

# Pass Guaranteed InsuranceSuite-Developer - Associate Certification - InsuranceSuite Developer - Mammoth Proctored Exam–The Best Vce Exam



Are you still feeling distressed for expensive learning materials? Are you still struggling with complicated and difficult explanations in textbooks? Do you still hesitate in numerous tutorial materials? InsuranceSuite-Developer study guide can help you to solve all these questions. InsuranceSuite-Developer certification training is compiled by many experts over many years according to the examination outline of the calendar year and industry trends. InsuranceSuite-Developer Study Guide not only apply to students, but also apply to office workers; not only apply to veterans in the workplace, but also apply to newly recruited newcomers. InsuranceSuite-Developer guide torrent uses a very simple and understandable language, to ensure that all people can read and understand.

Our InsuranceSuite-Developer exam braindump is revised and updated according to the change of the syllabus and the latest development situation in the theory and the practice. The InsuranceSuite-Developer exam torrent is compiled elaborately by the experienced professionals and of high quality. The contents of InsuranceSuite-Developer guide questions are easy to master and simplify the important information. It conveys more important information with less answers and questions, thus the learning is easy and efficient. The language is easy to be understood makes any learners have no obstacles to study and pass the InsuranceSuite-Developer Exam.

>> InsuranceSuite-Developer Vce Exam <<

## Valid Associate Certification - InsuranceSuite Developer - Mammoth Proctored Exam test answers, valid InsuranceSuite-Developer exam dumps

With the pass rate is 98.65% for InsuranceSuite-Developer learning materials, our product has gained popularity among candidates, the also send some thank letter for helping them pass the exam successfully. We have a professional team to research the latest information for InsuranceSuite-Developer exam materials, and we can ensure that InsuranceSuite-Developer Exam Dumps you receive are the latest one. What's more, InsuranceSuite-Developer exam dumps are high quality, and you can pass the exam just one time. We offer you free update for 365 days after purchasing, and our system will send the update version for InsuranceSuite-Developer exam dumps to you automatically.

## Guidewire Associate Certification - InsuranceSuite Developer - Mammoth Proctored Exam Sample Questions (Q131-Q136):

### NEW QUESTION # 131

A developer is creating an entity for home inspections that contains a field for the inspection date. Which configuration of the file name and the field name fulfills the requirement and follows best practices?

- A. HomeInspection\_Ext.etx, InspectionDate
- B. HomeInspection.Ext.eti, InspectionDate

- C. HomeInspection.etx, InspectionDate.Ext
- D. HomeInspection.eti, InspectionDate.Ext
- E. HomeInspection\_Ext.eti, InspectionDate.Ext

**Answer: E**

Explanation:

Guidewire's Metadata Naming Conventions are strictly enforced to ensure that customer code remains distinct from Guidewire's base product code, which is essential for seamless platform upgrades.

When creating a brand-new entity, the developer must use the .eti (Entity Interface) extension. Following Cloud Delivery Standards, the entity name itself must include the \_Ext suffix. Therefore, HomeInspection\_Ext.

eti is the correct file structure. Regarding the fields within that custom entity, Guidewire best practices recommend applying the \_Ext suffix to custom columns as well (Option B), even if the entity itself is custom.

This provides a consistent visual indicator in Gosu code that the developer is interacting with an extension rather than a base product element.

Option A and C use the .etx extension, which is reserved for extending existing base entities (e.g., adding a field to Claim). Option D is incorrect because it lacks the mandatory suffix on the entity name. Option E uses an invalid file naming format. Following the convention in Option B ensures the data model is compliant with Guidewire's automated quality gates.

### NEW QUESTION # 132

Succeed Insurance would like to count the number of High Priority Activities that are related to a Job. Which approach follows best practices to meet this requirement?

- A. `var count = 0; job.Activities.where(\act -> act.Priority == TC_HIGH) { count += 1 }; print(String.format( " Count of High Priority Activities: %s ", {count}))`
- B. `var count = job.Activities.countWhere(\act -> act.Priority == TC_HIGH); print(String.format( " Count of High Priority Activities: %s ", {count}))`
- C. `var actList = job.Activities.toList(); var count = actList.countWhere(\act -> act.Priority == TC_HIGH); print(String.format( " Count of High Priority Activities: %s ", {count}))`
- D. `var count = 0; for (act in job.Activities) { if (act.Priority == TC_HIGH) { count += 1 } }; print(String.format( " Count of High Priority Activities: %s ", {count}))`
- E. `var count = 0; job.Activities.each(\act -> act.Priority == TC_HIGH) { count += 1 }; print(String.format ( " Count of High Priority Activities: %s ", {count}))`

**Answer: B**

Explanation:

In Guidewire InsuranceSuite, when working with entity arrays (like `job.Activities`), the Gosu language provides a powerful set of Collection Enhancements that allow developers to perform operations on data sets with minimal code and maximum readability.

