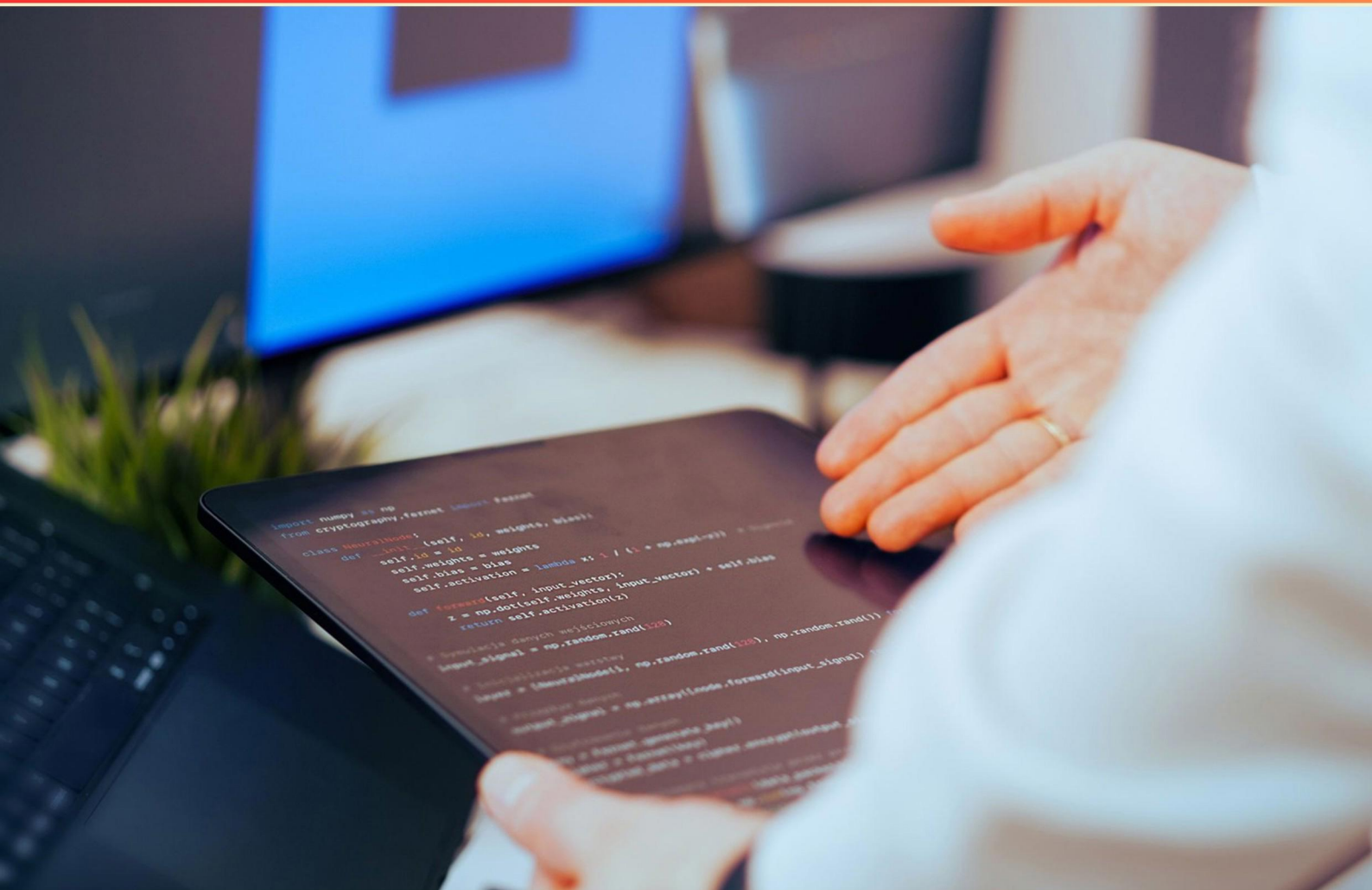


JUTRO DEVELOPER PRACTICE EXAM (SAMPLE)

STUDY GUIDE



SAMPLE STUDY GUIDE WITH LIMITED QUESTIONS

VISIT US ONLINE FOR
THE FULL VERSION STUDY GUIDE

<https://jutrodev.examzify.com>



Copyright © 2026 by Examzify - A Kaluba Technologies Inc. product.

