IBM Cloud Professional Certification Program

Study Guide Series

Exam C1000-003- IBM Mobile Foundation v8.0 Application Development

Purpose of Exam Objectives

When an exam is being developed, the Subject Matter Experts work together to define the role the certified individual will fill. They define all of the tasks and knowledge that an individual would need to have in order to successfully implement the product. This creates the foundation for the objectives and measurement criteria, which are the basis for the certification exam.

The Watson Developer Certification item writers used these objectives to develop the questions that they wroteand which will appear on the exam.

It is recommended that you review these objectives. Do you know how to complete the task in the objective? Do you know why that task needs to be done? Do you know what will happen if you do it incorrectly? If you are not familiar with a task, then go through the objective and perform that task in your own environment. Read more information on the task. If there is an objective on a task there is about a 95% chance that you WILL see a question about it on the actual exam.

After you have reviewed the objectives and completed your own research, then take the assessment exam. While the assessment exam will not tell you which question you answered incorrectly, it will tell you how you did by section. This will give you a good indication as to whether you are ready to take the actual exam or if you need to further review the materials.

Note: This is the high-level list of objectives. As you review these objectives, click for a more detailed level of how to perform the task.

High-level Exam Objectives

Section 1 - Development Environment Set-up	
1.1	Download and install Dev Kit
1.2	Set up a platform specific IDE and tools
1.3	Install Eclipse plug-in
1.4	Install Mobile Foundation CLI
Section 2 - Development: Architecture	
2.1	Describe Mobile Foundation components and architecture
2.2	Integrate with data sources
2.3	Understand and be able to differentiate the different types of client-side development enabled by Mobile Foundation
Section 3 - Using the Command Line Interface (CLI)	
3.1	Configure and use the CLI

3.2	Manage the server profile
3.3	Create and manage apps
3.4	Create and manage adaptors
Section 4 - Development: Client Side	
4.1	Add the Mobile Foundation SDK to an application
4.2	Connect to the Mobile Foundation Server
4.3	Customize the startup process
4.4	Use WL resources requests to connect to server side resources
4.5	Debug an application
4.6	Add multilingual support for Mobile Foundation to the application
4.7	Use direct update to refresh web resources
4.8	Use live update to segment users and deliver customized content
4.9	Use JSONStore
4.10	Implement the Simple Data Sharing feature
Section 5 - Development: Server Side	
5.1	Create Mobile Foundation adapters
5.2	Test adapters using the OpenAPI Specification
5.3	Invoke Java code from adapters
5.4	Invoke adapters from other adapters
5.5	Use Mobile First server side APIs
Section 6 - Notifications	
6.1	Configure Push notifications
6.2	Implement and test Push and SMS Notification mechanisms
Section 7 - Authentication and Security	
7.1	Describe the OAuth 2.0 based Mobile Foundation security framework
7.2	Manage device enrollment
7.3	Create predefined security checks
7.4	Create custom security checks
7.5	Implement application authenticity
7.6	Implement device single sign-on (SSO)
7.7	Create and use Confidential Clients
7.8	Implement Certificate Pinning
Section 8 - Deployment	
8.1	Build and deploy applications and adapters to multiple environments
8.2	Configure Mobile Foundation Server settings
8.3	Configure and deploy adapters

8.4	Migrating from earlier realeases
Section 9 - Analytics and Reports	
9.1	Understand the capabilities provided by Mobile Foundation Operational Analytics
9.2	Enable access to raw and analytic reporting
9.3	Use the Analytics REST API
9.4	Access client side crash logs
9.5	Use MobileFirst Analytics to capture custom user analytics

Detailed Exam Objectives

Section 1 - Development Environment Set-up

1.1. Download and install Dev Kit

SUBTASK(S):
1.1.1. Ensure all prerequisites are installed.
1.1.2. Download the IBM MobileFirst Platform Foundation Developer Kit from the Download page.
1.1.3. Start the installation program (follow the installation instructions).
1.1.4. Start the Mobile Foundation Server.

REFERENCES:
https://www.ibm.com/support/knowledgecenter/en/SSHS8R_8.0.0/com.ibm.worklight.dev.doc/dev/t_installing_dev_kit.html
https://mobilefirstplatform.ibmcloud.com/downloads/#collapseDevKit
https://mobilefirstplatform.ibmcloud.com/tutorials/en/foundation/8.0/installationconfiguration/development/mobilefirst/installation-guide/

1.2. Set up a platform specific IDE and tools SUBTASK(S):
1.2.1. Setup platform Specific IDE and Tools for Client Side Development
- 1.2.1.1. Cordova
  - 1.2.1.1.1. Ensure all prerequisites are installed.
  - 1.2.1.1.2. Open a terminal or command prompt
  - 1.2.1.1.3. Type the npm command to install Cordova
  - 1.2.1.1.4. Install text editor of choice (Visual Studio Code, Sublime Text, etc.)
- 1.2.1.2. iOS
  - 1.2.1.2.1. Ensure all prerequisites are met; iOS 8 or later
  - 1.2.1.2.2. Install Xcode 7.1 or later
  - 1.2.1.2.3. Install CocoaPods
  - 1.2.1.2.4. Register as an iOS Developer
- 1.2.1.3. Android
  - 1.2.1.3.1. Download Android Developer Studio from Download Site.
  - 1.2.1.3.2. Configure Android Developer Studio
  - 1.2.1.3.3. Configure Android Emulator
- 1.2.1.4. Windows
  - 1.2.1.4.1. Ensure all prerequisites are met
  - 1.2.1.4.2. Register for a Microsoft account
  - 1.2.1.4.3. Download and install Visual Studio
    - 1.2.1.4.3.1. Download and install Visual Studio 2013 or 2015 for Windows 8.1 Universal development

