => 'MasterCard', 'number' => '2720381993865020', 'name' => 'Dr. Ivy Gerhold Jr.', 'expirationDate' => '10/18']iban : Generate an IBAN string with a given country and bank code. By default, a random country and bank code will be used.// 'LI2690204NV3C0BINN164', 'NL56ETEE3836179630'// 'NL95ZOGL3572193597', 'NL76LTTM8016514526'swiftBicNumber: Generate a random SWIFT/BIC number string.// 'OGFCTX2GRGN', 'QFKVLJB7'What is Test Data? Why is it Important?Test data is actually the input given to a software program. It represents data that affects or is affected by the execution of the specific softwar feature. Some data may be used for positive testing, typically to verify that a given set of input to a given function produces an expected result. Other data may be used for negative testing to test the ability of the program to handle unusual, extreme, exceptional, or unexpected input.What's the benefit of a fake credit card maker?With no technical skills, You can create your online shop in a short time using integrated payment gateways. When you do this, you'll require fake credit card information to test. Online shop building tools are also unable to use real credit card numbers. Websites such as PayPal, Stripe, Simplify, etc., are each armed with their documents on credit card testing as well as dummy card numbers to test your knowledge.What are "valid fake card details?"While the data generated by this tool are entirely random, they are also subject to certain conditions and formulas. Payment tool testers check the fake numbers. However, they don't work in actual transactions.But they're not the real credit card. What does it mean to be valid is that they're generated using the same formula for numbers: the mod-10, or modulus 10 algorithm that creates an authentic credit card number.Test cardsYou can use the sample credit cards below to trigger different responses from our gateway. You can use them on test accounts but not on your live account.Real credit cards should never be used for testing, as per PCI-DSS compliance requirementsThe test cards do not have a card verification code and issue number.When using the cards, either through the API or HPP, you can enter any cardholder name, security code and future dated expiry.Test cards should not be used during live processing as these will be declined by the card networks and processing charges will occur.Test cards must pass the Luhn algorithm, also known as the MOD 10 check.(*) Any valid expiry date can be used but must be greater than the current month.For approval, it is recommended that you use "100" as the CVV value.(*) The CVV value can also be used to simulate various test responses.Software Testing MethodologiesBlack Box TestingBlack Box Testing is a software testing method that focuses on the functionality of a system without knowledge of its internal structure. Testers perform black box testing based on the specifications and requirements of the software, treating it as a black box. This approach allows testers to evaluate the systems inputs and outputs, making it particularly useful for validating the software against expected behavior. Equivalence partitioning, Boundary Value Analysis, and Cause Effect Graphing have commonly used test design techniques in black box testing. Equivalence partitioning involves dividing input data into classes to select representative test cases. Boundary Value Analysis focuses on testing the boundaries between these classes. Cause Effect Graphing identifies and tries different combinations of inputs and their corresponding outcomes. Black box testing is vital for uncovering defects in software by assessing its external behavior, and ensuring that it meets functional and non-functional requirements.White Box TestingWhite Box Testing is a software testing method that examines the internal structure, design, and implementation of the software being tested. Testers with knowledge of the systems inner workings can design test cases that target specific paths, branches, and data flows within the software. Control flow testing involves exercising different control paths within the software to ensure that all possible outcomes are adequately tested. Data flow testing focuses on data movement within the system and tests how data is modified and used throughout the software. Branch testing aims to test every decision point in the code, verifying that both true and false outcomes are correctly handled. Path testing explores all possible paths through the software to detect logical or functional errors. By understanding the inner workings of the system, white box testing can uncover issues related to code errors, missing functionality, or poor software design.Gray Box TestingGray Box Testing is a software testing approach combining elements of black box and white box testing methodologies. Testers conducting gray box testing need to gain more knowledge of the internal structure and design of the software. This allows them to better understand the system's inner workings than black box testers without possessing the full knowledge of white box testers. Gray box testing aims to balance validating the systems functionality and considering its internal implementation. Testers can design test cases based on their partial knowledge of the software to ensure that critical paths and potential issues are thoroughly tested. Gray box testing can be a practical approach when the internal details of the system are not