Release Notes

/ ForgeRock Identity Gateway 6.5

Latest update: 6.5.4

Mark Craig
Joanne Henry

ForgeRock AS
201 Mission St., Suite 2900
San Francisco, CA 94105, USA
+1 415-599-1100 (US)
www.forgerock.com

Abstract

Notes on prerequisites, fixes, and known issues for the ForgeRock® Identity Gateway.

Table of Contents

Preface

Identity Gateway (IG) integrates web applications, APIs, and microservices with the ForgeRock Identity Platform, without modifying the application or the container where they run. Based on reverse proxy architecture, IG enforces security and access control in conjunction with Access Management modules.

Chapter 1
What's New

1.1. Maintenance Releases

ForgeRock maintenance releases contain a collection of fixes and minor RFEs that have been grouped together and released as part of our commitment to support our customers. For general information on ForgeRock's maintenance and patch releases, see Maintenance and Patch Availability Policy.

IG 6.5.4 is the latest release targeted for IG 6.5.x deployments and can be downloaded from the *ForgeRock Backstage* website. To view the list of fixes in this release, see Key Fixes in IG 6.5.4.

The release can be deployed as an initial deployment or updated from an existing 6.5.x deployment.

1.2. New Features

What's New in IG 6.5.4

- **Hardeneded CSRF Protection**

  A new filter, CsrfFilter, is available to harden protection against CSRF attacks.

  For more information, see "Restricting Access to Studio" in the *Getting Started Guide*, and CsrfFilter(5) in the *Configuration Reference*.

What's New in IG 6.5.3

- **Support for All Name ID Formats In SAML SP-initiated SSO**

  In SAML SP-initiated SSO, IG can now act as an SP with an IDP that does not support the `transient` NameID Format. For SP-initiated SSO as well as for IDP-initiated SSO, the NameID Format can be any format supported by the IDP.

  In previous releases, for SP-initiated SSO, the NameID Format could be only `urn:oasis:names:tc:SAML:2.0:nameid-format:transient`.

  For more information, see "Using a Non-Transient NameID Format" in the *Gateway Guide*.

- **Spaces Maintained In Cookie Values**

As a cookie passes through IG, if the cookie value is not enclosed in quotes, spaces in the cookie value are not removed. In previous releases, spaces were removed.

- **Option to Reuse Connections After a Request**

  By default, IG tolerates characters that are disallowed in query string URL components, by applying a decode/encode process to the whole query string. **stateTrackingEnabled** is a new property of ClientHandler and ReverseProxyHandler to specify whether a connection can be kept open and reused after a request.

  For information, see ClientHandler(5) in the *Configuration Reference* or ReverseProxyHandler(5) in the *Configuration Reference*.

What's New in IG 6.5.2

- **Multiple OIDC Providers Can Use the Same clientID**

  In OpenID Connect with multiple client registrations, the same `clientId` can now be used for multiple client registrations if if the `issuerName` for each registration is different.

  The `clientId` must be unique in the context of a single issuer.

  In the OAuth2ClientFilter login service URI, specify both the `clientId` and the `issuerName`,

  For more information, see OAuth2ClientFilter(5) in the *Configuration Reference* and ClientRegistration(5) in the *Configuration Reference*, Issuer(5) in the *Configuration Reference*.

- **Request Policy Decisions From AM Using a Configurable Resource URL**

  In this release, IG can request policy decisions from AM by using the original request URL as the resource URL, or by using a script to generate the resource URL. In previous releases, IG could request policy decisions only by using the route `baseURI` as the resource URL.

  To reduce the cache size, query parameters can now be excluded from the resource URL.

  For more information, see the `"resourceUriProvider"` property of PolicyEnforcementFilter(5) in the *Configuration Reference*.

- **Infinite Loop Prevention in Single Sign-On**

  The SingleSignOnFilter has been adapted to prevent an infinite loop when a final redirect is returned without an AM session cookie name. The filter now checks the `goto` query parameter for the presence of the `_ig` marker parameter.

  For information, see SingleSignOnFilter(5) in the *Configuration Reference*, and CrossDomainSingleSignOnFilter(5) in the *Configuration Reference*.

- **SingleSignOnFilter Logout Can be Triggered By Any Aspect of a Request**

The new SingleSignOnFilter property `logoutExpression` can trigger logout based on any aspect of a request. Before this improvement, logout could be triggered only when a request matched the URI path.

For information, see SingleSignOnFilter(5) in the *Configuration Reference*.

What's New in IG 6.5.1

- **OAuth 2.0 Mutual TLS**

  IG now supports that ability for clients to authenticate to AM through OAuth 2.0 mutual TLS (mTLS) and X.509 certificates. You must use self-signed certificates or public key infrastructure (PKI), as per version 12 of the draft OAuth 2.0 Mutual TLS Client Authentication and Certificate Bound Access Tokens.

  For information about IG's support of Mutual TLS, see Access Token Resolvers in the *Configuration Reference*, and "*Acting as an OAuth 2.0 Resource Server*" in the *Gateway Guide*.

- **StatelessAccessTokenResolver can now rely on a SecretsProvider**

  A new heap object, SecretsProvider, is available to provide a secrets service for the StatelessAccessTokenResolver, that uses specified secret stores to resolve access_tokens.

