ForgeRock Access Management Customization and APIs (AM-421 Rev B.2)

Description

This course provides a hands-on technical introduction to ForgeRock® Access Management (AM) APIs and customization use cases. Students examine AM extension points and gain the skills required to extend and integrate an AM deployment in a real-world context. Development best practices are demonstrated in a series of labs.

Note that Revision B.2 of this course is built on version 6.5.2 of AM.

Target Audiences

The following are the target audiences for this course:

- Application Developers, adapting client applications to use AM capabilities
- Software Developers, extending and integrating AM services for their organizations
- System Consultants
- System Architects

Objectives

Upon completion of this course, you should be able to:

- List the extension points of AM
- List which customizable components are affected in common AM use cases
- Understand the basic concepts of scripting
- Use the administration interface to look up, edit, and configure scripts
- Describe how AM performs authentication
- Review authentication nodes and authentication trees
- Design and implement a custom authentication node
- Describe how scripted authentication works
- Explore how client-side scripts are used with authentication nodes and trees
- Describe how server-side scripted authentication operates with authentication nodes and trees
- Use the administration interface to create and test authentication trees containing scripted nodes
- Discuss the policy concepts in AM
- Implement an Entitlement Condition or a scripted condition
- Describe the ForgeRock® Common REST API (Common REST)
- Enable Cross-Origin Resource Sharing (CORS) in AM
- Authenticate users through the REST API
- Manage identities and realms through the REST API
- Implement password reset and user self-registration by using the REST API
- Query the list of dashboard applications through the REST API

- Use the policy engine to protect non-URL-based resources
- Describe the policy management and evaluation REST APIs
- Describe OAuth 2.0 and OpenID Connect, including how to use their HTTP endpoints
- Demonstrate scope validation and customize the default behavior
- Explain the basic concepts of user-managed access (UMA)
- Configure AM as an UMA authorization server
- Manage UMA resource sets
- Demonstrate how to customize the UMA workflow

Prerequisites

The following are prerequisites to successfully completing this course:

- Completion of the AM-400 Rev B course
- Basic knowledge and skills using the Linux operating system to complete labs
- Knowledge of JSON, JavaScript, AngularJS, REST, Java, Groovy, and XML is important for mastering understanding of material and examples
- Basic knowledge of LDAP may be helpful for understanding code and some examples

Duration: 5 days

Course Contents

Chapter 1: Introducing Customization in AM

Introduce customization with AM and identify the main functional areas where customization and extending of AM is possible. The course environment and application are discussed as the context wherein customizations are done.

Lesson 1: Using Extension (Customization) Points

Provide an overview of AM extension points where customizations are done. Discuss the main components of the AM architecture and related APIs through which AM services can be accessed:

- Introduce Java APIs, REST API, and REST API versioning
- Introduce customizing authentication
- Introduce customizing authorization and policy evaluation
- Describe use cases related to OAuth 2.0 and UMA
- Describe use cases related to SAML2
- Describe the course environment architecture
- Understand the course ContactList application functionality and its role in this course
- Manage (starting, stopping) the AM and Directory Services servers
- Describe development tools and scripts provided with the course environment

Chapter 2: Customizing Authentication

Implement custom authentication services by using authentication trees and nodes provided by AM. Learn to create a custom authentication node and use the node in an authentication tree to provide authentication services for the ContactList application. Explore customization of authentication with client-side and server-side scripts. Cover migration of authentication modules and chains to authentication nodes and trees.

Lesson 1: Introducing Authentication Trees and Nodes

Learn to create an authentication tree comprised of several authentication nodes, provided with AM without any customization, as the proof-of-concept use case for the course Contact List application. Test the tree implementation within a web browser and use command-line REST API requests to examine the HTTP request-response and data information exchanged between the client web browser and AM:

- Review the concept of authentication trees and nodes
- Create a basic authentication tree
- Add existing authentication nodes to an authentication tree
- Implement a choice collector authentication node
- Assign the user choice to a session property
- Configure the Session Property Whitelist Service for the realm
- Test the authentication tree in a web browser and with the REST API
- Run a REST API function to view the authenticated user’s session data
- Compare tree and chain authentication methods

