

Foundry

Certification Exam Guide:

Application Developer

Copyright © 2024
Palantir Technologies Inc.
All Rights Reserved

learn.palantir.com

Foundry Application Developer Certification

The Foundry Application Developer Certification Exam tests your ability to design and create operational, ontology-based applications using Palantir Foundry. The exam assesses knowledge of the applications in the Foundry suite as well as ability use them and integrate them with one another to develop real-world operational workflows for end users.

CANDIDATE PROFILE

A candidate for a Foundry Application Developer Certification is someone who knows how to use Ontology and application-building tools available in the Foundry suite. You should consider this certification if your job includes designing, building, or maintaining applications running on top of Foundry. The exam is recommended for people who have 6+ months of experience on the platform. General knowledge of TypeScript, user experience (UX), and database modeling best practices is recommended but not required.

EXAM DESIGN

The exam is composed of 60 questions covering a variety of topics within the selected certification track. It consists of the following question types:

- Multiple choice questions (select one, select multiple, discrete options)
- Hotspot questions (click on image) and matching questions (match one answer with another)

You will have 120 minutes to complete the exam, and a passing score of 70% is required to receive a certification. Questions will be presented in sequence without the ability to navigate back.



ABOUT THIS EXAM GUIDE

This exam guide is meant to provide insight into what will be tested on the certification exam. All study resources included here are meant to be directionally useful prep materials; test takers are encouraged to seek additional Foundry support resources to supplement knowledge. Questions may be asked on topics not explicitly included in the exam guide.

Application Developer Exam Content

The new Application Developer Certification Exam emphasizes testing of practical skills necessary to build and maintain custom user-facing applications on top of ontology. In addition to practical skills, it also tests general knowledge of platform capabilities and specific applications within the Foundry suite that are useful for custom application development. The exam is broken into 4 sections corresponding to 4 main responsibilities of application developers working on Foundry. Each section tests a variety of tasks described in greater detail on the following pages of this guide, with links to documentation and self-service training content to help you prepare for the exam.

APPLICATION DESIGN (30% of questions)

Can you design the ontology necessary to power custom applications based on business requirements? Can you choose between Workshop, Quiver, Slate, and Object Views to make trade-offs between development speed, maintainability, and user experience?

APPLICATION IMPLEMENTATION (30% of questions)

Do you know how to effectively use Workshop, Quiver, Functions, and Ontology Manager?

USER EXPERIENCE AND CROSS-APPLICATIONS WORKFLOWS (20% of questions)

Can you identify opportunities for embedding and redirecting between custom and built-in applications that improve user experience or maintainability? Can you identify situations where Scenarios and Foundry Rules can be used to let users interact with models and data pipelines?

APPLICATION MAINTENANCE (20% of questions)

Are you familiar with best practices of version control and release management? Can you identify and avoid issues related to data freshness and application performance? Do you know how to react when source data or users' requirements change?



Breakdown: Application Design

Can you design the ontology necessary to power custom applications based on business requirements? Can you choose between Workshop, Quiver, Slate, and Object Views to make trade-offs between development speed, maintainability, and user experience? Can you identify opportunities for turning static dashboards into useful operational workflows?

Task	Learning Resources
Choose appropriate Foundry applications for implementation of a given workflow	Documentation: <ul style="list-style-type: none"> Ontology-aware applications
Design ontology and pipelines based on business requirements	Documentation: <ul style="list-style-type: none"> Ontology What is a data pipeline? Self-service learning: <ul style="list-style-type: none"> Deep Dive: Creating your first Ontology
Identify opportunities for turning dashboards into operational workflows	Documentation: <ul style="list-style-type: none"> What is an operational application? Self-service learning: <ul style="list-style-type: none"> Deep Dive: Building your first Application
Optimize the UX of applications and cross-application workflows	Documentation: <ul style="list-style-type: none"> Carbon Self-service learning: <ul style="list-style-type: none"> Video: Workshop Design Principles Reference workflows: <ul style="list-style-type: none"> Configure layouts for optimal visual impact in Workshop



Breakdown: Application Implementation

Do you know how to effectively use Workshop, Quiver, Functions, and Ontology Manager?