  Before this improvement, the StatelessAccessTokenResolver used the global secrets service to resolve access_tokens, which searches for keys across the whole configuration. If multiple keys have the same label, there is a bigger risk that the wrong key is used.

  For backward compatibility, if SecretsProvider is not configured, the StatelessAccessTokenResolver uses the global secrets service.

  For information, see SecretsProvider(5) in the *Configuration Reference* and StatelessAccessTokenResolver(5) in the *Configuration Reference*.

- **Policy Enforcement Advice**

  If an AM policy decision denies a request with supported advices, the PolicyEnforcementFilter can now redirect the request to a URL specified in a SingleSignOnFilter, such as the URL of a custom login page. Previously, the filter always redirected the request back to AM.

  The URL is passed in a new property, `loginEndpoint`, in the `ssoToken` context. To use the redirect, configure `loginEndpoint` in the SingleSignOnFilter.

  For information, see SingleSignOnFilter(5) in the *Configuration Reference*.

- **New `toJSON` Function to Parse Strings as JSON**

  IG 6.5.1 provides a `toJSON` function that can be used in expressions to parse strings as JSON. For more information, see Functions(5) in the *Configuration Reference*.

- Preserve Query Strings In URLs

  A new property in `admin.json` allows you to preserve query strings as they are presented in URLs. Select this option when query strings must not change during processing, for example, in signature verification.

  By default, IG tolerates characters that are disallowed in query string URL components, by applying a decode/encode process to the whole query string.

  For information, see `preserveOriginalQueryString` in AdminHttpApplication(5) in the *Configuration Reference*.

What's New in IG 6.5.0

- **Commons Secret Service**

  IG now leverages the ForgeRock Commons Secrets Service for the management of passwords and secrets in the following objects: AmService, ClientHandler, ClientRegistration, JwtSession, KeyManager, JwtBuilderFilter, and CapturedUserPasswordFilter.

  Managing secrets with the Commons Secrets Service provides the following benefits:

  - Separation from other configuration so that configuration can be moved between environments
  - Storage in different secure backends, including file-based keystores, Hardware Security Modules (HSM), and Key Management Systems (KMS)
  - Provision through environment variables or unencrypted JSON, for deployment simplicity or where host/OS security is considered adequate.
  - Ease of rotation or revocation, regardless of the storage backend.

  In this release, routes generated in Studio do not use the Commons Secrets Service. Documentation examples generated with Studio use deprecated properties.

  For information about the SecretsService, see Secrets in the *Configuration Reference*. For information about new and deprecated properties, see "*Compatibility With Other Releases*".

- **Local Validation of Stateless Access-Tokens**

  The StatelessAccessTokenResolver is now available to validate stateless access_tokens without referring to AM. Use StatelessAccessTokenResolver with the access_token resolver in OAuth2ResourceServerFilter.

  Because IG can validate stateless access_tokens locally, without referring AM, this feature provides the following benefits:

  - Improved performance, by reducing the number of network hops required for validation
  - Improved robustness, by validating access_tokens even when AM is not available

Supported with OpenAM 13.5, and AM 5 and later versions.

For more information, see "Validating Stateless Access_Tokens With the StatelessAccessTokenResolver" in the *Gateway Guide* and StatelessAccessTokenResolver(5) in the *Configuration Reference*.

- **Transactional Authorization**

  IG can now respond to the `TransactionConditionAdvice` from AM to require users to perform additional actions when trying to access a resource protected by an AM policy.

  Performing the additional actions successfully grants a one-time access to the protected resource. Additional attempts to access the resource require the user to perform the additional actions again.

  Supported with AM 5.5 and later versions.

  For more information, see "*Hardening Authorization With Advice From AM*" in the *Gateway Guide*.

- **Disconnection Strategy WebSocket Notification Service**

  IG can now configure what happens to the session cache and policy enforcement cache when the WebSocket notification service is disconnected and then reconnected. By default, the caches are cleared on disconnect.

  For information, see `onNotificationDisconnection` in AmService(5) in the *Configuration Reference* and PolicyEnforcementFilter(5) in the *Configuration Reference*.

- **Dynamic Scope Evaluation for OAuth2ResourceServerFilter**

  The OAuth2ResourceServerFilter can now use a script to evaluate which scopes must be provided in an OAuth 2.0 access_token to access a protected resource. The script evaluates each request dynamically and returns the scopes that are required for the request to access the protected resource.

  Use this feature when protected resources can't be grouped within a set of static scopes, for example, when one set of URLs require one scope, and another set of URLs require another scope.

  For more information, see the `scopes` section and Examples section of OAuth2ResourceServerFilter(5) in the *Configuration Reference*.

- **JWT Encryption With JwtBuilderFilter**

  A new property, `encryption`, has been added to the JwtBuilderFilter to configure JWT encryption.

  For more information, see JwtBuilderFilter(5) in the *Configuration Reference*.

- **JwtBuilderFilter Template Declared as Expression**

  The `template` property of JwtBuilderFilter can now be configured as an expression that evaluates to a map. The referenced map will be serialized as a JSON object.

For more information, see JwtBuilderFilter(5) in the *Configuration Reference*.