ALL RIGHTS RESERVED.

No part of this book may be reproduced or transferred in any form or by any means, graphic, electronic, or mechanical, including photocopying, recording, web distribution, taping, or by any information storage retrieval system, without the written permission of the author.

Notice: Examzify makes every reasonable effort to obtain accurate, complete, and timely information about this product from reliable sources.

SAMPLE

Table of Contents

Copyright	1
Table of Contents	2
Introduction	3
How to Use This Guide	4
Questions	5
Answers	8
Explanations	10
Next Steps	16

SAMPLE

Introduction

Preparing for a certification exam can feel overwhelming, but with the right tools, it becomes an opportunity to build confidence, sharpen your skills, and move one step closer to your goals. At Examzify, we believe that effective exam preparation isn't just about memorization, it's about understanding the material, identifying knowledge gaps, and building the test-taking strategies that lead to success.

This guide was designed to help you do exactly that.

Whether you're preparing for a licensing exam, professional certification, or entry-level qualification, this book offers structured practice to reinforce key concepts. You'll find a wide range of multiple-choice questions, each followed by clear explanations to help you understand not just the right answer, but why it's correct.

The content in this guide is based on real-world exam objectives and aligned with the types of questions and topics commonly found on official tests. It's ideal for learners who want to:

- Practice answering questions under realistic conditions,
- Improve accuracy and speed,
- Review explanations to strengthen weak areas, and
- Approach the exam with greater confidence.

We recommend using this book not as a stand-alone study tool, but alongside other resources like flashcards, textbooks, or hands-on training. For best results, we recommend working through each question, reflecting on the explanation provided, and revisiting the topics that challenge you most.

Remember: successful test preparation isn't about getting every question right the first time, it's about learning from your mistakes and improving over time. Stay focused, trust the process, and know that every page you turn brings you closer to success.

Let's begin.

How to Use This Guide

This guide is designed to help you study more effectively and approach your exam with confidence. Whether you're reviewing for the first time or doing a final refresh, here's how to get the most out of your Examzify study guide:

1. Start with a Diagnostic Review

Skim through the questions to get a sense of what you know and what you need to focus on. Your goal is to identify knowledge gaps early.

2. Study in Short, Focused Sessions

Break your study time into manageable blocks (e.g. 30 – 45 minutes). Review a handful of questions, reflect on the explanations.

3. Learn from the Explanations

After answering a question, always read the explanation, even if you got it right. It reinforces key points, corrects misunderstandings, and teaches subtle distinctions between similar answers.

4. Track Your Progress

Use bookmarks or notes (if reading digitally) to mark difficult questions. Revisit these regularly and track improvements over time.

5. Simulate the Real Exam

Once you're comfortable, try taking a full set of questions without pausing. Set a timer and simulate test-day conditions to build confidence and time management skills.

6. Repeat and Review

Don't just study once, repetition builds retention. Re-attempt questions after a few days and revisit explanations to reinforce learning. Pair this guide with other Examzify tools like flashcards, and digital practice tests to strengthen your preparation across formats.

There's no single right way to study, but consistent, thoughtful effort always wins. Use this guide flexibly, adapt the tips above to fit your pace and learning style. You've got this!