fully available. Still, some insight into the system is necessary to design comprehensive test scenarios.Agile TestingAgile Testing is a software testing approach that aligns with the principles of agile software development. Agile methodology develops software incrementally and iteratively, focusing on delivering working software in short iterations or sprints. Agile testing embraces the collaborative nature of agile development and involves testers working closely with developers, product owners, and other stakeholders. Agile testing aims to ensure that software meets customer requirements, is of high quality, and can adapt to changing needs. Testers in agile teams contribute to defining user stories, creating acceptance criteria, and conducting continuous testing throughout the development process. They prioritize test cases based on business value and collaborate with the team to identify and fix defects promptly. Agile testing emphasizes frequent communication, feedback, and rapid delivery of tested increments, allowing teams to adapt and respond to changes efficiently.Ad Hoc TestingAd Hoc Testing is a software testing method where testers execute tests without predefined plans or documentation. Instead of following a structured approach, testers improvise and explore unscripted software, simulating real-world usage scenarios. Ad hoc testing is typically performed when there is limited time for formal testing or when exploring the softwares behavior in unconventional ways.Testers may vary their inputs, interact with the system unexpectedly, and assess its response. While ad hoc testing can uncover critical defects that might go unnoticed in formal testing, it has limitations. Due to its unstructured nature, reproducing and documenting discovered issues effectively can take time and effort. However, ad hoc testing can be valuable during early development stages or when dealing with time constraints, providing a quick way to gain insights into the softwares behavior and identifying immediate problems that require attention.Test your Payment Gateway Integration with our Test Credit Card Numbers.These cards will NOT work in PRODUCTION.These dummy card numbers with CVV as 123 and future expiration dates will work in Web Payment Software Payment Pages, Virtual Terminal, and Gateway API when set to test mode. These numbers will work with all major payment processors like Paytm, Razorpay, CCAvenue, Google Pay, PayUMoney, PayPal, 2Checkout, Sage Pay, USAePay, WorldRemit, etc.These credit card numbers DO NOT work! They are for testing purposes only. Use CVV as 123 and Future date as Expiry date. by Allan MacGregor When building e-commerce solutions or working with payment processing systems, its vital to test the different transaction scenarios to ensure that the payment processing is working as expected. Testing your system to make sure it can handle scenarios where payment transactions fail also helps protect your business and your customers alike. Payment transactions might fail due to a variety of reasons, and some of the most typical errors include the following: Do not honorInsufficient fundsStolen credit cardIdentity verification error You might be wondering, "How do I test these scenarios? Should I use a test card number or an actual card number? Although credit card companies have a reserved Bank Identification Number or BIN for test credit cards in an attempt to provide a quasi-universal standard of testing card numbers that can be used on any payment, not all payment gateways follow this convention. It's always recommended to check your gateway processor for details on the numbers and scenarios they support. Needless to say, test card numbers will only work on sandbox accounts, and fail on live payment gateways. Additionally, enabling sandbox mode will vary from payment processor to payment processor, so its recommended to consult their official documentation for more information. This article provides a list of test credit and debit card numbers that will work on most payment processors and elaborates on the specific scenarios that these numbers will simulate. Testing Credit and Debit Card Numbers While theres a wide variety of situations for testing your payment API, typically youll want to cover at least the following scenarios: Successful transactions.Payment errors like do not honor, insufficient funds, and stolen cards.Cardholder disputes.Testing 3D Secure The numbers and setup discussed in this article are specific to Rapyd payments, but the numbers should work on most APIs and the steps should also be much the same. hbspt.cta.load(5246303, '5fd0724d-a725-4f12-84ac-23227175b99a', {"useNewLoader":"true","region":"na1"}); Numbers for Successful Transactions A successful transaction is the most typical scenario that you might want to test. Testing a successful translation is useful to confirm that the final stages of your flow are working correctly, like confirmation pages and emails, order reviews, and so on. You can simulate a successful transaction using the following numbers, which will work on any payment processor. Card NumberCVVExp. DateAmountScenario4111111111111111AnyAny future dateAnySuccessful Transaction4462030000000000AnyAny future dateAnySuccessful Transaction Numbers for Error Transactions Knowing how your system will handle errors that lead to declined transactions is vital. Transaction errors occur as transactions are being processed, for example, during