

Payment Systems Test Checklist

Background

This checklist should be used for testing any updates or new development for systems that handle payments or process money in any way.

Development environments should be set up to send payment requests to a sandboxed environment. PayPal provides a sandbox for testing purposes, and other services usually provide something similar.

Pre-testing checklist

Before testing any payments, always verify with the developer that the environment is set up to post to the sandbox and not to the live payment processing system. It is also important to always use test account numbers when available (e.g. for credit card and check payments). This is a fail-safe so that even if the payment is posted to the live site, the transaction will simply be declined instead of charging an actual account.

The test account numbers are also necessary to properly validate data in sandboxed environments. Many times, bogus numbers like "1234567890" will not validate, even in test environments. The test numbers are set up to match the validation requirements, but are reserved so they will never be associated with a real account.

Checklist

1. You should be able to pay with all available types of currency that are supported through the portal.
 - a. Check:
 - i. Personal
Test account numbers:
<https://www.wepay.com/developer/reference/testing>
 - ii. Company
 - iii. Cashier's
 - b. Credit/debit card:
 - i. Should be able to pay with all different types of CC (Visa, MC, AMEX, etc.)
 - ii. Test account numbers:
https://www.paypalobjects.com/en_US/vhelp/paypalmanager_help/credit_card_numbers.htm
 - c. Cash (if logging for cash is available through payment portal)
 - d. PayPal direct:
 - i. Note: you typically will not be able to complete a full payment this way, but if a payment system links to PayPal directly, we should at least confirm that the data is being received correctly by PayPal.
 - e. Other web payment services
2. You should be able to pay *partially* with "company credits," if applicable (e.g. SwagBucks).
3. You should be able to pay *fully* with "company credits."

4. You should not be able to charge more than is due (e.g. negative balance).
5. If applicable, you should be able to split total balance across different payment types.
6. Test with very long data in the submission. If possible, look at the database and how it is storing the data. Look for any limits on those database fields and see if you can exceed them without failing validation.
7. You should get distinct response messages for payment approved, payment declined, and error. Details for testing declined payments can be found below.
 - a. PayPal: https://developer.paypal.com/docs/classic/lifecycle/sb_error-conditions/#test-avs-errors
 - b. Authorize.net: http://developer.authorize.net/hello_world/testing_guide/