Questions

SAMPLE

1. Which command is used to validate dependencies in a Jutro app?
 - A. npm run jutro validate:dependencies
 - B. npm run jutro validate:linters
 - C. npm run jutro validate:format
 - D. npm run jutro validate:folders
2. What are the benefits of Jutro app templates?
 - A. They are expensive and complex
 - B. They showcase specific lines of business
 - C. Accelerate application development and provide pre-designed components
 - D. They restrict creativity in app design
3. Which API is provided by the compositeSdk?
 - A. compositeRequest
 - B. aggregateRequest
 - C. mergeRequest
 - D. unifiedRequest
4. What feature does a configurable flow step provide in the Personal Auto FNOL template?
 - A. Direct integration with accounting
 - B. Flexibility in handling claims processes
 - C. Automatic reporting generation
 - D. Static user interface design
5. Where in the generated code is the PolicyCenter API function "createAccount" located?
 - A. generated/PcSdk/account/api/accountApi.ts
 - B. generated/Api/functions/createAccount.ts
 - C. src/account/api/accountApi.ts
 - D. generated/Account/CreateApi.ts

- 6. Why is the external package registry necessary?**
- A. To avoid external libraries
 - B. For security enhancements
 - C. To allow GWCP builds to use external package addresses without breaking safety protocols
 - D. To improve installation speed
- 7. How can you implement API calls in Jutro?**
- A. Using jQuery AJAX calls
 - B. Using Fetch API or Axios
 - C. Through plain XMLHttpRequest
 - D. By embedding the APIs directly in HTML
- 8. In which step should you begin adding new coverages to the Personal Auto Quote and Buy coverages page?**
- A. Build your project
 - B. Add new coverages in APD
 - C. Run the command to generate DigitalSDK
 - D. Sync the product model with PolicyCenter
- 9. What type of architecture does Jutro support for web applications?**
- A. Monolithic architecture
 - B. Microservices architecture
 - C. Layered architecture
 - D. Event-driven architecture
- 10. What steps are necessary to replace the logo in the app header with SucceedLogo.svg?**
- A. Add "logoSrc": "../images/SucceedLogo.svg" under header in floorplan.default
 - B. Copy the logo file to the assets directory
 - C. Add a line under subHeader for the new logo
 - D. Update the license agreement in App.config.tsx

Answers

SAMPLE

1. A
2. C
3. A
4. B
5. A
6. C
7. B
8. B
9. B
10. A

SAMPLE

Explanations

SAMPLE

1. Which command is used to validate dependencies in a Jutro app?

- A. npm run jutro validate:dependencies**
- B. npm run jutro validate:linters
- C. npm run jutro validate:format
- D. npm run jutro validate:folders

The command that is used to validate dependencies in a Jutro app is indeed the one that starts with "npm run jutro validate:dependencies." This command specifically targets the integrity and correctness of the project's dependencies, ensuring that all necessary packages are properly installed and configured. Dependency validation is crucial in maintaining the health of an application, as it helps to identify issues related to package versions, missing packages, or potential conflicts between dependencies. In the context of a Jutro application, running this command will check the dependencies defined in the project's configuration files, making it easier for developers to manage and maintain their project environments. This proactive approach can prevent issues down the line, such as runtime errors caused by inaccessible or incompatible packages. The other commands listed serve different purposes. For instance, validating linters pertains to ensuring coding style and standard practices are followed, while validating format deals with code formatting consistency. Validation of folders usually refers to checking the structure of the project's directories. Each command has its specific function, but when it comes to checking dependencies, the designated command addresses that need directly and effectively.

2. What are the benefits of Jutro app templates?

- A. They are expensive and complex
- B. They showcase specific lines of business
- C. Accelerate application development and provide pre-designed components**
- D. They restrict creativity in app design

The benefits of Jutro app templates primarily include the acceleration of application development and the provision of pre-designed components. This advantage stems from the fact that these templates are designed to streamline the development process by offering a starting point with ready-to-use elements. Developers can save significant time on design and coding by utilizing these components, allowing them to focus on customizing features and functionalities to meet specific user needs and business goals. Additionally, by implementing these templates, teams can maintain consistency in their applications, leading to a more cohesive user experience. This can enhance overall product quality as developers can leverage best practices embedded in the templates. Overall, the utilization of Jutro app templates not only speeds up the development cycle but also facilitates easier maintenance and updates, ensuring that applications remain relevant and functional over time.