Task	Learning Resources
Implement changes to ontology object types and link types	Documentation: <ul style="list-style-type: none"> Ontology Self-service learning: <ul style="list-style-type: none"> Deep Dive: Creating your first Ontology
Implement new ontology action types	Documentation: <ul style="list-style-type: none"> Actions
Implement typescript functions	Documentation: <ul style="list-style-type: none"> Development: Code repositories, Typescript basics Self-service learning: <ul style="list-style-type: none"> Video playlist: Functions
Implement interactive Quiver canvases	Documentation: <ul style="list-style-type: none"> Quiver: Overview, Analysis, Cards, Dashboards Self-service learning: <ul style="list-style-type: none"> Video playlist: Quiver
Implement workshop modules	Documentation: <ul style="list-style-type: none"> Workshop: Overview, Actions, Functions, Widgets Self-service learning: <ul style="list-style-type: none"> Deep Dive: Building your first Application Reference workflows: <ul style="list-style-type: none"> Create a dynamic inbox application for task triaging in Workshop Build a Common Operating Picture with Geospatial Data Create a guided data entry form in Workshop

Breakdown: UX and Cross-Application Workflows

Can you identify opportunities for embedding and redirecting between custom and built-in applications that improve user experience or maintainability? Can you identify situations where Scenarios and Foundry Rules can be used to let users interact with models and data pipelines?

Task	Learning Resources
Maximize user experience and collaboration by redirecting users to tools with desired functionality	Documentation: <ul style="list-style-type: none"> Object Explorer: Filter, Visualize, Save, Configure, Share URLs Integrations: Quiver Import/Export, Workshop Events, Notepad
Maximize maintainability by embedding applications within one another	Documentation: <ul style="list-style-type: none"> Object Views: Embedding Quiver, Embedding Workshop Workshop: Embedding Quiver, Embedding Object View Carbon: Modules Self-service learning: <ul style="list-style-type: none"> Video: Building a Ticket Framework in Foundry Reference workflows: <ul style="list-style-type: none"> Visualize time series in Workshop
Configure scenarios to embed models into an operational context	Documentation: <ul style="list-style-type: none"> Scenarios: Overview, Use Scenarios in Workshop Self-service learning: <ul style="list-style-type: none"> Video playlist: What-if analysis in Workshop
Configure Foundry Rules to allow users to customize pipeline logic	Documentation: <ul style="list-style-type: none"> Foundry Rules: Overview, Deployment, Customization Reference Workflows: <ul style="list-style-type: none"> Notify Alert Assignees Using Action Notifications



Breakdown: Application Maintenance

Are you familiar with best practices of version control and release management? Can you identify and avoid issues related to data freshness and application performance? Do you know how to react when source data or users' requirements change?

Task	Learning Resources
Use best practices related to release management	Documentation: <ul style="list-style-type: none"> Applications: Workshop, Quiver, Slate, Object Views, Notepad Ontology: Statuses, Functions
Ensure data used in applications is consistently fresh and correct	Documentation: <ul style="list-style-type: none"> Data Lineage: Overview, Explore, Find out-of-date datasets Data Health: Overview, Recommended health checks
Identify resources affected by Ontology changes and suggest possible migration strategies	Documentation: <ul style="list-style-type: none"> Ontology Metrics Data Lineage
Debug issues in applications	Documentation: <ul style="list-style-type: none"> Debugging: Functions, Quiver Graphs

Accessing self-service learning resources:

- Training courses are available on <https://learn.palantir.com/>
- Full platform documentation is available on <https://www.palantir.com/docs/>
- Video tutorials are available on the [Palantir Developers YouTube Channel](#)
- Example reference workflows are downloadable from <https://build.palantir.com/>