- **Connection to TLS-Protected Endpoints With TlsOptions**

  A new object, TlsOptions, is available to configure connections to TLS-protected endpoints for the ClientHandler, ReverseProxyHandler, and for WebSocket notifications in AmService.

  For more information, see TlsOptions(5) in the *Configuration Reference*.

- **Increased Flexibility for Retrieving and Caching User Profiles From AM**

  The UserProfileFilter provides new features to retrieve and cache user profile information.

  For more information, see UserProfileFilter(5) in the *Configuration Reference*.

  - **User Authentication From OAuth 2.0 Access Tokens With UserProfileFilter**

    The UserProfileFilter can now retrieve AM profile attributes for users identified by their `username`, and can be used in routes that rely on OAuth2ResourceServerFilter and the `/oauth2/introspect` endpoint to resolve access tokens.

    The filter can use the SsoTokenContext, SessionInfoContext, or OAuth2Context to retrieve profile attributes.

  - **Cache for User Profile Attributes with UserProfileFilter**

    The UserProfileFilter can now cache user profile attributes and reuse them without repeatedly querying AM.

    In previous releases, the UserProfileFilter had to query AM for each request to retrieve the required user profile attributes.

- **Simplified Configuration of Objects by Using AmService Agent**

  A new property, `agent`, in AmService defines a Java agent to act on behalf of IG, and simplify configuration of the following filters:

  - SingleSignOnFilter, where `agent` defines the AM service to use for authentication. Users can authenticate in the same realm as the agent, or in a different realm.
  - PolicyEnforcementFilter, where `agent` defines the AM agent with the right to request policy decisions from AM. The policy set can be located in the same realm as the agent, or in a different realm.
  - TokenTransformationFilter, where `agent` defines the AM agent with the right to authenticate IG as an AM REST STS client.

  The `agent` property is now mandatory in AmService and replaces properties in the above filters. For more information, see "Removed Functionality".

For more information, see AmService(5) in the *Configuration Reference*.

- **Configuration of WebSocket Notifications by Using AmService**

  A new property, `notifications`, has been added to AmService to disable WebSocket notifications, configure the time between attempts to re-establish lost WebSocket connections, and to configure WebSocket connections to TLS-protected endpoints.

  For more information, see "WebSocket Notification Service" in the *Configuration Reference*.

- **UserProfileFilter Configuration Moved to AmService**

  To simplify configuration, properties in UserProfileFilter have been deprecated and replaced with properties in AmService.

  For more information, see Deprecated Functionality in IG 6.5.0.

- **StudioProtectionFilter to Restrict Access to Studio In Development Mode**

  A new filter, StudioProtectionFilter, is available to protect the Studio endpoint when IG is running in development mode.

  When IG is running in development mode, by default the Studio endpoint is open and accessible. When StudioProtectionFilter is defined in `admin.json`, IG uses it to filter access to the Studio endpoint.

  For an example configuration, see "Restricting Access to Studio in Development Mode" in the *Gateway Guide*. For more information about StudioProtectionFilter, see "Provided Objects" in the *Configuration Reference*.

- **New Features in Studio**

  New features have been added to the technology preview of Studio to allow you to:

  - Configure a SplunkAuditEventHandler.
  - Upgrade HTTP connections to WebSocket protocol.
  - Enable a session cache.
  - Evaluate scopes dynamically for OAuth 2.0 authorization.

- **New Features in Freeform Studio**

  New features have been added to the technology preview of Freeform Studio to allow you to:

  - Create new routes that contain a SingleSignOnFilter, a PolicyEnforcementFilter, and an example AmService. Select the objects to configure them.

- Drag and drop a SingleSignOnFilter, a PolicyEnforcementFilter, or any filter type onto the canvas. Select the filter to configure it. For other filter types, select the type, name the filter, and add the JSON configuration.
- Define multiple AmService objects that you can choose from for filters.
- Drag and drop a DispatchHandler onto the canvas, select its input node to connect it to the start element or another object, and select its output node to connect to one or more handlers. Select the connections to define the conditions for the dispatch.
- Drag any filter into or out of a chain, and drag any filter or handler around the canvas. Select it to delete it.
- Ctrl-click to select multiple objects, and maneuver or delete them at the same time.
- View unconnected filters or handlers on the canvas as part of the JSON heap.
- View the object name on the canvas.

Routes created in previous version of Freeform Studio are automatically transitioned into JSON editor routes.

1.3. Product Improvements

Improvements in IG 6.5.4

- There are no additional improvements in this release.

Improvements in IG 6.5.3

- Sample Application User Passwords Updated

  The user passwords in the Sample application files are updated to align with the new AM password policy. For more information, see OPENIG-4829: SampleApplication: Update user passwords.

Improvements in IG 6.5.2

- **Secure and HttpOnly Options to the JwtCookieSession Cookie Config**

  IG 6.5.2 now provides secure and httpOnly options on the cookie created for JWT sessions.

  For more information, see CrossDomainSingleSignOnFilter(5) in the *Configuration Reference*.

- **Warning If Decoded Secret Starts or Ends in a Non-ASCII Character**

IG logs a warning when the decoded value of a BASE64-encoded secret starts or ends with a non-ASCII character.

If a text editor adds a carriage return to the end of a plain string value before it is encoded, non-ASCII characters can be added to the BASE64-encoded value. When the decoded value is used as part of a username/password exchange, it can then cause an authentication error.