1.2.1.4.3.2. Download and install Visual Studio 2015 for Windows 10 Universal Windows Platform (UWP) development

1.2.1.5. Xamarin

1.2.1.5.1. Download and install Xamarin Studio from Download site for Mac or Windows

1.2.1.6. Web

1.2.1.6.1. Ensure that all prerequisites are met

1.2.2. Setup platform Specific IDE and Tools for Server Side (Adapter) Development

1.2.2.1. Ensure prerequisites are installed
1.2.2.2. Install NodeJS
1.2.2.3. Install Mobile Foundation Command-Line-Interface (CLI)
1.2.2.4. Install Maven
1.2.2.5. Install an IDE with Java support such as IntelliJ or Eclipse

REFERENCES:

https://mobilefirstplatform.ibmcloud.com/tutorials/en/foundation/8.0/installationconfiguration/development/

Cordova

https://guides.cocoapods.org/using/getting-started.html

iOS

https://mobilefirstplatform.ibmcloud.com/tutorials/en/foundation/8.0/installationconfiguration/development/ios/

https://mobilefirstplatform.ibmcloud.com/tutorials/en/foundation/8.0/installationconfiguration/development/cordova/

Android

https://mobilefirstplatform.ibmcloud.com/tutorials/en/foundation/8.0/installationconfiguration/development/android/ https://developer.android.com/studio/index.html
https://developer.android.com/studio/intro/studio-config.html
https://developer.android.com/studio/run/emulator.html

Windows

https://mobilefirstplatform.ibmcloud.com/tutorials/en/foundation/8.0/installationconfiguration/development/windows/

Xamarin

https://mobilefirstplatform.ibmcloud.com/tutorials/en/foundation/8.0/installationconfiguration/development/xamarin/

Web
https://mobilefirstplatform.ibmcloud.com/tutorials/en/foundation/8.0/installationconfiguration/development/web/

1.3. Install Eclipse plug-in SUBTASK(S):
1.3.1. Ensure prerequisites are install, such as Eclipse Neon or later.
1.3.2. Install Eclipse Plugin THYM 2.0.0 from the Eclipse Marketplace.
1.3.3. Install Eclipse Plugin IBM MobileFirst Studio Version 8.0 from the Eclipse Marketplace.

REFERENCES:
https://mobilefirstplatform.ibmcloud.com/blog/2016/06/17/ibm-mobilefirst-studio-8-0plugin-for-eclipse-now-available/
https://marketplace.eclipse.org/content/ibm-mobilefirst-foundation-studio

1.4. Install the Mobile Foundation Command-Line-Interface (CLI) SUBTASK(S):
1.4.1. Ensure all prerequisites, such as node, maven, etc., are installed

1.4.1.1. OPTION 1 - Install Mobile Foundation CLI from Mobile Foundation Operations Console
1.4.1.1.1. With the Mobile Foundation Server Started, start the Mobile Operations Console
1.4.1.1.2. Navigate to the Downloads page on the Tools Tab
1.4.1.1.3. Download the Developer CLI
1.4.1.1.4. Open a terminal or command prompt
1.4.1.1.5. Type the npm command to install the Mobile Foundation CLI

1.4.1.2. **OPTION 2 -** Install Mobile Foundation CLI from npmjs repository
1.4.1.2.1. Open a terminal or command prompt
1.4.1.2.2. Type the npm command to install the Mobile Foundation CLI

REFERENCES:
https://www.ibm.com/support/knowledgecenter/en/SSHS8R_8.0.0/com.ibm.worklight.dev.doc/dev/t_wl_installing_cli.html
https://mobilefirstplatform.ibmcloud.com/downloads/#collapseCli8

Section *2*- Development: Architecture

2.1. Describe Mobile Foundation components and architecture

SUBTASK(S):
2.1.1. Describe the Mobile Foundation Server
2.1.2. Describe the Mobile Foundation Operations Console
2.1.3. Describe the Mobile Foundation Command-Line-Interface (CLI)

2.1.4. Describe client-side Mobile SDKs
2.1.5. Describe a Mobile Foundation Adapter
2.1.6. Describe the Mobile Foundation Analytics Server
2.1.7. Describe the IBM Application Center

REFERENCES:
https://mobilefirstplatform.ibmcloud.com/tutorials/en/foundation/8.0/installationconfiguration/development/mobilefirst/#mobilefirst-foundation-components

2.2. Integrate with data sources

SUBTASK(S):
2.2.1. Describe the benefits of using adapters
2.2.2. Describe the benefits specific to using Java adapters
2.2.3. Describe the types of backend systems that adapters can integrate with (HTTP, SQL, etc)

REFERENCES:
https://mobilefirstplatform.ibmcloud.com/tutorials/en/foundation/8.0/adapters/
https://www.ibm.com/support/knowledgecenter/en/SSHS8R_8.0.0/com.ibm.worklight.dev.doc/devref/c_overview_of_ibm_adaps_top_level.html
http://mobilefirstplatform.ibmcloud.com/tutorials/en/foundation/8.0/adapters/creatingadapters/

2.3. Understand and be able to differentiate the different types of client-side development enabled by Mobile Foundation

SUBTASK(S):
2.3.1. Describe the types of client applications that are supported by Mobile Foundation
    2.3.1.1. Cordova
    2.3.1.2. iOS
    2.3.1.3. Android
    2.3.1.4. Windows 8.1 Universal or Windows 10 UWP
    2.3.1.5. Xamarin
    2.3.1.6. Web
2.3.2. Determine a development approach (either web, hybrid or native) for a particular use case.
2.3.3. Identify 4 popular frameworks that can be used to enhance Cordova application development **REFERENCES:**

- https://mobilefirstplatform.ibmcloud.com/tutorials/en/foundation/8.0/applicationdevelopment/#applications
- https://mobilefirstplatform.ibmcloud.com/tutorials/en/foundation/8.0/applicationdevelopment/cordova-apps/

Section 3 - Using the Command Line Interface (CLI)

3.1. Configure and use the CLI

SUBTASK(S):

3.1.1. Explain the differences between Interactive CLI and Direct mode CLI

3.1.2. Execute the `mfpdev config` command to display a series of prompts to ask you which server setting you want to configure, and then prompts you for the appropriate value.

