DUMPSBOSS.

IBM Mobile Foundation v8.0 Application Development

IBM C1000-003

Version Demo

Total Demo Questions: 8

Total Premium Questions: 80

Buy Premium PDF

https://dumpsboss.co

support@dumpsboss.co

support@dumpsboss.co
dumpsboss.co

Topic Break Down

Topic	No. of Questions
Topic 1, Mobile Foundation Overview	6
Topic 2, Mobile Application Development	19
Topic 3, Adapter Development	21
Topic 4, Security	21
Topic 5, Application Testing	1
Topic 6, Deployment	12
Total	80

QUESTION NO: 1

Which two responsibilities can be implemented by a Mobile Foundation adapter before a response is returned to a mobile application? (Choose two.)

A. Invoke an enterprise backend service

B. Transform a backend response for the mobile client

C. Build an iOS application package

D. Publish an application to a public application store

E. Replace the mobile device operating system

ANSWER: A B

Explanation:

An adapter can invoke an enterprise backend service and transform or aggregate the backend response before returning application-oriented data to the mobile client. This allows the adapter to isolate backend protocols and payload formats from the application, and to apply server-side integration logic.

An adapter does not build a native mobile application package or publish an application to a public application store. Those activities are performed by the platform build and distribution processes rather than by a deployed Mobile Foundation adapter. See the IBM Mobile Foundation adapters documentation.

QUESTION NO: 2

Which component is included in the Mobile Foundation Developer Kit?

A. IBM Mobile Foundation Documentation

B. IBM Mobile Foundation Console Adapter

C. IBM Mobile Foundation Command Line Interface (CLI)

D. Eclipse Integrated Development Environment (IDE) with Mobile Foundation plug-in

ANSWER: C

Explanation:

IBM Mobile Foundation Command Line Interface (CLI) is included in the Mobile Foundation Developer Kit because it is the principal local development tool used to create and manage Mobile Foundation application artifacts. Developers install the CLI in their development environment and use its commands to generate application and adapter projects, build and deploy adapters, register applications, and interact with a Mobile Foundation Server during development and testing. The CLI supports the repeatable, command-driven workflow expected for Mobile Foundation application development, including automation from build scripts and continuous-integration pipelines. IBM's Mobile Foundation component overview identifies the Mobile Foundation CLI as a Developer Kit component, alongside the client-side development resources used to build mobile applications. The CLI is installed through npm, reflecting its role as a developer workstation tool rather than a server-only runtime service. IBM documents installation and use of the command-line interface in the Mobile Foundation documentation at Installing the MobileFirst Foundation CLI. Additional product-component context is available in the Mobile Foundation components overview.

QUESTION NO: 3

Which two types of changes can be delivered to a Cordova application by using Mobile Foundation Direct Update? (Choose two.)

A. JavaScript files

B. CSS files

C. Native Android code

D. Cordova native plugins

E. iOS provisioning profiles

ANSWER: A B

Explanation:

Changes to JavaScript files and CSS files can be delivered through Direct Update because they are web resources in a Cordova application. Direct Update packages and downloads updated content from the application's web layer, such as files in the `www` directory.

Changes to native code, native plugins, or native platform configuration cannot be delivered through Direct Update. Those changes require the application to be rebuilt and distributed as a new native application binary. See the IBM Mobile Foundation Direct Update documentation.

QUESTION NO: 4

Which method must the developer implement when instantiating WL.Client.createSecurityCheckChallengeHandler?

A. cancel()

B. handleChallenge()

C. handleAuthenticationError()

D. submitChallengeAnswer(answer)

ANSWER: B

Explanation:

`handleChallenge()` is correct because it is the callback through which a JavaScript application receives and processes a security challenge returned by Mobile Foundation. After creating a challenge handler with `WL.Client.createSecurityCheckChallengeHandler`, the developer supplies a `handleChallenge` implementation for that handler. Mobile Foundation invokes this method when the associated security check requires credentials or other challenge-response data from the client.