checkout in an e-commerce store, and typically fall into one of the following categories: Do not honor: Also called an invalid service code error, this indicates that the customer's issuing bank will not validate the transaction with the authorization code provided. There are several possible reasons for this error: The customer has a hold on their card.The card has had multiple denied payments in a row, so its temporarily locked.The merchant is operating in a different country than the customer.The transactions were flagged by the bank's anti-fraud system.Stolen card: This indicates that the card has been reported as stolen. If operating in a retail store, the recommended process is to retain the card, deny any goods or services, and immediately call the card issuer.Insufficient funds: The card doesn't have the necessary funds to complete the transaction. This is applicable to credit cards and debit cards. The numbers for simulating these transaction errors are as follows: Card NumberCVVExp. DateAmountScenario4111111111111105AnyAny future dateAnyDo not honor4111111111111143AnyAny future dateAnyStolen card4111111111111151AnyAny future dateAnyInsufficient funds In addition to the above numbers, you can also use a specific transaction amount to simulate the following errors: Payment gateway not responding: This error simulates the payment gateway not responding to the request due to an internal error.Payment expired: Simulates a scenario where the transaction cannot be processed because the payment has expired.Insufficient funds: Simulates an instance of the test credit card not having enough funds to complete the transaction.Processor unavailable: Simulates the payment processor not being available and the transaction failing as result of a network error.Settlement decline: Simulates the transaction failing due to an instance where the payment could not be processed because the settlement amount was declined. The following table contains the test numbers along with the transaction amount ranges you must enter to simulate these particular errors. Card NumberCVVExp. DateAmount (in cents)Scenario4111111111111111AnyAny future date2000 - 2009Settlement decline4111111111111111AnyAny future date2080 - 2089Processor unavailable4111111111111111AnyAny future date2120 - 2129Insufficient funds4111111111111111AnyAny future date2180 - 2189Payment expired4111111111111111AnyAny future date4000 - 4009Payment gateway not responding Numbers for Cardholder Disputes Cardholder disputes is another category of tests that you should consider running on your e-commerce or payment processing system. These occur when the initial transaction has been processed, and the cardholder is then disputing the transaction. Normally, the customer will open a dispute through their credit card company and the business will receive notification of the dispute and be offered a chance to refute the claim. Cardholder disputes typically fall under one of the following categories: Goods or services not provided: This error indicates that the customer has not received the goods or services they ordered, so theyre disputing the transaction with their credit card company.Subscription canceled: For SaaS companies, customers will often dispute subscription payments claiming the subscription had been canceled.Counterfeit merchandise: This simulates an instance of a fraudulent transaction where the customer claims the merchandise is counterfeit.Credit not processed: A common case for retailers is when customers claim to have a receipt for credit or a voided transaction, but have not yet had their credit processed.Cardholder dispute - miscellaneous: Simulates a generic dispute where the customer has a dispute with the credit card company, but no specific reason is attached. The numbers for simulating these cardholder disputes are as follows: Card NumberCVVExp. DateAmount (in cents)Scenario4539922288211219AnyAny future dateAnyGoods or services not provided4556028624137080AnyAny future dateAnySubscription canceled4556942399248726AnyAny future dateAnyCounterfeit merchandise5374500863109043AnyAny future dateAnyCredit not processed5132803130357186AnyAny future dateAnyCardholder dispute - miscellaneous Testing 3D Secure Another important aspect of credit card processing thats worth implementing and testing is 3D Secure, or 3DS. This is a feature that verifies the identity of the cardholder before allowing them to complete a transaction, with the aim of adding another layer of security to your payment processing system. Different payment processors implement 3DS as different products, for example: MasterCard SecureCodeVerified by VisaDiscover ProtectBuy With Rapyd, we can simulate 3DS authentication for the following Rapyd Payment API operations: Create payment: Set amount to 1000 or above, while avoiding the amounts listed in the 2000 range for error transactions.Create customer: This request will require 3DS for all payment methods.Add payment method to customer: This request will require 3DS for all payment methods. Let's go ahead with the API request to create a payment with a 3DS-enabled verification: Request URL: POST "amount": 1100, "currency": "USD", "payment_method": { "type": "us_visa_card", "fields": { "number": "4111111111111111", "expiration_month": "12", "expiration_year": "25", "cvv": "789", "name": "John Doe" } }, "capture": true} Once you receive the response, you can check the status, redirect_url, and next_action