- **Support for SameSite Cookies**

  CrossDomainSingleSignOnFilter and JwtSession have a new property called `sameSite` to manage the circumstances in which a cookie is sent to the server. Use this property to manage the risk of cross-site request forgery (CSRF) attacks.

  For information, see the `authCookie` property of CrossDomainSingleSignOnFilter(5) in the *Configuration Reference*, or JwtSession(5) in the *Configuration Reference*.

- **Support for Applications Using a Payload in GET or HEAD Requests**

  The payload body of a GET or HEAD request is now honored. In previous releases, the payload body was removed when the internal Request representation was created. TRACE is the only request that does not support a payload.

- **Correct Maintenance of Cookies With `sameSite` Flag**

  Cookies that arrive at IG with the `sameSite` flag set are correctly maintained.

- **ResourceException Not Logged at Error Level when AM Returns 401**

  Previously, if the user's SSO session had expired or become otherwise invalid and was used in a request to IG, calling the AM session info endpoint to get session status would return a 401 response. This 401 response was valid but ended up being logged by IG at Error level, which was misleading, and would generate a large amount of additional logging data.

  IG now only logs an error message when the response from an AM session info endpoint is not a 401. IG still logs it as a debug message to show that it was a 401 response.

Improvements in IG 6.5.1

- There are no product improvements other than those listed in Improvements in IG 6.5.0 and What's New in IG 6.5.1.

Improvements in IG 6.5.0

- **TimerDecorator Publishes Metrics to the MetricRegistry**

  When a TimerDecorator is set to `true` in a route, the metrics are now written to the Prometheus Scrape Endpoint and the ForgeRock Common REST Monitoring Endpoint.

For information, see TimerDecorator(5) in the *Configuration Reference*.

- **Audit Logging to Standard Output**

  Support has been added for an audit handler to send access log messages to standard output.

  For information, see JsonStdoutAuditEventHandler(5) in the *Configuration Reference* and "Recording Audit Events to Standard Output" in the *Gateway Guide*.

- **Default Configurations for Objects In AdminHttpApplication**

  AdminHttpApplication now declares default configurations for the following objects: ClientHandler, ReverseProxyHandler, ForgeRockClientHandler, ScheduledThreadPoolExecutor, and TransactionIdOutboundFilter.

  For more information, see AdminHttpApplication(5) in the *Configuration Reference*.

- **Improved Security for Authentication Cookies in CrossDomainSingleSignOnFilter and JwtSession**

  By default, the JwtCookieSession cookie and CrossDomainSingleSignOnFilter authentication cookie and are now flagged as `HttpOnly`.

  CrossDomainSingleSignOnFilter has additional properties to set or unset cookie flags for `HttpOnly` and `secure`. For more information, see CrossDomainSingleSignOnFilter(5) in the *Configuration Reference*.

- **WebSocket Traffic for TLS Connections**

  IG can now detect requests to upgrade from HTTPS to the WebSocket protocol, and create a secure, dedicated tunnel to send and receive WebSocket traffic.

  For information, see the `websocket` property of ClientHandler(5) in the *Configuration Reference* or ReverseProxyHandler(5) in the *Configuration Reference*.

1.4. Security Advisories

ForgeRock issues security advisories in collaboration with our customers and the open source community to address any security vulnerabilities transparently and rapidly. ForgeRock's security advisory policy governs the process on how security issues are submitted, received, and evaluated as well as the timeline for the issuance of security advisories and patches.

For details of all the security advisories across ForgeRock products, see Security Advisories in the *Knowledge Base library*.

Chapter 2
Before You Install

This chapter describes the requirements for running IG.

Tip

If you have a request to support a component or combination not listed here, contact ForgeRock at info@forgerock.com.

2.1. Downloading IG Software

Download the following product software from the ForgeRock BackStage download site:

- IG .war file, `IG-6.5.4.war`
- Web application for testing IG configurations, `IG-sample-application-6.5.4.jar`

2.2. Java Requirements

The following table lists supported Java versions:

JDK Requirements

Vendor	Versions
Oracle JDK	8
OpenJDK	8, 11

If you are using IG on Tomcat with SSL enabled, to prevent mismatch between client-side ciphers and server-side ciphers, use OpenJDK 1.8.0_121 or later versions.

For the latest security fixes, ForgeRock recommends that you use the most recent update.

2.3. Web Application Containers

IG runs in the following web application containers:

- Apache Tomcat 8.5.x or 9
- Jetty 9
- JBoss EAP 7.2

Deploy IG to the root context of the container. Deployment in other contexts causes unexpected results, and is not supported.

For information about setting up a web application container see "Configuring Deployment Containers" in the *Gateway Guide*.

2.4. AM Java Agents

IG supports several versions of Java Agents. For supported container versions and other platform requirements related to agents, see the *Java Agents Release Notes* .

If you install Java Agents in the same container as IG, use a Java release that is also supported by the agent.

If you install an AM policy agent in the same container as IG, use Java Agents 3.5 or later. Earlier versions might not shut down properly with the web application container.

You cannot run Java Agents 5.5.0 and IG in the same Tomcat container.

