Transact Qualified Platform Developer

Avoka Certified Exam Guide and Preparation

Exam name: *Transact 18.x Qualified Platform Developer*
Exam number: *PR000006*

Becoming Certified with Avoka

The Avoka certification program is a professional accreditation that Avoka offers to its employees, partners, and clients who demonstrate proficiency in basic and advanced topics relating to developing forms and applications using the Avoka Transact Platform. The Transact Platform Developer certification is a recognized standard proving that application developers have the knowledge and skills required to develop cutting-edge applications using the Avoka platform.

Becoming a qualified and certified resource is the first step towards engaging in Avoka client projects and ensuring exceptional customer experiences.

This guide will help you prepare for the Transact 18.x Qualified Platform Developer certification exam. It describes the knowledge and skills that are required to achieve a passing grade

.

Exam Details

- Number of questions: 70
- The Exam Time Limit: 90 minutes
- Passing Score: 70%
- Questions types are:
    - Multiple Choice
    - True or False
    - Multiple Selection

You are allowed a total of three attempts over a period of one year to pass the exam. If you require a second attempt, you will need to wait at least three days before attempting the exam again. If you require a third attempt, you will need to wait at least fourteen days before attempting the exam again.

Avoka's certification exams are delivered by Kryterion Global Testing Solutions. The exam is an on- line proctored format. For best results, be certain to review the Test Taker Guide, On Line Proctoring Series, provided by Kryterion in advance of the exam. Key considerations for an optimal On-Line Proctor test environment include:

- The room is well lit, quiet, and free from distractions.
- Headphones are not allowed during testing.
- Cell phones must be removed from the area.
- The testing surface is clean and clutter-free.
- Reading the exam aloud is not permitted.
- Interaction with another individual is not allowed during testing.
- You must remain in clear view of the camera during testing.
- Engage your On-Line Proctor as needed for any additional questions during the exam.

Registering for the Exam

Avoka's certification exams are delivered by Kryterion Global Testing Solutions. The *Transact 18.x Qualified Platform Developer* is a proctored exam and is only delivered online. You will need to register with Kryterion to take the exam.

To book an exam, go to https://www.webassessor.com/avoka/index.html and login or create an account. Once you are logged in, you will be able to schedule an exam.

The Avoka Certification FAQ.pdf provides step-by-step instructions for registering for the exam. You can find the FAQ guide on our partner portal at this location: https://community.avoka.com/training/f/certification.

Exam details:

- Exam Name: Transact 18.x Qualified Platform Developer
- Exam Number: PR000006
- Exam Cost: 175.00 USD

Exam Topics

The exam measures several competencies related to designing, developing and deploying a applications built on the Transact Platform. There are seven main topic areas:

1. Writing Services for Transact Manager
2. Describing the Development Process for Transact Manager Applications
3. Creating Collaboration Jobs
4. Describing General Features of Transact Manager
5. Identifying Security Features in Transact Manager
6. Describing Services in Transact Manager
7. Using the Avoka Exchange Catalog

Within each topic, there are specific subtopics that describe typical tasks that a platform developer would need to perform:

Topics	Subtopics and Tasks
Writing Services for Transact Manager	You should be able to: • Describe the Transact REST API • Use the Fluent Groovy Programming Language • Manage and Manipulate Data through Code • Develop Applications with the Transact SDK • Log and Debug Services
Describing the Development Process for Transact Manager Applications	You should be able to: • Create Projects, Applications, and Services with the Transact SDK • Work with Version and Source Control Systems • Write Unit Tests for Services • Document your Code
Creating Collaboration Jobs	You should be able to: • Describe Collaboration Jobs on the Avoka Platform • Build a Collaboration Job that chains several Forms together • Build a Review and Approve Collaboration Job
Describing General Features of Transact Manager	You should be able to: • Work with the out-of-the-box features and settings provided by the Transact Manager Platform to populate Forms and deliver Transactions • Describe, at a high level, the architecture of Transact Manager