Lesson 2: Customizing with Authentication Trees and Nodes

Present the AM authentication node API to develop a custom authentication node for use in authentication trees. Implement a custom authentication node to replace the functionality of the choice collector, and to set session property nodes used in the initial authentication tree:

- Create a custom authentication node project using the Maven archetype from the command line
- Create a custom authentication node project using the Maven archetype within NetBeans
- Write the configuration interface for a custom authentication node
- Manage updates to the authentication node configuration interface
- Write the business logic for a custom authentication node
- Deploy a custom authentication node
- Modify an existing authentication tree to add the custom authentication node
- Test the custom authentication node using a web browser interface or its REST API

Lesson 3: Developing Scripts with Scripting APIs

Learn to execute client-side and server-side scripts in the context of an authentication tree. Explore how client-side scripts can be run by using a custom authentication node. Process client-side data with a server-side script created for use in a Scripted Decision node in an authentication tree:

- Explore client-side scripting with authentication nodes
- Deploy a custom authentication node that runs specific client-side scripts
- Include a client-side script with the custom authentication node in an authentication tree
- Create a script for use by a Scripted Decision node in an authentication tree to process the client-side data and return an authentication decision
- Receive and process data from the client-side script in a server-side script with a Scripted Decision node
- Understand client-side scripting with authentication trees by examining source code
- Configure the scripting engine properties and manage the APIs available to server-side scripts
- Test the script-based authentication with authentication trees and nodes

Lesson 4: Migrating Authentication Modules to Authentication Trees and Nodes

Explore the source code of a custom authentication module and chain implemented for AM versions prior to version 5.0 and the course application. Explore how it is migrated in this course to create custom authentication trees to meet the Contact List application requirements. Examine the use case with a client-side and server-side scripted module in a chain that is migrated for use with a custom authentication node (for the client-side script), and the standard Scripted Decision node (for the server-side scripts) to be implemented in authentication trees:

- Migrate a server-side authentication script to be used in a Scripted Decision node of an authentication tree
- Modify the server-side script to receive client-side data in the authentication tree context
- Design the server-side authentication script outcome values for use in the authentication tree
- Migrate a client-side authentication (module-based) script to be used by a custom authentication node
- Write the client-side logic to send client data to the custom authentication node in the context of an authentication tree

Chapter 3: Customizing Authorization

Create and test a set of policies enforcing the security constraints to enable users to access REST endpoints provided by the course Contact List application.

Lesson 1: Customizing Authorization

Learn to write and test a custom policy condition script (using JavaScript) which queries the maintenance mode state of the contact list application:

- Review the main elements of the AM policy API
- Discuss the concept of resource types and policy sets (formerly applications)
- Describe the concept of application types
- Illustrate the policy structure
- Review the main groups of built-in policy conditions and their important members
- Discuss where an Entitlement Condition and a script condition can be used
- Implement, build, and deploy an Entitlement Condition
- Implement, create, and deploy a scripted condition
- Review the execution flow of the scripted condition
- Discuss the variables available to the scripted condition
- Use a scripted condition through the administration interface and the REST API
- Develop a custom policy condition for the contact List application
- Modify the policy condition to return information about the maintenance mode
- Complete the policy set

Chapter 4: Customizing with REST Clients

Modify the sample contact List application's authentication mechanism to use the AM authentication tree service instead of its proprietary REST service.

Lesson 1: Using the REST API

Learn to access AM services though the REST API by using the REST API Explorer in the administration interface and in the contact List application written in AngularJS. Enable the CORS functionality in AM:

- Explore AM services available through the REST API
- Describe the ForgeRock Common REST API
- Review the main characteristics of the REST API
- Discuss the verbs available in the REST API
- Review the status codes returned by the REST API
- Describe filtering, paging, sorting, and pretty printing
- Explain the REST API versioning
- Access the REST API from the administration interface by using a web browser
- Use the REST API from jQuery
- Use the REST API from AngularJS
- Describe and enable CORS
- List the configuration options for the CORSFilter
- Configure the CORSFilter in AM
- Modify the contact List application to use AM for authentication
- Examine the client-side and server-side components of the contact List application
- Modify an AngularJS module in contact List that uses AM authentication services