Within `handleChallenge()`, the application examines the challenge payload and presents the appropriate user experience, such as a login form, a one-time-password prompt, or a custom credential screen. When the user has provided the necessary information, the handler can submit the response through its challenge-answer mechanism. This callback is therefore the essential integration point between the server-side security check and the client-side authentication flow. It allows the application to respond dynamically to the data supplied by the security check rather than treating authentication as a fixed request sequence.

IBM's Mobile Foundation documentation describes implementing JavaScript challenge handlers and defining `handleChallenge` to process challenges: Implementing challenge handlers in JavaScript applications. Additional authentication and security guidance is available in the Mobile Foundation authentication and security documentation.

QUESTION NO: 5

A Cordova application must receive notifications sent to a particular Push notification tag.

Which two actions must the application perform? (Choose two.)

A. Register the device with the Push service

B. Subscribe the device to the required tag

C. Deploy a Direct Update package

D. Create an HTTP adapter procedure

E. Enable Application Center distribution

ANSWER: A B

Explanation:

The application must register the device with the Mobile Foundation Push service and subscribe the registered device to the required tag. Device registration associates the application instance with a push-capable device, while tag subscription identifies the notification categories in which that device is interested.

When a notification is sent to the tag, Mobile Foundation identifies the devices that are subscribed to that tag and sends the notification through the configured push provider. Creating an adapter or deploying a Direct Update package does not subscribe a device to Push notifications. See the IBM Mobile Foundation Push notifications documentation.

QUESTION NO: 6

How can a Mobile Foundation application developer override properties during run time, without having to redeploy adapters?

A. Properties cannot be overridden at runtime

B. Use the built-in connection policy properties

C. Declare instance variables in the Java adapter

D. Use the Mobile Foundation Operations Console

ANSWER: D

Explanation:

Use the Mobile Foundation Operations Console. Mobile Foundation supports updating an already deployed adapter's configuration properties at runtime through the Operations Console, so an administrator or developer can adjust environment-specific values without building and redeploying the adapter artifact. This is intended for operational configuration that can vary after deployment, such as endpoint-related settings and other adapter properties exposed in the adapter configuration. The updated value is stored as part of the adapter configuration for the selected Mobile Foundation runtime and takes effect without the normal adapter redeployment workflow. This capability helps teams separate deployable adapter code from operational values, allowing configuration changes to be made safely and quickly in a running environment. The Operations Console is therefore the appropriate administrative interface for overriding applicable adapter properties during runtime. IBM's Mobile Foundation documentation describes adapter administration and runtime configuration features in the IBM Mobile Foundation 8.0 documentation, including the Operations Console's role in managing deployed artifacts. IBM also provides Mobile Foundation product information and documentation access at IBM Mobile Foundation.

QUESTION NO: 7

What is the primary purpose of a Mobile Foundation security check?

A. To package a Cordova application for deployment

B. To protect a resource by authenticating and authorizing requests

C. To persist application data in a local database

D. To send a push notification to a device

ANSWER: B

Explanation:

A Mobile Foundation security check is used to protect a resource by validating a client request and, when required, challenging the client for authentication information. A security check can establish a user identity, validate credentials, enforce device or application conditions, and determine whether access should be granted.

When a protected adapter resource is invoked, Mobile Foundation evaluates the associated security check. If additional credentials or information are required, the server returns a challenge that the client challenge handler processes. See the IBM Mobile Foundation security checks documentation.

QUESTION NO: 8

A developer is configuring an `MFP.Server.invokeHttp(options)` call in a JavaScript HTTP adapter procedure.

Which two optional properties can be used to further define the outbound HTTP request? (Choose two.)

A. method

B. headers

C. securityCheck

D. adapterName

E. clientId

ANSWER: A B

Explanation:

The `method` property can be used to select an HTTP method other than the default GET method, and the `headers` property can be used to provide HTTP request headers required by the external service. For example, a procedure can use POST and provide an Accept or authorization header when the backend requires it.

These properties supplement the required request path and expected returned content type. A security check is configured as part of adapter resource protection and is not an `invokeHttp` request option. See the IBM Mobile Foundation HTTP adapter documentation.