3.1.2.1. Set "Development Browser" to change the default browser that is used to preview Cordova applications

3.1.2.2. Set "Default Preview Type" to use either the browser or the "MobileFirst Mobile Browser Simulator" to preview Cordova applications

REFERENCES:

IBM MobileFirst Platform Foundation CLI Description
https://www.ibm.com/support/knowledgecenter/SSHS8R_8.0.0/com.ibm.worklight.dev.doc/dev/c_wl_cli_description.html

IBM MobileFirst Platform Foundation CLI Package Repository on NPMJS
https://www.npmjs.com/package/mfpdev-cli

`mfpdev help <command>` Displays help for MobileFirst CLI (mfpdev) commands. With an argument, displays more specific help text for each command type or command. i.e "mfpdev help server add"

3.2. Manage the server profile

SUBTASK(S):

3.2.1. Execute the `mfpdev server <command action>` with one of the following command actions to create and manage apps:

3.2.1.1. info – Displays information about all the MobileFirst Server instances available to be used

3.2.1.2. add – Adds a new MobileFirst server profile to your environment

3.2.1.3. edit – Enables you to edit the details of a registered server definition

3.2.1.4. remove – Removes a server definition from your environment

3.2.1.5. console – Opens the MobileFirst Operations Console in a browser

3.2.1.6. clean – Unregisters apps and removes adapters from the MobileFirst Server

REFERENCES:

IBM MobileFirst Platform Foundation CLI Description
https://www.ibm.com/support/knowledgecenter/SSHS8R_8.0.0/com.ibm.worklight.dev.doc/dev/c_wl_cli_description.html

IBM MobileFirst Platform Foundation CLI Package Repository on NPMJS
https://www.npmjs.com/package/mfpdev-cli

`mfpdev help <command>` Displays help for MobileFirst CLI (mfpdev) commands. With an argument, displays more specific help text for each command type or command. i.e “mfpdev help server add”

3.3. Create and manage apps

SUBTASK(S):

3.3.1. Execute the `mfpdev app <command action>` with one of the following command actions to create and manage apps:

- 3.3.1.1. register – Registers your app with a MobileFirst Server. An application must be registered in a MobileFirst Server when it is ready to be executed.
- 3.3.1.2. config – Enables you to specify the back-end server and runtime to use for your app. In addition, for Cordova apps, enables you to configure several additional aspects such as the default language for system messages and whether to do a checksum security check. Other configuration parameters are included for Cordova apps.
- 3.3.1.3. pull – Retrieves an existing app configuration from the server.
- 3.3.1.4. push – Sends an app"s configuration to the server.
- 3.3.1.5. preview – Enables you to preview your Cordova app without requiring an actual device of the target platform type. You can view the preview in either the Mobile Browser Simulator or your web browser.
- 3.3.1.6. webupdate – Packages the application resources contained in the www directory into a .zip file that can be used for the direct update process.

REFERENCES:

IBM MobileFirst Platform Foundation CLI Description
https://www.ibm.com/support/knowledgecenter/SSHS8R_8.0.0/com.ibm.worklight.dev.doc/dev/c_wl_cli_description.html
IBM MobileFirst Platform Foundation CLI Package Repository on NPMJS
https://www.npmjs.com/package/mfpdev-cli
`mfpdev help <command>` Displays help for MobileFirst CLI (mfpdev) commands. With an argument, displays more specific help text for each command type or command. i.e “mfpdev help server add”

3.4. Create and manage adapters

SUBTASK(S):

3.4.1. Execute the `mfpdev adapter <command action>` with one of the following command actions to create and manage apps:

- 3.4.1.1. create – Creates an adapter
- 3.4.1.2. build – Builds an adapter
- 3.4.1.3. build all – Finds and builds all of the adapters in the current directory and its subdirectories

3.4.1.4. deploy – Deploys an adapter to the MobileFirst Server
3.4.1.5. deploy all – Finds all of the adapters in the current directory and its subdirectories, and deploys them to the MobileFirst Server
3.4.1.6. call – Calls an adapter‟s procedure on the MobileFirst Server
3.4.1.7. pull – Retrieves an existing adapter configuration from the server
3.4.1.8. push – Sends an adapter‟s configuration to the server

REFERENCES:

IBM MobileFirst Platform Foundation CLI Description
https://www.ibm.com/support/knowledgecenter/SSHS8R_8.0.0/com.ibm.worklight.dev.doc/dev/c_wl_cli_description.html
IBM MobileFirst Platform Foundation CLI Package Repository on NPMJS
https://www.npmjs.com/package/mfpdev-cli
`mfpdev help <command>` Displays help for MobileFirst CLI (mfpdev) commands. With an argument, displays more specific help text for each command type or command. i.e “mfpdev help server add”

