NetSuite SuiteCommerce Developer Exam

Study Guide: April 2018

Contents

About the NetSuite SuiteCommerce Developer Exam

Passing this SuiteCommerce Developer Exam certifies that you have the knowledge and skills needed to create, extend and implement NetSuite commerce applications. See Description of a Qualified Candidate below.

Fee: $250

Conditions:

- This will be a live proctored examination. Online proctored testing is also available.
- No written or online reference materials may be used during the exam.
- 120 minutes allotted to complete approximately 77 multiple-choice and matching questions

Description of a Qualified Candidate:

The clearly qualified candidate has 1-2 years of working with all of the relevant SuiteCommerce Technologies. They have functional knowledge in addition to development experience with NetSuite. They can architect solutions to customer requirements which may include workflow, scripting, or integration with an outside system. Additionally, a candidate may have 2-3 years of relevant web development experience.

This person has the knowledge and skills needed to create, extend and implement NetSuite commerce applications. This person has an awareness of secure, scalable, reliable and performant solutions to customize, extend and/or interact with NetSuite. This person has a basic knowledge of SuiteScript.

Knowledge of JavaScript, JSON, HTML, CSS and REST technologies as well as ecommerce fundamentals will also be helpful.

Maintaining Your Certification

There will be 2 ongoing requirements for maintaining your NetSuite Certified SuiteCommerce Developer title:

Annual Release Maintenance

NetSuite will publish a once per year unproctored “SuiteCommerce New Release Quiz” to validate awareness of new NetSuite features related to SuiteCommerce that have been released in the prior year. Information on availability of this New Release Quiz will be communicated in the following ways:

- Posted to the Announcements on the Certification webpage: www.netsuite.com/certification
- Posted to the closed LinkedIn NetSuite Certified SuiteCommerce Developer Group
- Emailed out to the email of record in the NetSuite Certification Status Search Tool

Refresh SuiteCommerce Developer Exam

NetSuite will rewrite the SuiteCommerce Developer Exam every 3-4 years. NetSuite Certified SuiteCommerce Developers will be required to retake and pass this Exam every 3-4 years to maintain their Certification.

Recommended Training and Background:

Recommended Training Courses:

- SuiteCommerce Advanced for Developers

Recommended Skill Level:

1-2 years of experience working with a range of SuiteCloud Technologies
2-3 years of relevant software development experience
Knowledge of JavaScript, SQL, SOAP, XML, JSON, and REST technologies helpful

Subject Areas Covered by the Test:

All exam questions are based upon SuiteCommerce Advanced Denali Release (Sept. 2015).

These are the broad subject areas covered on the Exam:

- SuiteCloud
- APIs and Services
- SuiteCommerce Advanced applications
- Account Configuration and Set Up
- Design Fundamentals

Below is each subject area broken out into more detail:

Topic	Objective
SuiteCloud	Identify the impact of execution context for scripts on the webstore.
	Identify how .ss and .ssp should be used for commerce applications.
	Identify the purpose of touchpoints in SSP applications.
	Identify the reason to use global libraries within SSP applications.
	Identify considerations of using custom records specific to the webstore.
	Given a customization use case, identify the appropriate custom field types and/or appropriate configuration.
	Identify implications of using saved searches and coded searches in the webstore.
	Identify how to create, install, and share various bundle types.
	Identify situations when scriptable cart should be used.
APIs and Services	Given a scenario about the Commerce API , identify the code snippet that will achieve the stated result.
	Given a commerce scenario, identify which API (Commerce, Item Search, and SuiteScript) should be used to achieve the stated result.
	Given a commerce scenario, identify the most performant code snippet that achieves the stated goals.
	Given a commerce scenario, identify the proper fieldset setup for the Item Search API.

	Given a commerce scenario, identify the proper search field setup for the Item Search API.
	Given a commerce scenario, identify the proper facet field setup for the Item Search API.
	Given a commerce scenario, select the correct input parameters for the item search API query that achieves the stated goal.
	Given SuiteScript API request and response objects, identify the method and its appropriate use.
	Determine how to manipulate and query records using the SuiteScript API.
	Given a CMS scenario about adding content areas to a template, identify the code snippet that achieves the stated result.
	Given a commerce scenario, identify why or how to include or exclude specific functionality from the SEO generator.
	Given a commerce scenario, identify the code snippet that correctly uses application cache.
	Given a commerce scenario, identify the code snippet that will set the appropriate CDN cache parameters.
SuiteCommerce Advanced applications	Identify the characteristics of a single-page application.
	Given a commerce scenario including third-party integration (e.g. Google Analytics and AdWords), determine the proper configuration.
	Given a scenario to modify or expose a new item field across the application, identify the process that should be used.
	Identify how to include additional data in services that return item and transaction information (e.g. live order and placed order)
	Identify how to configure the display of facets and item options.
	Identify techniques to create and/or customize a module.
	Identify techniques to create and/or customize a backend model.
	Identify how modules are structured including versioning, ns.package.json, and distro.json.
	Identify the functionality of backbone library primitives including Backbone.Router, Backbone.View, Backbone.Model, and Backbone.Collection.
	Identify the core functionality of the SCA framework including Backbone.CachedCollection, Backbone.CachedModel, NavigationHelper, and ApplicationSkeleton.
	Identify the core functionality of the composite pattern including Backbone.ChildView, Backbone.CollectionView, and Backbone.View.Extensions.
	Identify how to bootstrap data on initial page load including user specific and site data.
	Given requirements, identify the correct use of jQuery Promises.
	Identify how SASS should be used to support responsive design and/or reusable components.
	Identify the usage of translation files and translation functions in multi-language web sites.
	Identify how to expose data from views to templates.
	Identify templates that follow SCA best practices.

