braze learning

Braze Certified Developer



Exam Level: 2

Exam Cost: $100

Exam Time: 90 minutes

Passing Score: 74%

# of Questions: 50 multiple-choice

Earning the Braze Developer digital credential validates that individuals have experience in Braze with:

- Braze Technical Basics
- SDK Integration Basics
- Messaging Integration Basics
- Testing and Troubleshooting

Who should take this exam?

The Braze Certified Developer knows how to integrate Braze into their organization's tech stack. They understand Braze SDK integration, data ingestion methods, and message customization. Additionally, they regularly troubleshoot issues in their daily work and are familiar with Braze developer use cases.

Common Job Titles include:

- Mobile Engineer
- Product Manager
- Solutions Architect
- Backend Engineer
- Solutions Engineer
- Data Analyst
- Technical Project Manager
- Technical Marketer

How is this exam administered?

This is a 90 minute exam that uses AI-assisted proctoring in a secure browser. To sign up for the exam, visit braze.com/certification

Recommended Prerequisites:

Before taking this exam, we recommend you have:

- Experience integrating the Braze SDK
- A deep understanding of data ingestion methods and message customization
- The ability to regularly troubleshoot common Braze SDK issues
- Experience with Braze developer use cases

Recommended Braze Learning:

Take the **Braze Developer** Learning Path, where you'll learn how to integrate Braze into your organization's tech stack. These courses cover SDK integration, data ingestion methods, partner integrations, and troubleshooting for your data flows. You'll gain the tools you need to engineer data ingestion efficiently and automate processes for your marketing teams.

Braze Certified Developer Study Guide

Exam Content

DOMAIN	% OF EXAMINATION
1.0 Braze Technical Basics	30%
2.0 SDK Integration Basics	35%
3.0 Messaging Integration Basics	25%
4.0 Testing and Troubleshooting	10%
Total	100%

1.0 Braze Technical Basics - 30%

- 1.01 Explain what Braze is and common technical tasks needed for integrating it into your tech stack
- 1.02 Make a plan for how Braze will interact with your tech stack
- 1.03 Explain common roles a developer will play in Braze onboarding
- 1.04 Explain how user data lives in Braze and how it is modeled and used
- 1.05 Explain how user lifecycles interact with data ingestion and export

2.0 SDK Integration Basics - 35%

- 2.01 Plan which aspects of SDK integration will be needed for your organization
- 2.02 Use developer workspaces following best practices
- 2.03 Link your SDK code to the appropriate Braze workspace
- 2.04 List the key functions for controlling user lifecycle
- 2.05 Explain why and how to add aliases to users

- 2.06 Explain how to control the amount of data that is automatically collected by Braze
- 2.07 Submit standard user attributes to Braze through the correct SDK functions
- 2.08 Submit custom attributes, custom events and purchase events via the SDK
- 2.09 Integrate basic off the shelf messaging into your app or website
- 2.10 Implement location tracking via the SDK

3.0 Messaging Integration Basics - 25%

- 3.01 Differentiate push messages, in-app messages and content cards
- 3.02 Explain what a basic out-of-the-box integration provides for each of these channels
- 3.03 Describe common customizations for each channel
- 3.04 Locate resources for customization options
- 3.05 List common steps involved in channel customization
- 3.06 Explain the complexity of different levels of customization

4.0 Testing and Troubleshooting - 10%

- 4.01 Confirm the successful integration of the SDK
- 4.02 Confirm that data is correctly communicated to Braze by the SDK
- 4.03 Ensure messaging is correctly sending from Braze to your online ecosystem
- 4.04 Identify appropriate tools for solving common SDK integration issues
- 4.05 Submit a support ticket
- 4.06 Use Braze dashboard tools to troubleshoot SDK issues

Sample Test Questions

(see end of document for Answer Key)

Q1: Which tool provides real-time data ingestion from within your app or website?

A. Braze SDK

B. Braze CSV import

C. Braze Currents

D. Braze Batch Imports

Q2: What is the primary use of the Braze API?

A. To ensure that your apps and websites are sending the appropriate standard and custom data information into Braze

B. To activate in-app and device messaging channels

C. To import historical data, communicate between components of your tech stack, send transactional messages, and provide continuous updates not triggered via the SDK

D. To implement any planned UI component customizations

Q3: Which are examples of out-of-the-box or automatically collected data? (Select 2) *(Select 2)*

A. Lapsing user status

B. First name

C. Session count

D. Device ID

Q4: Charlie logs into the Pyrite Financial app with his user profile. What is the key ID code used to identify this profile in Braze?

A. braze_id

B. external_id

C. user_alias

D. db_key

Q5: Which of the following should be sent as a custom event?

A. User completes registration

B. User's first and last name

C. Last purchase performed

D. User's preferred clothing size

Q6: Which one of the following statements is true about using Internal Groups?

A. You cannot test any Liquid in your message using this feature.

B. You can only add users with an external ID.

C. You can only test email messages.

D. You can test user event logging in addition to message delivery.

Q7: Which of the following is the correct first argument to the function changeUser() on most platforms?

A. user_email

B. user_alias

C. ssn

D. external_id

Q8: Which two of the following channels typically require SDK integration in order to be displayed to end-users? (Select 2)
(Select 2)

A. Push

B. Email

C. SMS

D. In-App Messages

Q9: The marketers at Calorie Rocket have limited access to engineering resources beyond a basic Braze SDK integration. They plan to use in-app messages frequently, but are worried about their ability to customize the look-and-feel of the messages without engineering assistance. Is this a significant concern?

A. Yes, in-app messages require engineering assistance for almost all forms of customization

B. No, complex in-app messages can be composed in the Braze dashboard with no additional engineering support required, using the drag-and-drop editor

Q10: True or False: You can customize notification sounds for push messages.

A. True

B. False

Answer Key:

Q1: A **Q2:** C **Q3:** C,D **Q4:** B **Q5:** A **Q6:** D **Q7:** D **Q8:** A,D **Q9:** B **Q10:** A



Contact us:
certification@braze.com



Visit our webpage:
www.braze.com/certification

Braze is a comprehensive customer engagement platform that powers relevant and memorable experiences between consumers and the brands they love.

braze