Section 4 - Development: Client Side

4.1. Add the Mobile Foundation SDKto an application

SUBTASK(S):
4.1.1. Identify the different Cordova plugins that make up the Mobile Foundation Cordova SDK
4.1.2. Understand the platform versions supported by the Cordova SDK
4.1.3. Add the Mobile Foundation SDKto an application
    4.1.3.1. Cordova-based application
    4.1.3.2. iOS-based application
    4.1.3.3. Android-based application
    4.1.3.4. Windows based application
    4.1.3.5. Xamarin-based application
    4.1.3.6. Web-based application

REFERENCES:
https://mobilefirstplatform.ibmcloud.com/tutorials/en/foundation/8.0/applicationdevelopment/sdk/(and its sub-pages)

4.2. Connect to the Mobile Foundation Server

SUBTASK(S):
4.2.1. Describe how to use the WLAuthorizationManager class to manage tokens in a mobile application
4.2.2. Describe how to use the WLAuthorizationManager class to login to a Mobile Foundation Server
4.2.3. Describe how to use onSuccess and onFailure challenge handlers

REFERENCES:
http://mobilefirstplatform.ibmcloud.com/tutorials/en/foundation/8.0/api/client-side-api/

4.3. Customize the startup process

SUBTASK(S):
4.3.1. Describe the startup flows for each application type/SDK
4.3.2. Describe the Options parameter used across much of the API, describe its contents and how it is used.
4.3.3. Describe the initialization methods and APIs for each SDK environment and explain when they are called during the application initialization process

REFERENCES:
http://mobilefirstplatform.ibmcloud.com/tutorials/en/foundation/8.0/api/client-side-api/

4.4. Use WL resources requests to connect to server side resources

SUBTASK(S):

4.4.1. Describe the resource types that can be accessed using WLResourceRequest
4.4.2. Make a resource request
    4.4.2.1. JavaScript
    4.4.2.2. Java
    4.4.2.3. Swift/Objective-C
    4.4.2.4. C#

REFERENCES:
https://mobilefirstplatform.ibmcloud.com/tutorials/en/foundation/8.0/applicationdevelopment/resource-request/

4.5. Debug an application

SUBTASK(S):
4.5.1. Understand when to use the following debugging tools:
    4.5.1.1. Mobile Browser Simulator
    4.5.1.2. Native platform debuggers, such as those included with Xcode, Android Studio and Microsoft Visual Studio
    4.5.1.3. Physical devices
4.5.2. Add WL.Logging statements to your sample application and view the results in the Analytics Console
4.5.3. Set the trace specification for different application server types
4.5.4. Understand over the wire debugging
4.5.5. Capture client logs from a physical device
4.5.6. Use the IDE Console to view and capture client logs from a physical device

REFERENCES:
- Debugging JavaScript (Cordova, Web) Applications
- MustGather -- Applications Wireshark

4.6. Add multilingual support for Mobile Foundation to the application

SUBTASK(S):
4.6.1. Enable application for multilingual support
    4.6.1.1. JavaScript
    4.6.1.2. Java
    4.6.1.3. Swift/Objective-C
    4.6.1.4. C#

REFERENCES:
https://mobilefirstplatform.ibmcloud.com/tutorials/en/foundation/8.0/applicationdevelopment/translation/

4.7. Use direct update to refresh web resources

SUBTASK(S):

4.7.1. Understand packaging of resources to be updated
4.7.2. Define which mobile platforms supported
4.7.3. Use of mfpdev CLI
4.7.4. Cusomization of the direct update UI **REFERENCES:**

- Using Direct Update in Cordova applications

4.8. Use live update to segment users and deliver customized content

SUBTASK(S):

4.8.1. Add the Live Update SDK to applications.
4.8.2. Configure Live Update settings in the Mobile Foundation Server.
4.8.3. Configure application security for Live Update (add a scope element).
4.8.4. Configure a schema and segments for Live Update.
4.8.5. Implement and test live update

REFERENCES:

- https://mobilefirstplatform.ibmcloud.com/tutorials/en/foundation/8.0/applicationdevelopment/live-update/

4.9. Use JSONStore (e.g., syncing, encrypting, storing)

SUBTASK(S):

4.9.1. Create multiple stores that contain different collections in a single Mobile Foundation application.
4.9.2. Secure collections in a store by encrypting them.
4.9.3. Collect key pieces of analytics information that are related to JSONStore
4.9.4. Work with external data (pull or push).
4.9.5. Implement data synchronization by using an adapter.
4.9.6. Search for content in the JSONStore

REFERENCES:

- https://mobilefirstplatform.ibmcloud.com/tutorials/en/foundation/8.0/applicationdevelopment/jsonstore/

4.10. Implement the Simple Data Sharing feature

SUBTASK(S):

4.10.1. Enable the Simple Data Sharing feature (iOS, Android, etc).
4.10.2. Set, get, and delete tokens from the shared credential storage (JavaScript, Java, Objective-C, etc).