2.5. Features Supported With ForgeRock Access Management

This section describes the IG features that are supported with AM:

Features Supported With AM

Feature	Supported In AM Version
Support for OAuth 2.0 Mutual TLS (mTLS). For more information, see ConfirmationKeyVerifierAccessTokenResolver(5) in the *Configuration Reference*, and "Validating Access_Tokens Obtained Through mTLS" in the *Gateway Guide*.	AM 6.5.1 and later versions.
Eviction of entries from the AmService `sessionCache`, using WebSocket notifications from AM. For more information, see AmService(5) in the *Configuration Reference*.	AM 5.5 when the user manually whitelists the `AMCtxId` session property, and with AM 6 and later versions (where the `AMCtxId` session property is whitelisted by default).
AM password capture and replay, as described in "*Getting Login Credentials From AM*" in the *Gateway Guide*.	Supported with AM 5 and later versions, and with AM 6 and later versions when the `AES` keyType is used to decrypt the password.

Feature	Supported In AM Version
AM policy enforcement, as described in "*Enforcing Policy Decisions From AM*" in the *Gateway Guide*.	AM 5 and later versions
OpenID Connect dynamic registration and discovery, as described in "Using OpenID Connect Discovery and Dynamic Client Registration" in the *Gateway Guide*.	OpenAM 13.5, and AM 5 and later versions
Token transformation, as described in "*Transforming OpenID Connect ID Tokens Into SAML Assertions*" in the *Gateway Guide*.	OpenAM 13.5, and AM 5 and later versions
User Managed Access 2.x, for IG 5.5, as described in "*Supporting UMA Resource Servers*" in the *Gateway Guide*.	AM 5.5 and later versions
User Managed Access 1.x, for IG 5 and earlier versions.	AM 5.1 and earlier versions
Single sign-on, as described in "About SSO Using the SingleSignOnFilter" in the *Gateway Guide*.	AM 5 and later versions
Cross-domain single sign-on, as described in "About CDSSO Using the CrossDomainSingleSignOnFilter" in the *Gateway Guide*.	AM 5.5 and later versions
Capture and storage of AM session information, as described in SessionInfoFilter(5) in the *Configuration Reference*.	AM 5 and later versions
Capture and storage of AM user profile attributes, as described in UserProfileFilter(5) in the *Configuration Reference*.	AM 5 and later
Support for transactional authorization, as described in "*Hardening Authorization With Advice From AM*" in the *Gateway Guide*.	AM 5.5 and later versions
Validation of stateless access_tokens, as described in "Validating Stateless Access_Tokens With the StatelessAccessTokenResolver" in the *Gateway Guide*.	OpenAM 13.5, and AM 5 and later versions

2.6. Third-Party Software Required for Encryption

To use RSASSA-PSS for signature encryption in the JwtBuilderFilter, install Bouncy Castle. For information, see *The Legion of the Bouncy Castle*.

Chapter 3

Compatibility With Other Releases

This chapter describes major changes to existing functionality, deprecated functionality, and removed functionality.

3.1. Important Changes to Existing Functionality

Important Changes in IG 6.5.4

- Validation of `goto` Parameter in OAuth2ClientFilter

  To prevent redirects to malicious web sites, IG now validates the `goto` query parameter in requests to OAuth2ClientFilter `/login` and `/logout` endpoints.

  The goto URL must use the same scheme, host, and port as the original URI, or be a relative URI (just the path). Otherwise, the request fails with an error.

  For more information, see OAuth2ClientFilter(5) in the *Configuration Reference*.

Important Changes in IG 6.5.3

- SAML 2.0 Deployments Require Additional Configuration

  When IG uses AM federation libraries generated from AM 6.5.2 or earlier, add the following lines to the `FederationConfig.properties` file:

```
# Specifies implementation for
# org.forgerock.openam.federation.plugin.rooturl.RootUrlProvider interface.
# This property defines the default base url provider.
com.sun.identity.plugin.root.url.class.default=org.forgerock.openam.federation.plugin.rooturl.impl
.FedletRootUrlProvider
```

Important Changes in IG 6.5.2

- There are no additional changes to existing functionality in this release.

Important Changes in IG 6.5.1

- See What's New in IG 6.5.1 for a list of important changes to existing functionality.

Important Changes in IG 6.5.0

- **Agent Credentials Mandatory in AmService**

  The `agent` property of AmService is now mandatory. The agent defines the credentials of an AM Java agent that acts on behalf of IG to authenticate with AM, request policy decisions from AM, and communicate WebSocket notifications from AM to IG.

  This is a breaking change for all filters that use AmService, and for the following filters where `agent` replaces properties that are removed in this release:

  - SingleSignOnFilter, where `agent` replaces previously deprecated properties.
  - PolicyEnforcementFilter, where `agent` replaces previously deprecated properties and the following properties: `pepUsername` and `pepPassword`.
  - TokenTransformationFilter, where `agent` replaces previously deprecated properties and the following properties: `username` and `password`.

  For more information, see Removed Functionality in IG 6.5.0.

- **Agent Session Logged Out When AmService Stopped**

  When a route containing an AmService is reloaded, or when an AmService is stopped, the agent session is logged out.

  For more information, see org.forgerock.openig.tools.am.AmService.