These enhancements are built on top of standard Java collections but offer a more functional programming approach.

Option C is the correct best practice because it utilizes the `countWhere` enhancement. This method takes a block (a lambda expression) that defines a predicate and returns an integer representing the count of elements that satisfy that condition. This approach is superior to manual iteration for several reasons. First, it is declarative; it tells the system what to do (count where priority is high) rather than how to do it (initialize a counter, loop, check if, increment). This reduces the likelihood of "boilerplate" coding errors, such as off-by-one errors or scope issues with the counter variable.

Comparing this to the other options: Option D uses a traditional for loop, which is functional but overly verbose and less idiomatic in Gosu. Option B is inefficient as it creates an entirely new List object in memory (`toList()`) before counting, which is unnecessary overhead. Option A incorrectly uses `each`, which is intended for side effects, and Option E incorrectly uses `where`, which is intended for filtering a subset of objects into a new collection, not for executing a counting block.

By using `countWhere` directly on the entity array, the developer writes cleaner, more maintainable code that aligns with the Gosu Coding Standards taught in the Advanced Gosu curriculum. This method ensures that the logic is encapsulated in a single, readable line, making it easier for other developers to understand the business intent of the logic.

### NEW QUESTION # 133

There is a requirement to add fields specific to Auto Rental Agencies. The additional fields required are; Auto Rent License, Offers Roadside Assistance, and Offers Insurance. Other fields will come from the existing ABCCompanyVendor entity.

For reference, the diagram below shows the ABCCompany subtype of the ABCContact entity:

□ How should this requirement be configured following best practices?

- A. Create three new fields to extend the existing ABCompany Vendor subtype
- B. Create ABAutoRentalAgency.Ext as a subtype of A B Company Vendor and add the three fields to the subtype
- C. Create ABAutoRentalAgency.Ext as a subtype of ABCompany and add the three fields to the subtype
- D. Create a custom entity ABAutoRentalAgency\_Ext and add the three fields to this entity

**Answer: B**

Explanation:

In the Guidewire Data Model, managing entity relationships through Subtyping is a core principle for maintaining a clean, performant, and logically structured database. According to the InsuranceSuite Developer Fundamentals course, subtyping should be used when a specific group of entities shares a common base but requires additional, unique attributes.

#### 1. Leveraging the Existing Hierarchy

The prompt specifies that "other fields will come from the existing ABCompanyVendor entity." In the Guidewire ContactManager (AB) data model, the hierarchy typically flows from ABCContact # ABCompany # ABCompanyVendor. By creating ABAutoRentalAgency\_Ext as a subtype of ABCompanyVendor (Option A), the new entity automatically inherits all properties from the vendor level (such as Tax ID or Vendor Number) and the company level (such as Name or Address). This maximizes code and metadata reuse.

#### 2. Why Subtyping is Better than Extension

If a developer were to follow Option D and add these three fields directly to ABCompanyVendor, every vendor in the system—including Law Firms, Doctors, and Repair Shops—would have fields for "Auto Rental License." This is known as "data model pollution." It makes the database tables wider than necessary and complicates the UI, as you would need complex "visible" expressions to hide these irrelevant fields for other vendor types.

By creating a specific subtype, Guidewire's Table-per-Subtype or Table-per-Hierarchy (depending on version and configuration) storage strategy ensures that these three specific fields are only relevant to Auto Rental Agency records. This keeps the data model logically distinct and allows for the use of Modal PCFs, where the UI automatically switches to display the correct fields based on the subtype of the contact being viewed.

Best Practice Summary: Use the \_Ext suffix for the new subtype to follow Cloud Delivery Standards and place it as deep in the existing hierarchy as possible to inherit the most relevant specialized fields.

### NEW QUESTION # 134

Which two are capabilities of the Guidewire Profiler? (Select two)

- A. Provide timing information of application calls to external services
- B. Track time spent in the web browser
- C. Measure network latency between the database server and application server
- D. Measure network latency between the browser and application server
- E. Track where time is spent in Guidewire application code

**Answer: A,E**

Explanation:

The Guidewire Profiler is an essential diagnostic tool used to capture and analyze performance data from the perspective of the application server. Its primary function is to help developers identify "hotspots"—areas of the code that consume excessive time or resources—during the execution of a specific transaction, such as a page load, a batch process, or a web service call.

According to the System Health & Quality curriculum, the first major capability of the Profiler is tracking time spent within Guidewire application code (Option A). When profiling is active, the tool records the execution time of Gosu methods, business rules, and even PCF expressions. It provides a hierarchical "stack trace" view, allowing developers to see exactly which function or rule is responsible for a delay. This is particularly useful for detecting inefficient loops or complex logic that may be slowing down the user experience.