Topics	Subtopics and Tasks
Identifying Security Features in Transact Manager	You should be able to: • Use Content Security Policies • Describe Transact Manager Security Manager options • Describe the differences between Groovy Services and Fluent Groovy Services
Describing Services in Transact Manager	You should be able to: • Describe the Transact Application Framework • Describe when Services are called during the lifecycle of an Application • Identify the diverse types of Services available in the Transact Manager Platform • Set Service properties
Using the Avoka Exchange Catalog	You should be able to: • Describe, at a high-level, the contents of an Exchange Component • Identify the types and use cases for the Avoka Exchange Components

Distribution Weight of Exam Topics

Each exam question carries the same weight, and there is no partial credit given for any question – all questions have the same value. The questions for the exam are distributed as follows:

Topics	Percentage Range of Exam
Writing Services for Transact Manager	30 - 40%
Describing the Development Process for Transact Manager Applications	10 - 20%
Creating Collaboration Jobs	10 - 20%
Describing General Features of Transact Manager	10 - 20%
Identifying Security Features in Transact Manager	10 - 20%
Describing Services in Transact Manager	10 - 20%
Using the Avoka Exchange Catalog	5 - 10%

Description of the Transact Platform Developer Role

This exam is designed for individuals who are Java developers and are expanding their skill set to include developing solutions on the Transact Manager platform, or for individuals that have experience performing the tasks and functions normally associated with a Transact Manager Services Developer role.

You are expected to have a minimum level of knowledge, skills, and abilities commonly associated with the services developer role before attempting the exam.

As a Transact Manager Services Developer, you should be able to perform the following tasks:

- Design and call services based on standard web architecture protocols such as REST
- Write services using an API
- Write unit and integration test using common frameworks for Java programming
- Debug applications using log files and the inherent debugging features found in most IDE applications
- Implement a source control process for application assets
- Select the appropriate service type and service template based on the Transaction Processing Sequence and the required use case
- Generate or modify JSON and XML data files using standard Java objects
- Set organizational and form level properties to accomplish specific application requirements
- Architect services to increase reusability and lower maintenance
- Work with form developers and identify integration points for an application
- Deploy and configure Avoka Exchange components
- Read and create JAVA docs

Typically, a candidate attempting this exam would have the following work experience in addition to having worked with the Avoka Transact platform.

- Experience with Object Oriented analysis, development and design using Java and/or Groovy
- Experience with developing in an IDE (for example Eclipse or IntelliJ)
- Experience with continuous integration as part of the development cycle
- Experience with unit and integration testing
- Experience creating services for the Java platform
- Knowledge of Agile delivery methodology
- Knowledge of server-side debugging and optimization techniques
- Strong understanding of HTML5 and CSS, CSS-Less (optional)
- Knowledge of creating secure communication channels, credentials, certificates, and keystores

Preparing for the Exam

Avoka Training

Avoka offers Transact Manager for developers training onsite or virtually. Training is not mandatory for certification but highly recommended. Keep in mind that training, on its own, will not provide you with the knowledge and skills needed to pass the exam successfully. It is only one step in the journey to certification. Self-study and successful hands-on experience are needed to gain the knowledge and experience needed to pass the exam.

Additional Resource

In addition to training, the Avoka Community website is an excellent resource. Make sure you sign-up to get the most out of the Avoka Community. Don't have an account? You can always sign-up on the landing page. There you will find the:

- Transact Fluent SDK Documentation
- Transact Services Guide
- Fluent API Java Docs
- Core API Java Docs (Note: The Core APIs are not covered in the exam.)

Self-Assessment Quiz

Use the following table to review the exam topics and assess your readiness to take the exam.