- **Disconnection Strategy for Session Cache and PolicyEnforcementFilter Cache**

  When the WebSocket notification service is disconnected, by default the session cache and policy enforcement cache is cleared. In previous releases, the caches were not cleared.

  For information, see `onNotificationDisconnection` in AmService(5) in the *Configuration Reference* and PolicyEnforcementFilter(5) in the *Configuration Reference*.

- **DS API Change for Secure LDAP Connection**

  DS 6.5 has updated its client API for establishing SSL connections. The `SslContextBuilder` class has been removed and related usages have been integrated into `SslOptions`.

  This has an impact on existing scripts that are using IG's `LdapClient` for connecting to a secure LDAP server.

  Previously working script:

```
import org.forgerock.opendj.security.SslContextBuilder;
//...
SslContextBuilder builder = new SslContextBuilder();
builder.trustManager(TrustManagers.trustAll());
SslOptions sslOptions = SslOptions.newSslOptions(builder.build())
.enabledProtocols("TLSv1.2");
```

Usage of the new API:

```
SslOptions sslOptions = SslOptions.newSslOptions(null, TrustManagers.trustAll())
.enabledProtocols("TLSv1.2");
```

3.2. Deprecated Functionality

Deprecated Functionality in IG 6.5.4

- There is no additional functionality deprecated in this release.

Deprecated Functionality in IG 6.5.3

- There is no additional functionality deprecated in this release.

Deprecated Functionality in IG 6.5.2

- There is no additional functionality deprecated in this release.
- The OpenAmAccessTokenResolver property `endpoint` was deprecated in a previous release, but is undeprecated in this release. When you are not using AM SSO or policy components, use `endpoint` with OpenAmAccessTokenResolver instead of creating an AM agent and maintaining credentials.

Deprecated Functionality in IG 6.5.1

Deprecated Configuration Settings

Configuration Object	Deprecated Settings	Replacement Settings
StatelessAccessTokenResolver	`signatureSecretId`	Replaced by `verificationSecretId`.
	`encryptionSecretId`	Replaced by `decryptionSecretId`.

Deprecated Functionality in IG 6.5.0

Automatically Transfered Upgrade Routes

During IG upgrade, routes that were previously created in Studio are automatically transferred to the new version of IG. Where possible, IG replaces deprecated settings with the newer evolved setting. If IG needs additional information to upgrade the route, the route status becomes ⚠ Compatibility update required. Select the route, and provide the requested information.

Routes Generated in Studio Do Not Use the Commons Secrets Service

In this release, routes generated in Studio do not use the Commons Secrets Service. Documentation examples generated with Studio use deprecated properties.

IG Route Monitoring Endpoint

The IG Route Monitoring Endpoint is deprecated and will be removed in a later release. As a replacement, IG provides Prometheus Scrape Endpoint and Common REST Monitoring Endpoint.

For more information, see "Prometheus Scrape Endpoint" in the *Gateway Guide*, and "Common REST Monitoring Endpoint" in the *Gateway Guide*,

Support for .war File Delivery

The delivery of a .war file is deprecated in this release and may be removed in the next release.

Support AM Policy Agents

Support for the use of AM policy agents in password capture and replay is deprecated in this release.

By using CapturedUserPasswordFilter, you can get login credentials from AM without setting up an AM policy agent. For more information, see "*Getting Login Credentials From AM*" in the *Gateway Guide*, and CapturedUserPasswordFilter(5) in the *Configuration Reference*.

Deprecated Configuration Settings

Configuration Object	Deprecated Settings	Replacement Settings
AmService	`password`	Replaced by `passwordSecretId`. If the deprecated and replacement properties are both provided, the replacement property takes precedence.
ClientHandler	`proxy` subproperty `password`	Replaced by `passwordSecretId`. If the deprecated and replacement properties are both provided,

Configuration Object	Deprecated Settings	Replacement Settings
		the replacement property takes precedence.
	• `keyManager` • `sslCipherSuites` • `sslContextAlgorithm` • `sslEnabledProtocols` • `trustManager`	Replaced by the TlsOptions object. For more information, see TlsOptions(5) in the *Configuration Reference*.
	`websocket` subproperties: • `keyManager` • `sslCipherSuites` • `sslContextAlgorithm` • `sslEnabledProtocols` • `trustManager`	Replaced by the TlsOptions object. For more information, see TlsOptions(5) in the *Configuration Reference*.
ReverseProxyHandler	• `keyManager` • `sslCipherSuites` • `sslContextAlgorithm` • `sslEnabledProtocols` • `trustManager`	Replaced by the TlsOptions object. For more information, see TlsOptions(5) in the *Configuration Reference*.
	`websocket` subproperties: • `keyManager` • `sslCipherSuites` • `sslContextAlgorithm` • `sslEnabledProtocols` • `trustManager`	Replaced by the TlsOptions object. For more information, see TlsOptions(5) in the *Configuration Reference*.
JwtSession	`password`	Replaced by `passwordSecretId` If the deprecated and replacement properties are both provided, the replacement property takes precedence.
	Combination of `password`, `alias`, and `keystore`	Replaced by `encryptionSecretId`