	Identify the development and deployment process that leverages the SCA Developer Tools.
	Identify the tasks executed during the build process.
Account Configuration and Set Up	Identify how to configure and troubleshoot item images.
	Identify the steps to configure website domains.
	Given a commerce scenario about site set-up, identify the transaction form that should be defined for a given business model (e.g., B2C, B2B).
	Given a commerce scenario, identify the proper item configuration (e.g., item pricing, stock behavior, related items, and correlated items).
	Identify the characteristics of different item types including the details of matrix items.
	Identify how to configure URL components.
	Identify how to configure shipping items.
	Identify how to configure payment methods
	Identify NetSuite functionality and recommended practices related to credit card data (PCI).
Design Fundamentals	Recognize transaction types and/or their place in a standard lead-to-cash and returns workflow.
	Recognize how inventory impacts the commerce experience (e.g., back ordering, drop shipping and locations).
	Determine how to troubleshoot and debug front-end and back-end of a commerce application (e.g. browser console, server execution logs, NetSuite debugger, SEO debugger, CDN cache and other cache).
	Determine how to troubleshoot and debug item display and site search results.
	Identify the SuiteCommerce support policy, versioning strategy, and release cycle.
	Determine how to optimize performance, scalability and reliability in the front-end and back-end of a commerce application (e.g. caching, SuiteScripts, fieldset size, page load time, and page size).
	Given a scenario, identify the securely or insecurely implemented code snippet (e.g., security risks detailed in OWASP Top 10).
	Identify considerations when working with different environments (e.g. differences in behavior, testing, customizations, settings, and preferences, production vs. non production, and different accounts)

Sample Questions:

1. A user event script is deployed to the customer record. In which two instances might it execute? (Choose 2)

   A. Register a customer in the web store.

   B. Search on customers using SuiteTalk

   C. Submit a sales order from the web store

   D. Import a customer record using the csv import tool

   E. Create a customer from a different user event script

(Correct answer: A, D)

2. A developer wants to configure one page to show CMS content. Which three CMS scope type filters are valid? (Choose 3)

```
A.  ----------------------------------------
    <div data-cms-area-filters="path"></div>
    ----------------------------------------
B   ----------------------------------------
    <div data-cms-area-filters="global"></div>
   ----------------------------------------
C.  ----------------------------------------
    <div data-cms-area-filters="all_pages"></div>
    ----------------------------------------
D.  ----------------------------------------
    <div data-cms-area-filters="page_type"></div>
    ----------------------------------------
E.  ----------------------------------------
    <div data-cms-area-filters="content_type"></div>
    ----------------------------------------
F.  ----------------------------------------
    <div data-cms-area-filters="current_type"></div>
    ----------------------------------------
```

(Correct answer: A, B, D)

3. What are two purposes of the distro.json file? (Choose 2)

    a. To specify which modules will be included in an application in the ns.package.json file, define the custom files to compile.
    b. To specify which modules are compiled in the SuiteCommerce Advanced application
    c. To specify which custom files are compiled in the SuiteCommerce Advanced application
    d. To specify which gulp tasks are run when compiling the SuiteCommerce Advanced application
    e. To specify which JavaScript files, within a module, are compiled in the SuiteCommerce Advanced application

(correct answer: A, B)

4. Which three statements are true about back end scripting best practices?
    a. Using the validateLine() function is more performant than putting all logic in the recalc() function
    b. Using the validateLine() function is less performant than putting all logic in the recalc() function
    c. Using scheduled scripts when possible is considered a performance best practice
    d. Deploying scripts to "all roles" prevents them from executing for shopping customers and is therefore a performance best practice
    e. Deploying scripts to "all employees" prevents them from executing for shopping customers and is therefore a performance best practice

(Correct answer: c, e)