REFERENCES:

https://mobilefirstplatform.ibmcloud.com/tutorials/en/foundation/8.0/applicationdevelopment/simple-data-sharing/

Section 5 - Development: Server Side

5.1. Create Mobile Foundation adapters (HTTP, SQL, and Java)

SUBTASK(S):
5.1.1. Create an HTTP adapter
5.1.2. Create an SQL adapter
5.1.3. Create a Java adapter
5.1.4. Build an adapter
5.1.5. Deploy an adapter using command-line-interface
5.1.6. Deploy an adapter using Mobile Foundation Operations Console 5.1.7.
REFERENCES:
https://mobilefirstplatform.ibmcloud.com/tutorials/en/foundation/8.0/adapters/creatingadapters/
https://www.ibm.com/support/knowledgecenter/en/SSHS8R_8.0.0/com.ibm.worklight.dev.doc/devref/c_overview_of_ibm_adaps_top_level.html

5.2. Test adapters using the OpenAPI Specification (Swagger)

SUBTASK(S):
5.2.1. Understand OpenAPI Specification (Swagger)
5.2.2. Describe the advantages to using the OpenAPI specification

REFERENCES:
http://swagger.io/getting-started
https://mobilefirstplatform.ibmcloud.com/tutorials/en/foundation/8.0/adapters/testingand-debugging-adapters/#testing-adapters

5.3. Invoke Java code from adapters

SUBTASK(S):
5.3.1. Create a custom Java class with a valid package name
5.3.2. Create a Javascript adapter
5.3.3. Place the Java class in the appropriate directory for the adapter to access
5.3.4. Call the Java class
5.3.5. Return the response to the client

REFERENCES:
https://www.ibm.com/support/knowledgecenter/en/SSHS8R_8.0.0/com.ibm.worklight.dev.doc/devref/t_calling_java_code_from_a_javas.html
https://mobilefirstplatform.ibmcloud.com/tutorials/en/foundation/8.0/adapters/javascriptadapters/using-java-in-javascript-adapters/

5.4. Invoke adapters from other adapters

SUBTASK(S):
5.4.1. Calling a JavaScript adapter procedure from a JavaScript adapter
5.4.2. Calling a Java adapter from a Java adapter
5.4.3. Calling a JavaScript adapter procedure from a Java adapter
5.4.4. Data mashup **REFERENCES:**
https://mobilefirstplatform.ibmcloud.com/tutorials/en/foundation/8.0/adapters/advancedadapter-usage-mashup/

5.5. Use Mobile Foundation server side APIs

SUBTASK(S):
5.5.1. Calling common server side SDK classes and methods
5.5.2. Calling JavaScript server-side API
5.5.3. Calling Java server-side API
5.5.4. Calling REST API for Mobile Foundation Runtime

REFERENCES:
https://www.ibm.com/support/knowledgecenter/en/SSHS8R_8.0.0/com.ibm.worklight.dev.doc/devref/c_java_server_side_api.html
https://mobilefirstplatform.ibmcloud.com/tutorials/en/foundation/8.0/api/server-side-api/
https://mobilefirstplatform.ibmcloud.com/tutorials/en/foundation/8.0/api/rest/

Section 6 - Notifications

6.1. Configure Push notifications (scope mapping)

SUBTASK(S):
6.1.1. Start the server console
6.1.2. Navigate to the application to Push Settings
6.1.3. Configure push settings for iOS, Android and/or Windows
6.1.4. Create push notification tags
6.1.5. Test push notification using console

REFERENCES:
http://mobilefirstplatform.ibmcloud.com/tutorials/en/foundation/8.0/notifications/
http://mobilefirstplatform.ibmcloud.com/tutorials/en/foundation/8.0/notifications/sendingnotifications/

6.2. Implement and test Push and SMS Notification mechanisms (tags, authentication)

SUBTASK(S):
6.2.1. Add push notification plug-in
6.2.2. Subscribe
    6.2.2.1. Initialize Push Notification in app
    6.2.2.2. Register device
    6.2.2.3. Subscribe to tags
    6.2.2.4. Handle notifications
6.2.3. Unsubscribe
    6.2.3.1. Unsubscribe from tags
    6.2.3.2. Unregister device
6.2.4. Calling REST Push API
6.2.5. Architecture of Mobile Foundation Push Notifications

REFERENCES:
http://mobilefirstplatform.ibmcloud.com/tutorials/en/foundation/8.0/notifications/
http://mobilefirstplatform.ibmcloud.com/tutorials/en/foundation/8.0/notifications/rest-apis/
http://mobilefirstplatform.ibmcloud.com/tutorials/en/foundation/8.0/notifications/handlingsms-notifications/
http://mobilefirstplatform.ibmcloud.com/tutorials/en/foundation/8.0/notifications/handlingpush-notifications/

Section 7 - Authentication and Security

7.1. Describe the OAuth 2.0 based Mobile Foundation security framework

SUBTASK(S):

7.1.1. Describe the Mobile Foundation Security Overview
7.1.2. Understand the OAuth2 Specification
    7.1.2.1. Access tokens
    7.1.2.2. Authorization code grant type
    7.1.2.3. Scopes
7.1.3. Illustrate how to obtain an access token
7.1.4. Describe how scope, security checks, and challenge handlers work together to secure Mobile Foundation
7.1.5. Use a token to access a protected resource **REFERENCES:**

- https://tools.ietf.org/html/rfc6749
- https://www.ibm.com/support/knowledgecenter/en/SSHS8R_8.0.0/com.ibm.worklight.dev.doc/dev/c_oauth_security_model.html
- https://mobilefirstplatform.ibmcloud.com/tutorials/en/foundation/8.0/authenticationand-security/

7.2. Manage device enrollment

SUBTASK(S):

7.2.1. Define your login module, security tests, and realms in your adapter.
7.2.2. Add Challenge Handler
7.2.3. Implement code to save device ID in your chosen storage
7.2.4. Remove or check for enrollment status.
7.2.5. Pin Code
7.2.5.1. Set a pin code for the client.
7.2.5.2. Remove a pin code previously set by the client.
7.2.5.3. Protect access to a feature or resource using a security check by implementing a security check to validate pin code.