Configuration Object	Deprecated Settings	Replacement Settings
	Combination of `passwordSecretId`, `alias`, and `keystore`	If the deprecated and replacement properties are both provided, the replacement property takes precedence.
	`sharedSecret`	Replaced by `signatureSecretId` If the deprecated and replacement properties are both provided, the replacement property takes precedence.
KeyManager	`password`	Replaced by `passwordSecretId`. If the deprecated and replacement properties are both provided, the replacement property takes precedence.
KeyStore	`password`	Replaced by `passwordSecretId`. If the deprecated and replacement properties are both provided, the replacement property takes precedence.
CapturedUserPasswordFilter	`key`	Replaced by `keySecretId`. If the deprecated and replacement properties are both provided, the replacement property takes precedence.
JwtBuilderFilter	`signature` subproperties: • `keystore` • `alias` • `password`	Replaced by `signature` property `secretId`. If the deprecated and replacement properties are both provided, the replacement property takes precedence.
Route	`monitor`	Replaced by the Prometheus Scrape Endpoint and Common REST Monitoring Endpoint. For information, see Monitoring Endpoints(5) in the *Configuration Reference*.
UserProfileFilter	`ssoToken`	Replaced by `username` in UserProfileFilter.
	`amService` and `profileAttributes`	Replaced `amService` and `profileAttributes`, as sub-properties of `userProfileService`

Configuration Object	Deprecated Settings	Replacement Settings
ClientRegistration	`keyStore`	Replaced by `keystore`.
	`clientSecret`	Replaced by `clientSecretId`. If the deprecated and replacement properties are both provided, the replacement property takes precedence.
The environment variable and system property that define the file system directory for configuration files.	`OPENIG_BASE` and `openig.base`	Replaced by `IG_INSTANCE_DIR` and `ig.instance.dir`. If neither the deprecated setting nor the replacement setting are provided, configuration files are in the default directory `$HOME/.openig` (on Windows, `%appdata%\OpenIG`). If the deprecated setting and the replacement setting are both provided, the replacement setting is used.
OpenAmAccessTokenResolver	`endpoint`	Replaced by the AmService property `url`. For information, see OpenAmAccessTokenResolver in OAuth2ResourceServerFilter(5) in the *Configuration Reference*.
PolicyEnforcementFilter	`cache` subproperty `maxTimeout`	Replaced by `cache` property `maximumTimeToCache`.
OAuth2ResourceServerFilter	`cacheExpiration`	Replaced by `cache` and its sub-properties `enabled`, `defaultTimeout`, and `maxTimeout`. If `cacheExpiration` is configured and `cache` is not configured, the cache is enabled and the value of `cacheExpiration` is used as `maxTimeout`. The following values for `cacheExpiration`, supported in previous releases, are not supported in this release: `zero`, `unlimited`. For more information, see OAuth2ResourceServerFilter(5) in the *Configuration Reference*.

3.3. Removed Functionality

Removed Functionality in IG 6.5.4

- There is no additional functionality removed from this release.

Removed Functionality in IG 6.5.3

- There is no additional functionality removed from this release.

Removed Functionality in IG 6.5.2

- There is no additional functionality removed from this release.

Removed Functionality in IG 6.5.1

- There is no additional functionality removed from this release.

Removed Functionality in IG 6.5.0

This section lists removed functionality, as defined in "ForgeRock Product Interface Stability":

-

Removed Configuration Settings

Configuration Object	Removed Settings	Newer Evolving Settings
PolicyEnforcementFilter(5) in the *Configuration Reference*	Deprecated previously, removed in this release: • `amHandler` • `openamUrl` • `realm` • `ssoTokenHeader`	Replaced by AmService properties: • `amHandler` • `url` • `realm` • `ssoTokenHeader`
	Deprecated and removed in this release: • `pepUsername` • `pepPassword`	Replaced by AmService property: • `agent`
SingleSignOnFilter(5) in the *Configuration Reference*	Deprecated previously, removed in this release: • `amHandler`	Replaced by AmService properties: • `amHandler` • `url`

Configuration Object	Removed Settings	Newer Evolving Settings
	• `openamUrl` • `realm` • `ssoTokenHeader`	• `realm` • `ssoTokenHeader`
TokenTransformationFilter(5) in the *Configuration Reference*	Deprecated previously, removed in this release: • `amHandler` • `openamUrl` • `realm` • `ssoTokenHeader`	Replaced by AmService properties: • `amHandler` • `url` • `realm` • `ssoTokenHeader`
	Deprecated and removed in this release: • `username` • `password`	Replaced by AmService property: • `agent`

Chapter 4

Fixes, Limitations, and Known Issues

IG issues are tracked at https://bugster.forgerock.org/jira/browse/OPENIG. This chapter covers the status of key issues and limitations at release 6.5.

