

Using the Test Site

Added by Trent, last edited by Trent on Aug 02, 2015

We've deployed a test source/institution called "Test Site" (with id = 100100) to our production servers that you can use to test various success and failure scenarios in a deterministic manner.

This single test site is used to test multiple workflows. The way the workflow is communicated to the test site is by using a particular password string in the account credentials. This is admittedly a little inconvenient. Someday we'd like to find a way to have distinct institutions for each test workflow to make selection as easy as locating an institution in on the "Connections" page by a particular name. The other inconvenience is that the password entry field obfuscates the text you type and it's very easy to mistype the magic string. Copying and pasting into this field is recommended for that reason.

One more caveat, and this is a change to the original behavior of the test site, is that most of the failure test cases let you go through the workflow only once. To do so again, you'll need to create a new connection. For most questions, the answer provided does not matter. It is the fact that the question has been answered at all that allows things to proceed. Later we may want to define certain answers as correct or incorrect, but that isn't essential to testing right now.

In general, failure cases work like this: When the test connection is first created, it will enter a "question" state. When the question is answered, it will be accepted and the fetch will complete.

The user-locked-out workflow, for example, can be tested in this way:

1. Create a new connection with username "user" and password "lockedoutfail". (This is the same code as before.)
2. Wait for the notification that the connection has entered the "question" state.
3. Pose the question text to the user, as usual. For this question type, the text will tell the user to visit his brand's site and have his account unlocked.
4. Make the FileThis API call to "answer" the question.
5. Observe that FileThis proceeds with the fetch, delivering our test documents.

The other informational pseudo-questions, user-action-required, user-must-set-up-account, account-not-supported, and account-not-paperless behave in the same way.

The two-letter-state workflow can be tested by answering by returning any of the state choices:

1. Create a new connection with username "user" and password "statefail". (Same code as before.)
2. Wait for the notification that the connection has entered the "question" state.
3. Pose the question dialog to the user, as usual. The tester can choose any state (eg. Arizona).
4. Make the FileThis API call to "answer" the question.
5. Observe that FileThis proceeds with the fetch, delivering our test documents.

The branch-code and pin questions behave in the same way. It doesn't matter what you enter for a response to those questions. Answering it will allow the fetch to succeed.

Note that there is no way to get connections back into the "question" state by updating the credentials, as was possible before. You can delete this connection and make a new one, however.

For whatever reason, the credentials-type questions behave differently, in that it **does** matter what your answer is. To have the fetch proceed, you must enter username "user" and password "pass". The mfa-challenge-question, one-time-auth-code cases also have correct and incorrect answers.

The following defines each test case in detail:

Credentials Invalid

The credentials for this site are username: 'user' and password: 'wrong'.

Two Letter State Invalid

The credentials for this site are username: 'user' and password: 'statefail'.

Branch Code Invalid

The credentials for this site are username: 'user' and password: 'branchfail'.

Pin Invalid

The credentials for this site are username: 'user' and password: 'pinfail'.

User Locked Out

The credentials for this site are username: 'user' and password: 'lockedoutfail'.

User Action Required

The credentials for this site are username: 'user' and password: 'useractionrequiredfail'.

User Must Set Up Account

The credentials for this site are username: 'user' and password: 'setupaccountfail'.

Account Not Supported

The credentials for this site are username: 'user' and password: 'notsupportedfail'.

Account Not paperless

The credentials for this site are username: 'user' and password: 'notpaperlessfail'.

Challenge Question

The credentials for this site are username: 'user' and password: 'mfaquestion'.

This source will always pose an mfa question: 'What is your mother's maiden name?'.

The correct answer is 'Smith'.

If answered incorrectly, this site will fail.

If answered correctly, this site will proceed and download documents.

One-time Auth Code

The credentials for this site are username: 'user' and password: 'onetimeauth'

This source will always ask for a One-time Auth Code, which is a non-persistent question. The job data will return two options to choose from. Once one option is selected the job will proceed to the next step, which is to ask for the code.

The correct code is '123456'.

If answered incorrectly, this site will fail.

If answered correctly, this site will proceed and download documents.

Deliver Bills

The credentials for this site are username: 'user' and password: 'bills'.

This source will always deliver documents with 'Bills' related metadata like 'amountDue' and 'dueDate'.

Deliver Statements

The credentials for this site are username: 'user' and password: 'statements'.

This source will always deliver documents with 'Statement' related metadata like 'accountBalance' and 'availableBalance' if possible.

Deliver Notices

The credentials for this site are username: 'user' and password: 'notices'.

This source will always deliver documents with 'Notice' related metadata like having 'currentDocument' as false and an actionDate if possible.

[Like](#) Be the first to like this

No labels