REFERENCES:

- https://mobilefirstplatform.ibmcloud.com/tutorials/en/foundation/7.1/advancedtopics/device-enrollment/
- https://mobilefirstplatform.ibmcloud.com/tutorials/en/foundation/8.0/authenticationand-security/enrollment/

7.3. Create predefined security checks

SUBTASK(S):

7.3.1. Predefined
    7.3.1.1. EnrollmentUserLogin
    7.3.1.2. EnrollmentPinCode

REFERENCES:

- https://mobilefirstplatform.ibmcloud.com/tutorials/en/foundation/8.0/authenticationand-security/creating-a-security-check/

7.4. Create custom security checks

SUBTASK(S):

7.4.1. Define security check definition XML file
7.4.2. Implement custom security check in Java.
7.4.2.1. ExternalizableSecurityCheck
7.4.2.2. CredentialsValidationSecurityCheck
7.4.2.3. UserAuthenticationSecurityCheck
7.4.3. Configure a security check at application or adapter level
7.4.4. Add RememberMe functionality
7.4.5. Use the security check to protect an external resource
7.4.6. Create a Challenge Handler for custom security check
7.4.6.1. Register Challenge Handler to Security Check (scope)
7.4.6.2. Implement Challenge Handler in Client Code
7.4.6.3. Retrieve authenticated user details

REFERENCES:

- https://mobilefirstplatform.ibmcloud.com/blog/2016/06/22/challenge-handlers/
- https://mobilefirstplatform.ibmcloud.com/tutorials/en/foundation/8.0/authenticationand-security/user-authentication/security-check/

7.5. Implement application authenticity (basic and dynamic)

SUBTASK(S):

7.5.1. Enable the application-authenticity security check
7.5.2. Configure application authenticity to set expiration
7.5.3. Enable Build Time Secret (iOS only)
7.5.4. Reset the application-authenticity algorithm
7.5.5. Run CLI command to set validation type to either dynamic or static

REFERENCES:

- https://mobilefirstplatform.ibmcloud.com/tutorials/en/foundation/8.0/authenticationand-security/application-authenticity/

7.6. Implement device single sign-on (SSO) (LTPA)

SUBTASK(S):

7.6.1. Configure the LTPA predefined security check
7.6.2. Configure Device SSO in application descriptor file

REFERENCES:

- https://mobilefirstplatform.ibmcloud.com/tutorials/en/foundation/8.0/authentication -and-security/device-sso/
- https://mobilefirstplatform.ibmcloud.com/tutorials/en/foundation/8.0/authentication -and-security/ltpa-security-check/

7.7. Create and use Confidential Clients

SUBTASK(S):

7.7.1. Register a confidential client in the Mobile Foundation Operations Console
7.7.2. Use basic authentication and make a POST request to obtain an access token
7.7.3. Add the bearer token to the header and make a request to protected resource as confidential client.

REFERENCES:

- https://mobilefirstplatform.ibmcloud.com/tutorials/en/foundation/8.0/authentication - and-security/confidential-clients/

7.8. Implement certificate pinning

SUBTASK(S):

7.8.1. Configure the Mobile Foundationserver or DevKit keystore
7.8.2. Add certificate to application bundle (iOS), asset folder (Android), or certificate folder (Cordova)
7.8.3. Pin the certificate using the appropriate API method for your platform

REFERENCES:

- https://mobilefirstplatform.ibmcloud.com/tutorials/en/foundation/8.0/authentication - and-security/certificate-pinning/
- http://mobilefirstplatform.ibmcloud.com/tutorials/en/foundation/8.0/authenticationand-security/configuring-the-mobilefirst-server-keystore/

Section 8 - Deployment

8.1. Build and deploy applications and adapters to multiple environments

SUBTASK(S):

8.1.1. Ensure endpoint is properly configured.
8.1.2. Register an application to the server.
8.1.3. Build an application for deployment to a test or production environment
8.1.4. Transfer application artifacts to the server by using mfpdev, admin service, REST API, or Mobile Foundation Operations Console.
8.1.5. Deploy a new application version to production (increment version number).

REFERENCES:

- https://mobilefirstplatform.ibmcloud.com/tutorials/en/foundation/8.0/installationconfiguration/production/server-configuration/#registering-applications-anddeploying-adapters-to-different-runtimes
- Deploying Mobile Applications on Bluemix with IBM Mobile Foundation Course code ZK504 ERC 1.

- https://mobilefirstplatform.ibmcloud.com/tutorials/en/foundation/8.0/administeringapps/deployment/

8.2. Configure Mobile Foundation Server settings

SUBTASK(S):

8.2.1. Export and import a runtime configuration to another Mobile Foundation Server (using REST APIs or admin service).
8.2.2. Create whitelists and blacklists for endpoints of the IBM MobileFirst Server
8.2.3. Configure data sources.
8.2.4. Configure logging.
8.2.5. Configure license tracking.
8.2.6. Configure SSL.