Lesson 2: Authenticating with REST

Use the REST API to perform authentication with AM services implemented as authentication trees:

- Use the REST API to authenticate a user (sign in)
- Compare the simplified (username/password) and full authentication APIs
- Discuss application call back types
- Use the simplified and full authentication API
- Describe advanced authentication options (realm, authentication attributes, session upgrade)
- Use the REST API to log out
- Validate tokens and manage sessions
- Describe the session REST API
- Discuss the identity management REST API
- Read user attributes
- Create a realm
- Modify the ContactList application to use AM for all authentication functions
- Complete the AngularJS service interfacing AM to cover all authentication functions
- Modify the login service to use the testSelectRole authentication tree in AM

Lesson 3: Woking with RESTful User Self-Service API

Explore how to implement a password-reset function with the REST API:

- Review the characteristics of the self-service API
- Illustrate the flow of password reset
- Enable the password reset functionality
- Perform a password reset through the REST API
- Discuss the flow of user self-registration
- Enable the user self-registration functionality
- Perform user self-registration
- Describe the concept of a user dashboard
- List dashboard applications through the REST API
- Implement password reset in the ContactList application
- Configure AM to use a local email server
- Emulate password reset using the command line
- Add password reset functionality to the ContactList application

Lesson 4: Authorizing with REST

Learn to implement authorization in applications by using the REST API:

- Describe how to protect URL-based resources
- Explain how to protect non-URL-based resources
- List the main elements of the policy management API
- Discuss the entities of the policy service
- Describe the policy evaluation REST API
- Explain the concept of policy sets
- Request policy evaluation for a set of resources
- Demonstrate how policy evaluation can be used to determine which user interface components to show in a JavaScript client
- Modify the ContactList application to use AM for authorization
- Create and test policy sets tailored to the ContactList application
- Extend the backend of ContactList to use the authorization REST API
- Extend the front end of ContactList to use the authorization REST API

Chapter 5: Federating with OAuth 2.0

Learn how to federate a client application with AM using the OAuth 2.0/OpenID Connect protocol.

Lesson 1: Implementing OAuth Custom Scopes

Implement a Custom OAuth 2.0 Scope Validator:

- Explain the benefits of OAuth 2.0
- List the main elements of OAuth 2.0
- Illustrate the authorization code flow
- Describe the OAuth 2.0-related HTTP services available in AM
- Explain the benefits of OpenID Connect
- List the main elements of OpenID Connect
- Illustrate the authorization code flow extended with OpenID Connect
- Describe the Token Info endpoint
- Describe the User Info endpoint
- Discuss the OpenID Connect HTTP services
- Explain how scope validation is implemented in AM
- Implement and register a custom scope validation implementation
- Describe the default OpenID Connect script
- Create a custom OpenID Connect script
- Modify the ContactList application to use OAuth 2.0/ OpenID Connect for authentication and authorization
- Configure OAuth 2.0 and OpenID Connect in AM
- Create a customized scope validator and token response
- Modify the ContactList example application to use OpenID Connect for authentication
- Modify ContactList to behave as an OAuth 2.0 resource server

Chapter 6: Using User-Managed Access

Introduce the UMA architecture and the UMA flows, and use UMA to add sharing functionality to an OAuth 2.0-secured application. Implement an UMA-compatible resource server and implement an UMA client.

Lesson 1: Customizing with UMA

Implement contact group sharing by using UMA:

- Explain the benefits and list the elements of UMA
- Describe the various tokens and tickets used in UMA
- Illustrate the UMA protocol flow
- Enable and configure an UMA Provider in AM
- Configure UMA stores
- Use the UMA discovery endpoint
- Manage resources on the UMA administration page
- Understand the UMA REST API
- Describe the resource set and user label endpoints
- Discuss the policy endpoint
- Explain the permission request, requesting party token, and pending request endpoints
- Understand UMA customization points
- Register UMA filters
- Implement resource sharing in the example application