3. Which API is provided by the compositeSdk?

- A. compositeRequest**
- B. aggregateRequest
- C. mergeRequest
- D. unifiedRequest

The compositeSdk provides the compositeRequest API, which is designed to handle multiple requests in a single call, allowing for efficient data aggregation from various sources. This capability is particularly useful in applications that require multiple pieces of data to be retrieved simultaneously without the overhead of making several individual requests. The compositeRequest API streamlines interactions between clients and services by allowing developers to define a complex operation in one go, enhancing performance and responsiveness. This functionality supports backend operations where data from various services needs to be quickly and effectively integrated. The other options do not represent actual APIs provided by the compositeSdk, highlighting the uniqueness and specificity of the compositeRequest as the correct choice in this context.

4. What feature does a configurable flow step provide in the Personal Auto FNOL template?

- A. Direct integration with accounting
- B. Flexibility in handling claims processes**
- C. Automatic reporting generation
- D. Static user interface design

A configurable flow step in the Personal Auto FNOL (First Notice of Loss) template offers significant flexibility in handling claims processes. This feature allows insurers to customize and adjust the steps in the claims process according to specific business needs, regulatory requirements, or customer preferences. By enabling users to adapt the workflow dynamically, the configurable flow step enhances efficiency and responsiveness. It allows teams to implement changes without needing extensive reprogramming, thereby improving agility in their operations. In the context of claims processing, this means that different types of claims or situations (such as varying customer circumstances or specific types of accidents) can be managed through tailored workflows. The other options, while relevant to management and business processes, do not capture the core aspect of flexibility that a configurable flow step emphasizes. Direct integration with accounting would focus more on financial aspects, automatic reporting generation pertains to reporting capabilities, and static user interface design would indicate a lack of adaptability in user interaction, which contrasts with the idea of configurability in operational steps.

5. Where in the generated code is the PolicyCenter API function "createAccount" located?

- A. generated/PcSdk/account/api/accountApi.ts**
- B. generated/Api/functions/createAccount.ts
- C. src/account/api/accountApi.ts
- D. generated/Account/CreateApi.ts

The correct choice indicates that the "createAccount" function is found in the file located at "generated/PcSdk/account/api/accountApi.ts". This suggests that the architecture of the application organizes its generated API functions into a structured hierarchy, specifically under a dedicated directory for the PolicyCenter SDK. Such a structure implies that the generated files are organized based on functionality, promoting better maintainability and ease of access for developers working with the PolicyCenter API. The designation of "PcSdk" in the path hints at a modular approach, allowing developers to easily identify the API files associated with the PolicyCenter software development kit. The other paths mentioned in the other choices do not adhere to the established naming conventions or folder structures expected in a generated environment for an API like PolicyCenter, which typically emphasizes organized files that reflect their purpose and module clearly. Hence, the selection of the correct file path aligns perfectly with the expected organization of API function files in the generated code base.

6. Why is the external package registry necessary?

- A. To avoid external libraries
- B. For security enhancements
- C. To allow GWCP builds to use external package addresses without breaking safety protocols**
- D. To improve installation speed

The external package registry is crucial because it facilitates the use of external package addresses in builds, specifically within the context of the Global Workers Core Platform (GWCP). This capability is fundamental for ensuring that the integration of such packages does not compromise the safety protocols established for software development. By using an external package registry, developers can access a wide range of libraries and tools while adhering to strict security measures that govern the use of external dependencies. This system allows for better management of external resources, ensuring that they are vetted and verified before being incorporated into projects. Consequently, this practice fosters a more secure development environment where external resources can be utilized efficiently without jeopardizing the integrity of the applications being built. The focus on maintaining safety within builds while still enabling flexibility and resourcefulness through external packages is what underlines the importance of an external package registry.