REFERENCES:

- Deploying Mobile Applications on Bluemix with IBM Mobile Foundation Course code ZK504 ERC 1.0
- https://mobilefirstplatform.ibmcloud.com/tutorials/en/foundation/8.0/installationconfiguration/production/server-configuration/

8.3. Configure and deploy adapters

SUBTASK(S):

8.3.1. Deploy or update an existing adapter in a target environment.
8.3.2. Verify the server-side configuration of an adapter (custom properties in the console)

REFERENCES:

- Deploying Mobile Applications on Bluemix with IBM Mobile Foundation Course code ZK504 ERC 1.0
- https://mobilefirstplatform.ibmcloud.com/tutorials/en/foundation/8.0/administeringapps/deployment/

8.4. Migrating from earlier releases

SUBTASK(S):

8.4.1. Mobile Foundation development process changes
8.4.2. Installing and use of the migration assessment tool
8.4.3. Moving a hybrid mobile application to Cordova
8.4.4. Use of Apache Maven for transformation of Adapters
8.4.5. Configuration changes to push notifications
8.4.6. Using the Mobile Foundation SDK‟s

REFERENCES:

Migrating from earlier releases
>>https://mobilefirstplatform.ibmcloud.com/tutorials/en/foundation/8.0/upgrading/ -changes-in-the-development-and-deployment-process
Mobile Foundation Cookbook
>>https://mobilefirstplatform.ibmcloud.com/tutorials/en/foundation/8.0/upgrading/migration-cookbook/

Section 9 - Analytics and Reports

9.1. Understand the capabilities provided by Mobile Foundation Operational Analytics

SUBTASK(S):

9.1.1. Describe the type of data that can be collected by Mobile Foundation Operational Analytics

9.1.1.1. Mobile Foundation Operational Analytics collects data from app-to-server activities, client logs, client crashes, and server-side logs from the MobileFirst Server and client devices. Included are: default reports of user retention, crash reports, device type and operating system breakdowns, custom data and custom charts, network usage, push notification results, in-app behavior, debug log collection, and more.

9.1.1.2. MobileFirst Server comes pre-instrumented with network infrastructure reporting. When both the client and server are reporting network usage, the data is aggregated so you can attribute poor performance to the network, the server, or the back-end systems. In addition, you can control which logger data is accessed and used by analytics by defining filters both on the client-side and on the MobileFirst Analytics Server. You choose the verbosity and data retention policy of the reported events, set conditional alerts, build custom charts and engage with new data.

9.1.2. Describe the 2 client side APIs of MobileFirst Foundation Operational Analytics

9.1.2.1. The Analytics client API collects data on a wide range of events and sends them to the MobileFirst Analytics Server.

9.1.2.1.1. By default, only logging from the package „com.worklight‟ is sent to analytics. To add logging from additional packages see “Forwarding logs to the Analytics Server” in the references.

9.1.2.1.2. List and describe the metrics available for capture: 9.1.2.1.2.1. Lifecycle events - app usage rates, usage duration, app crash rates

9.1.2.1.2.2. Network usage - breakdown of API call frequencies, network performance metrics

9.1.2.1.2.3. Users - users of the app that are identified by a supplied user ID

9.1.2.1.2.4. Custom analytics - custom key/value pairs that are defined by the app developer

9.1.2.2. The Logger client API functions as a standard logger. From the client you can send logger data to the MobileFirst Analytics Server at any logging level. (TRACE, DEBUG, LOG, INFO, WARN, ERROR, FATAL)

9.1.2.2.1. List and describe the various logging levels available:

9.1.2.2.1.1. TRACE - used for method entry and exit points

9.1.2.2.1.2. DEBUG - used for method result output

9.1.2.2.1.3. LOG - used for class instantiation

9.1.2.2.1.4. INFO - used for reporting initialization
9.1.2.2.1.5. WARN - used to log deprecated usage warnings
9.1.2.2.1.6. ERROR - used for unexpected exceptions
9.1.2.2.1.7. FATAL - used for unrecoverable crashes or hangs
9.1.2.2.2. Describe the Crash Capture available using MobileFirst Foundation Operational Analytics.
9.1.2.2.2.1. The MobileFirst client SDK, on Android and iOS applications, captures a stack trace upon application crash and logs it at FATAL level. This type of crash is a true crash where the UI disappears from the user‟s view. In Cordova applications, captures JavaScript global errors and if possible a JavaScript call stack, and logs it at FATAL level. This type of crash is not a crash event, and might or might not have any adverse consequences to the user experience at run time.
9.1.2.2.2.2. Crashes, uncaught exceptions, and global errors are caught, logged, and sent to the MobileFirst server automatically once the app is running again.
9.1.3. Understand the technology stack on which MobileFirst Foundation Operational Analytics is built.
9.1.3.1. MobileFirst Foundation Operational Analytics is built on top of the Open SourceSoftware, Elasticsearch 1.5x. Elasticsearch is a real-time distributed search and analytics engine that increases the speed and scale rates for data storage and exploration. Elasticsearch is used for full-text search, structured search.
9.1.3.2. Elasticsearch is used for storing all mobile and server data in JSON format in the Elasticsearch instances on the MobileFirst Analytics Server.
9.1.3.3. The Elasticsearch instances are queried in real-time to populate the MobileFirst Analytics Console.
9.1.3.4. MobileFirst Analytics exposes all Elasticsearch functionality. The user is able to take full advantage of Elasticsearch queries, debugging, and optimization.

REFERENCES:

1. https://mobilefirstplatform.ibmcloud.com/tutorials/en/foundation/8.0/analytics/
2. https://mobilefirstplatform.ibmcloud.com/tutorials/en/foundation/8.0/applicationdevelopment/client-side-log-collection/
3. For more information about Elasticsearch functionality, see the Elasticsearch documentation.
4. Forwarding logs to the Analytics Server

9.2. Enable access to raw and analytic reporting

SUBTASK(S):

9.2.1. Configure analytics on the server side

- 9.2.1.1. Open the mfpconsole in a web browser
- 9.2.1.2. Click on “Runtime Settings”
- 9.2.1.3. Click on “Edit”
- 9.2.1.4. Set “Data collection enabled” to “true”
- 9.2.1.5. Click on “Save”