Topics and Subtopics	Yes	No
Writing Services for Transact Manager		
I know the common REST APIs available on the Transact Manager platform and the typical scenario for their use.		
I understand the syntax of the Groovy programming languages as well as the nuances that the language incorporates to allow for more efficient development.		
I can create and modify JSON and XML data using Groovy and the Fluent API classes.		
I know the use cases for most of the Fluent API top-level classes		
I can create and deploy services using an IDE		
I can use the Groovy service console		
I can use the Transact Platform log files, and IDE debug features to troubleshoot my application		
Describing the Development Process in Transact Manager		
I can create and modify Ant scripts to deploy a service		
I can describe the architecture of a Fluent SDK project		
I can explain the importance of source control management systems and how they can be used with the Transact Manager platform		
I can write integration and unit tests for my services using frameworks like JUnit and Mockito		
I can document my code in such a manner that the Java Docs are generated automatically		
Creating Collaboration Jobs in Transact Manager		
I can describe the types of collaboration jobs supported by Transact Manager		
I can describe the anatomy of a Collaboration Job		
I can create a Collaboration job that chains several forms together		
I can create a Collaboration job that can be used to place a form submission into a queue		
I can extend the functionality of a Collaboration job using Job Actions		
Describing Core Features of Transact Manager		
I can add properties to a form or an organization and explain the benefits and use cases for those properties		
I can describe, at a high-level, the architecture of the Transact Platform		

Identifying Security Features in Transact Manager		
I can describe the purpose of the Content Security Policy and how to modify using Transact Manager		
I can describe the Transact Platform's security manager and authentication options		
Describing Services in Transact Manager		
I can describe the Transact Manager processing sequence		
I can create the appropriate service type that leverages the Transact Manager processing sequence		
I can identify and extend service properties		
Using the Avoka Exchange Catalog		
I can identify the functionality provided by the various Exchange components		
I can describe what is included in an Exchange component		

Sample Test Questions

Below are some examples of the types of questions you will find on the exam.

Questions

Question 1:
To map a value from an authenticated user into your form fields, which feature of the Transact Manager platform would be most appropriate to use?

- Transaction Function called using a Form Open trigger
- Property Prefill Mapping
- Render Offline Submission Form service
- Organization Property

Question 2:
What transaction assets are available to the REST Delivery API on retrieval of a completed submission?
Select 3 that apply:

- Form XML
- Form Data Schema
- PDF Receipt
- Attachments
- Transaction History

Question 3:
True or False: A Transact Function using the Form Open trigger fires after the Tracking Number service in the Transaction Lifecycle?

- True
- False

Question 4:
Which of the following are valid Job Action Types you can use in a Collaboration Job?
Select 4 that apply.

- Job Delivery
- Job Form Start
- Job Form Stop
- Job Task Wait
- Job Process Message

Question 5:
What will be the outcome of running the following snippet of Groovy code in the Groovy console?

```
def x = 5 + 1;
def X = 6 + 1;
println(x);
```

- 7 will be printed because Groovy is not case sensitive like Java and uppercase X was declared after lowercase x
- Null because x is not declared as an integer and the expression is not in quotes
- 6 will be printed because Groovy is case sensitive
- Null because the expression is not wrapped in closures

Answers to the above Sample Questions:

Question 1:
To map a value from an authenticated user into your form fields, which feature of the Transact Manager platform would be most appropriate to use?

- Correct Answer: (b) Property Prefill Mapping

Question 2:
What transaction assets are available to the REST Delivery API on retrieval of a completed submission?
Select 3 that apply:

- Correct Answer: (a) Form XML, (c) PDF Receipt, (d) Attachments

---

Question 3:
True or False: A Transact Function using the Form Open trigger fires after the Tracking Number service in the Transaction Lifecycle?

- Correct Answer: (b) False

Question 4:
Which of the following are valid Job Action Types you can use in a Collaboration Job?
Select 4 that apply.

- Correct Answer: (a) Job Delivery, (b) Job Form Start, (d) Job Task Wait, (e) Job Process Message

---

Question 5:
What will be the outcome of running the following snippet of Groovy code in the Groovy console?

```
def x = 5 + 1;
def X = 6 + 1;
println(x);
```

- Correct Answer: (c) 6 will be printed because Groovy is case sensitive

---