fields to validate the response and simulate the 3DS redirection flow. For example:{ "status": { "error_code": "", "status": "SUCCESS", "message": "", "response_code": "", "operation_id": "42416643-4159-49a8-8c65-4a17e8d4e489" }, "data": { "id": "payment_46bd13214fe6e7ab9f12f70e64d5743c", "amount": 0, "original_amount": 1050, "is_partial": false, "currency_code": "USD", "country_code": "US", "status": "ACT", // ... "payment_method": "card_761c1a5ce9b7af77f5f6cd539fb1fcbb", "payment_method_data": { "id": "card_761c1a5ce9b7af77f5f6cd539fb1fcbb", "type": "us_visa_card", "category": "card", // ... "next_action": "3d_verification" }, // ... "redirect_url": " // ... "next_action": "3d_verification", // ... },} Looking at the data section of the response, you can see that the next_action field is set to 3d_verification. This means that the payment processor has requested a 3DS authentication: status: The status of the payment is ACT (active). The payment is active and awaiting completion of 3DS authentication.redirect_url: The URL for 3DS authentication.next_action: The value is 3d_verification. 3DS authentication is required to complete the payment. As part of the testing, you can open the redirect_url in a new tab and complete the 3DS authentication. Once you open the URL in the browser, you should see the following. From this page, we can simulate the following: 3DS failure: By clicking the Cancel button.3DS success: By clicking the Continue button. Conclusion In this article, you learned the importance of testing your payment processing system regardless of whether youre a merchant, a SaaS company, or a retailer; testing will ensure your system can correctly process credit and debit cards and protect you against fraud. You also learned about some of the challenges involved with testing your payment processing system and the existence of test credit and debit card numbers. While every payment processor has their own set of test credit and debit card numbers, and the testing scenarios they support vary, most will support the scenarios we covered in this article. This article simulated these transactions and errors using Rapyd Payment API, a payment processing API that makes payment processing from a myriad payment methods, such as credit cards, debit cards, and e-wallets, fast and easy. After the integration is complete, you must test the integration before you go live and start collecting payment. You can start accepting actual payments from your customers once the test is successful.You can make test payments using one of the payment methods configured at the Checkout.The UPI in-app and UPI intent flow is not available in the Test mode.Things to remember while testing an integration:To test the integration make sure that you are making a transaction call to the test endpoint.Use your test key and salt for the transaction requests. For more information, refer to Access Test Key and Salt.Set the value of the environment parameters to 1. These test cards, UPI, and Wallet credentials must only be used in the sandbox environment. Using these test cards in production environment may cause validation error. Following are the payment methods supported in PayU Test mode. Card NumberExpiryCVVOTP5123456789012346O5/25123123456 Use the following credentials to test the Net Banking integration:user name: payupassword: payuOTP: 123456 The UPI in-app and UPI intent flow is not available in the Test mode.You can use either of the following VPAs to test your UPI-related integration:anything@payuFor Testing the UPI Collect flow, Please follow the below steps:-Once you enter the VPA click on the verify button and proceed to pay.In NPCI page timer will start, Don't "CLICK" on click text. Please wait on the NPCI page.The below link opens in the browser Paste the transaction ID at the end of the URL then click on the success/failure simulator page. After that, your app will redirect to your app with the transaction response. Txn_id>For AndroidYou can add the below metadata under the application tag in the manifest file to test the UPI Collect flow on test env:- // set the value to false for production environment //Comment in case of Production--> //Comment in case of Production--> You can use the following Debit and Credit cards to test EMI integration.Bank/Card TypeCard DetailsKotak DC EMICard Number: 4706-1378-0509-9594Expiry: any future date (mm/yy)CVV: 123OTP: 111111Name: Any nameMobile Number: 9123412345 (mandatory for EMI)AXIS DC EMICard Number: 4011-5100-0000-0007Expiry: any future date (mm/yy)CVV: 123OTP: 111111Name: Any nameMobile Number: 9123412345 (mandatory for EMI)HDFC CC EMICard Number: 4453-3410-65876437Expiry: any future date (mm/yy)CVV: 123OTP: 111111Name: Any nameMobile Number: 9123412345 (mandatory for EMI)ICICI CC EMICard Number: 4453-3410-65876437Expiry: any future date (mm/yy)CVV: 123OTP: 111111Name: Any nameMobile Number: 9123412345 (mandatory for EMI) You can use the following wallets and their corresponding credentials to test wallet integration. Wallet Mobile Number OTP PayTM 7777777777 888888 PhonePe Use the Phonepe Pre-Prod app for testing purposes as described in the following PhonePe doc. location: [(Download the app and register your mobile number and follow the instructions as described in the above PhonePe docs. NA AmazonPay You can test using your original Amazon account details.

Paytm card test. Paytm test debit card details. Paytm test card number. Paytm debit card test.