9.2.2. Configure analytics on the client side

- 9.2.2.1. Import the corresponding libraries to initialize the analytics support
  - 9.2.2.1.1. For Cordova applications, no setup is required and initialization is built-in. 9.2.2.1.2. For Web applications:
    - 9.2.2.1.2.1. Add the MobileFirst Web SDK to your project, then include the Analytics .js file („ibmmfpfanalytics.js‟)
    - 9.2.2.1.2.2. Enable logging by running ibmmfpfanalytics.logger.config({analyticsCapture: true}); on Document ready.
  - 9.2.2.1.3. For iOS applications:
    - 9.2.2.1.3.1. Import the WLAnalytics library
    - 9.2.2.1.3.2. For Swift applications, call WLAnalytics.sharedInstance() before calling other methods of the WLAnalytics class.
  - 9.2.2.1.4. For Android applications:
    - 9.2.2.1.4.1. Import the WLAnalytics library
    - 9.2.2.1.4.2. Intialize Analytics inside the onCreate method of your main activity.

REFERENCES:

1. Adding the MobileFirst SDK to Web applications

9.3. Use the Analytics REST API through a browser or tool such as Postman

SUBTASK(S):

9.3.1. Search and view data by executing:
http://localhost:9500/<tennant‟s_name>/_search

9.3.2. View MobileFirst Analytics cluster health by executing:
http://localhost:9500/_cluster/health

9.3.3. View information on curent nodes http://localhost:9500/_nodes/jvm?pretty

9.3.4. View the current mappings http://localhost:9500/<tennant‟s_name>/_mapping

9.3.5. Import and export analytics data

REFERENCES:

1. https://mobilefirstplatform.ibmcloud.com/tutorials/en/foundation/8.0/analytics/elasticsearch/
2. https://mobilefirstplatform.ibmcloud.com/tutorials/en/foundation/8.0/analytics/analytics-rest-api/

9.4. Access client side crash logs

SUBTASK(S):

9.4.1. Understand the Crash Capture available using MobileFirst Foundation Operational Analytics.

9.4.1.1. The MobileFirst client SDK, on Android and iOS applications, captures a stack trace upon application crash and logs it at FATAL level. This type of crash is a true crash where the UI disappears from the user‟s view. In Cordova applications, captures JavaScript global errors and if possible a JavaScript call stack, and logs it at FATAL level. This type of crash is not a crash event, and might or might not have any adverse consequences to the user experience at run time.

9.4.1.2. Crashes, uncaught exceptions, and global errors are caught, logged, and sent to the MobileFirst server automatically once the app is running again.

9.4.2. View the crash logs

9.4.2.1. You can quickly see an app crashes Overview in the Dashboard section of the MobileFirst Analytics Console.

9.4.2.1.1. Open the Overview page of the Dashboard section

9.4.2.1.2. The Crashes bar graph shows a histogram of crashes over time.

9.4.2.2. Query specific crash logs in the MobileFirst Analytics Console:

9.4.2.2.1. Choose the Apps panel from the navigation bar

9.4.2.2.2. Click the Client Log Search tab.

9.4.2.2.3. Optionally filter by Application Name, Log Level, Operating System, or Package Name **REFERENCES:**

1. https://mobilefirstplatform.ibmcloud.com/tutorials/en/foundation/8.0/analytics/workflows/#app-crash-monitoring

9.5. Use MobileFirst Analytics to capture custom user analytics

SUBTASK(S):

9.5.1. Create a custom user analytics event

9.5.1.1. Call the following client side API methods to create custom events:

9.5.1.1.1. For Cordova apps call: WL.Analytics.log({"key" : 'value'});

9.5.1.1.2. For JavaScript web apps call: ibmmfpfanalytics.addEvent({"key" : 'value'});

9.5.1.1.3. For native Android apps call: WLAnalytics.log("Message",json);

9.5.1.1.4. For iOS Swift call: WLAnalytics.sharedInstance().log("hello", withMetadata: metadata)

9.5.2. Send analytics data to the server

9.5.2.1. Call the following client side API methods to send analytics data to the server:

9.5.2.1.1. For Cordova apps call: WL.Analytics.send();

9.5.2.1.2. For JavaScript web apps call: ibmmfpfanalytics.send();

9.5.2.1.3. For native Android apps call: WLAnalytics.send();

9.5.2.1.4. For iOS Swift call: WLAnalytics.sharedInstance().send()

9.5.3. To track individual users, use the „setUserContext" method

9.5.3.1. Call the following client side API methods to track individual users:

9.5.3.1.1. For native Android apps call:

WLAnalytics.setUserContext("John Doe");

9.5.3.1.2. For iOS Swift call:

WLAnalytics.sharedInstance().setUserContext("John Doe");

9.5.3.2. Call the following client side API methods to un-track individual users:

9.5.3.2.1. For native Android apps call: WLAnalytics.unsetUserContext();

9.5.3.2.2. For iOS Swift call:

WLAnalytics.sharedInstance().unsetUserContext

REFERENCES:

1. https://mobilefirstplatform.ibmcloud.com/tutorials/en/foundation/8.0/analytics/analytics-api/#custom-events

Next Steps

1. Take the IBM Mobile Foundation v8.0 Application Development assessment test. Use the promotion code **2018StudyAssess20** for $20 off each assessment.

2. If you pass the assessment exam, visit pearsonvue.com/ibm to schedule your testing sessions. Use the promotion code **2018StudyCert20** to receive 20% off the exam.

3. If you failed the assessment exam, review how you did by section. Focus attention on the sections where you need improvement. Keep in mind that you can take the assessment exam as many times as you would like ($10 per exam), however, you will still receive the same questions only in a different order.