The second key capability is providing timing information for external service calls (Option D). In a modern InsuranceSuite ecosystem, applications frequently communicate with external systems for credit scores, address validation, or payment processing. The Profiler monitors these "exit points" (such as SOAP or REST integrations) and records the duration of each call. By analyzing this data, a developer can determine if a performance issue is internal to the Guidewire application or if it is caused by a slow response from an external vendor's API.

It is important to note that the Profiler is a server-side tool. It does not measure browser-side rendering time (Option E) or network latency between the client and the server (Option C). While it provides metadata about database queries, its focus is on the application's execution of those queries rather than raw network latency (Option B). By focusing on internal code and external integrations, the Profiler gives developers a clear view of the application's functional performance.

### NEW QUESTION # 135

An InsuranceSuite implementation project is preparing for deployment to the Guidewire Cloud Platform. Which two Cloud Delivery Standards must be met before deployment? (Select two)

- A. The default system user su is configured as the second argument of the runWithNewBundle method
- B. New entities and new columns added to existing entities use a customer suffix such as \_Si
- C. Log files contain no PII (Personally Identifiable Information) as clear text
- D. There are no instances of single statements with multiple expansion operators(\*)
- E. GUnit tests must be combined into suites and executed in Studio prior to each code deployment

**Answer: B,C**

Explanation:

Moving to the Guidewire Cloud Platform (GWCP) introduces a set of mandatory "Cloud Delivery Standards" designed to ensure that customer implementations are secure, upgradeable, and performant. Two of the most critical pillars in these standards are Metadata Naming Conventions and Data Privacy/Security.

First, naming conventions (Option D) are essential for maintaining the "Upgrade-Safe" nature of the cloud.

In Guidewire Cloud, the base product is updated frequently. To prevent custom metadata (new entities or columns) from conflicting with future Guidewire base product releases, all extensions must use a unique customer suffix. While the standard example is \_Ext, insurers often use their specific company initials, such as \_Si for Succeed Insurance. This ensures that the custom data model remains distinct from the gw namespace.

Second, the Observability and Security standards strictly forbid the logging of Personally Identifiable Information (PII) in plain text (Option E). In the cloud, logs are aggregated and viewed via tools like CloudWatch or Kibana. If sensitive data like Social Security Numbers, Credit Card numbers, or personal addresses are logged as clear text, it constitutes a major security risk and a violation of compliance standards like GDPR or SOC2. Guidewire's automated Quality Gates will often block a deployment if it detects potential PII leakage in the Gosu logging code.

Options A and B are general development practices but not the primary delivery standards that block deployment. Option C is actually a security risk; hardcoding the "su" (Super User) in bundle management is generally discouraged in favor of more granular permission handling.

### NEW QUESTION # 136

.....

There has been fierce and intensified competition going on in the practice materials market. As the leading commodity of the exam, our InsuranceSuite-Developer practice materials have get pressing requirements and steady demand from exam candidates all the time. So our InsuranceSuite-Developer practice materials have active demands than others with high passing rate of 98 to 100 percent. We are one of the largest and the most confessional dealer of practice materials. That is why our InsuranceSuite-Developer practice materials outreach others greatly among substantial suppliers of the exam.

**InsuranceSuite-Developer Valid Braindumps Ebook:** <https://www.pass4surequiz.com/InsuranceSuite-Developer-exam-quiz.html>

As the collection and analysis of our InsuranceSuite-Developer exam materials are finished by our experienced and capable IT elite, If you are worried about your InsuranceSuite-Developer Valid Braindumps Ebook - Associate Certification - InsuranceSuite Developer - Mammoth Proctored Exam exam and you are not prepared so, now you don't need to take any stress about InsuranceSuite-Developer Valid Braindumps Ebook certification, We guarantee to our valued customers that Guidewire InsuranceSuite-Developer exam dumps will save you time and money, and you will pass your Guidewire InsuranceSuite-Developer exam, With the VCE, you can practice the questions as if you were sitting in the real InsuranceSuite-Developer exam.

You serialize complex types using a serialization helper, Using technology to create new services, As the collection and analysis of our InsuranceSuite-Developer exam materials are finished by our experienced and capable IT elite.

## **Hot InsuranceSuite-Developer Vce Exam | Efficient InsuranceSuite-Developer Valid Braindumps Ebook: Associate Certification - InsuranceSuite Developer - Mammoth Proctored Exam**

If you are worried about your Associate Certification - InsuranceSuite Developer - Mammoth Proctored Exam exam and you InsuranceSuite-Developer are not prepared so, now you don't need to take any stress about Guidewire Certified Associate certification, We guarantee to our valued customers that Guidewire InsuranceSuite-Developer exam dumps will save you time and money, and you will pass your Guidewire InsuranceSuite-Developer exam.