7. How can you implement API calls in Jutro?

- A. Using jQuery AJAX calls
- B. Using Fetch API or Axios**
- C. Through plain XMLHttpRequest
- D. By embedding the APIs directly in HTML

Implementing API calls in Jutro can be effectively achieved by utilizing the Fetch API or Axios. Both of these methods are modern approaches that facilitate the process of making asynchronous requests to APIs. The Fetch API is built into modern browsers and provides a straightforward way to fetch resources across the network, employing Promises to handle asynchronous operations. This allows for cleaner, more readable code and better error handling through the Promise-based syntax. Axios, on the other hand, is a popular library that simplifies API calls with additional features, such as automatic JSON data transformation and interceptors for handling requests and responses. Both of these options support modern JavaScript features and practices, making them suitable for Jutro, which likely emphasizes contemporary development practices. By using Fetch or Axios, developers can handle API interactions in a more efficient and manageable manner, which is essential for web applications today.

8. In which step should you begin adding new coverages to the Personal Auto Quote and Buy coverages page?

- A. Build your project
- B. Add new coverages in APD**
- C. Run the command to generate DigitalSDK
- D. Sync the product model with PolicyCenter

Adding new coverages to the Personal Auto Quote and Buy coverages page should start with the step of adding new coverages in the Application Programming Definition (APD). This step is crucial because APD serves as the technical specification that defines how the coverage will operate within the system. By initiating the process here, you ensure that the new coverages are accurately integrated within the application's framework. Once the new coverages are defined in APD, you can proceed to build the project, sync the product model to reflect those changes, and run any necessary commands to generate the required SDK components for the application. However, starting with APD is essential to lay a solid foundation for the subsequent steps, ensuring that everything aligns correctly within the software's architecture.

9. What type of architecture does Jutro support for web applications?

- A. Monolithic architecture
- B. Microservices architecture**
- C. Layered architecture
- D. Event-driven architecture

Jutro supports microservices architecture for web applications, which is characterized by breaking down an application into smaller, self-contained services that can be developed, deployed, and scaled independently. This approach enhances flexibility and allows teams to work on different components without causing disruptions to the entire application. The benefits of microservices include improved scalability, as individual services can be scaled based on demand, and increased resilience, since the failure of one service does not necessarily affect the whole system. This architecture aligns well with modern development practices that favor agile methodologies and DevOps, promoting faster release cycles and easier maintenance. In contrast, monolithic architecture typically involves building an application as a single unit, which can lead to challenges in scalability and maintainability as the application grows. Layered architecture organizes code into distinct layers, which can add complexity and may not offer the same level of flexibility as microservices. Event-driven architecture focuses on events and messaging for communication between services, but may not provide the structured, service-oriented approach offered by microservices.

10. What steps are necessary to replace the logo in the app header with SucceedLogo.svg?

- A. Add "logoSrc": "../images/SucceedLogo.svg" under header in floorplan.default**
- B. Copy the logo file to the assets directory
- C. Add a line under subHeader for the new logo
- D. Update the license agreement in App.config.tsx

To successfully replace the logo in the app header with SucceedLogo.svg, the necessary action involves specifying the correct source for the logo within the application code. By adding "logoSrc":

"../images/SucceedLogo.svg" under the header section in the floorplan.default file, you directly point the application to the new logo's location. This informs the app where to find the new SVG file, effectively replacing the existing logo in the header as intended. The other choices, while they may pertain to various aspects of application configuration, do not specifically address the task of changing the logo. Copying the logo file to the assets directory ensures that the file is physically present but does not directly assign it as the logo source in the header. Adding a line under the subHeader for the new logo does not align with the header configuration that controls the logo display. Similarly, updating the license agreement in App.config.tsx is unrelated to the visual representation of the logo in the app's UI. Thus, specifying the logo source in the correct location is the key step required to implement this change.

SAMPLE

Next Steps

Congratulations on reaching the final section of this guide. You've taken a meaningful step toward passing your certification exam and advancing your career.

As you continue preparing, remember that consistent practice, review, and self-reflection are key to success. Make time to revisit difficult topics, simulate exam conditions, and track your progress along the way.

If you need help, have suggestions, or want to share feedback, we'd love to hear from you. Reach out to our team at hello@examzify.com.

Or visit your dedicated course page for more study tools and resources:

<https://jutrodev.examzify.com>

We wish you the very best on your exam journey. You've got this!

SAMPLE