4.1. Key Fixes

Key Fixes in IG 6.5.4

- OPENIG-4034: AuditService does not delete old files when maxDiskSpaceToUse is reached
- OPENIG-5084: WebSocket connections are not being proxied when baseURI scheme is wss
- OPENIG-5268: IG 6.5.3 Studio UI Welcome screen has formatting/layout issues

Key Fixes in IG 6.5.3

- COMMONS-580: Race condition on Tomcat regarding closure of ServletInputStream
- COMMONS-608: Http Servlet never uses the HttpApplication Buffer Factory
- OPENIG-3783: ClassCastException in scriptable access token resolver occurs when invalid token is returned by delegated access token resolver
- OPENIG-3934: Error with AMService reconnection notifications
- OPENIG-4168: CacheAccessTokenResolver : missing requests to amService (not available in capture)
- OPENIG-4186: UI: Update Bootstrap to 3.4.1
- OPENIG-4190: A WebSocket Origin header is missing the scheme from the URL
- OPENIG-4216: Pickup COMMONS-533 - allow spaces in unquoted cookie values
- OPENIG-4244: Error when loading route of the GatewayGuide (Jetty & AmService)
- OPENIG-4376: Copyright is not up to date on IG welcome page
- OPENIG-4391: SequenceHandler should cater for bindings generating RuntimeException responses

- OPENIG-4405: A missing AMCtxId session property is not logged leading to an invalid session cache
- OPENIG-4653: UI: Unable to edit imported route

Key Fixes in IG 6.5.2

- CHF-159: Default port(:443/:80) is always present in the originalUri
- CHF-200: HttpFrameworkServlet does not cope with content lengths > java.lang.Integer#MAX_VALUE when creating the CHF Request
- CHF-208: Provide support for GET and HEAD request to include a payload (entity)
- COMMONS-511: Add support for SameSite=None cookies
- OPENIG-2568: Add a switch to the PolicyEnforcementFilter to enable policy requests based on original URI rather than the baseURI
- OPENIG-2982: Infinite loop with SingleSignOnFilter when cookie domain does not match
- OPENIG-3146: Provide secure and httpOnly options to the config of the JwtCookieSession cookie
- OPENIG-3296: UserProfileFilter and usernames with colons
- OPENIG-3492: Request and response logged in different files when capture:all and global captureDecorator are in config.json
- OPENIG-3647: Generate warning when using Base64 based secrets if the decoded secret ends in non-ASCII value
- OPENIG-3659: SSOFilter logoutEndpoint does not take query parameters into consideration
- OPENIG-3734: Support SameSite cookie attribute
- OPENIG-3819: WebSocket requests should be built using the raw query parameters
- OPENIG-3837: WebSocketAdapter#writeBuffersIfStreamIsReady should check if stream is ready before calling flush
- OPENIG-3941: Provide support for GET and HEAD request to include a payload (entity)
- OPENIG-4037: Global decorators declared in a route cannot refer to decorators declared in the same route
- OPENIG-4046: Multiple OIDC providers for same clientID (OAuth2ClientFilter)
- OPENIG-4082: Don't log ResourceException at error level when AM returns 401 during CrestSessionService#getSessionInfo

Key Fixes in IG 6.5.1

- OPENIG-3328: CDSSOFilter : although using a valid token, user can't access the protected resource

- OPENIG-3403: ContentTypeHeader quoted directives should be maintained
- OPENIG-3443: Don't attempt to create the groovy script directories if they already exist
- OPENIG-3454: StatelessAccessTokenResolver: incorrect usage of COMMONS Secrets API to get the keys
- OPENIG-3457: Provide a toJson function that can be used in expressions to parse strings as JSON
- OPENIG-3484: IdTokenValidationFilter reports problem with iat with valid token
- OPENIG-3491: Invalid token is returned when signature verification is enabled in OAuth2ResourceServer filter
- OPENIG-3523: UI: Fix open studio in IE

Key Fixes in IG 6.5.0

- OPENIG-3231: OpenDJ SslContextBuilder has been removed
- OPENIG-3219: When using scan feature in logback.xml the ig.instance.dir property is lost on reload
- OPENIG-3113: Not possible to use token substitutions within a monitor decorator of a Route

4.2. Limitations

Limitations in IG 6.5.4

- OPENIG-3248: IG fails to proxy websocket messages when running with Jetty
- OPENIG-2417: Thread starvation on CHF response reception/consumption with async http client
- OPENIG-1557: UI: Unable to deploy route when custom router is configured
- OPENIG-813: auditService : fileRotation may overwrite existing audit file
- OPENIG-291: Class cast exception when using SAML federation & policy agent together
- OPENIG-234: Federation doesn't work if we used incomplete user in IDP
- OPENIG-221: Cannot specify which certificate to present to server if server requires mutual authentication in https

Limitations in IG 6.5.3

- There are no new limitations in this release.

Limitations in IG 6.5.2

The following limitations exist in this release in addition to those in IG 6.5.0.

- OPENIG-3245: Inconsistent error messages on negative testcases on JWT page in sample app
- OPENIG-3262: OAuth2ResourceServerFilter: realm is not taken into account
- OPENIG-3404: Websocket disconnects in Sample app after sending a message with odd count of regular ascii char with JBoss
- OPENIG-3413: Websockets in JBoss doesn't work properly for multiple mesages from server to client
- OPENIG-3861: UI: Generic filters cannot move correctly outside of a chain in freeform editor
- OPENIG-3925: UMA: Sharing resource via IG auto generates resource name
- OPENIG-4008: Improve debug logging when baseUri hostname is invalid
- OPENIG-4034: AuditService does not delete old files when maxDiskSpaceToUse is reached

Limitations in IG 6.5.1

- There are no new known limitations in IG 6.5.1, other than those identified in Limitations in IG 6.5.0.

Limitations in IG 6.5.0

- **SamlFederationHandler Doesn't Support Filtering (OPENIG-3275)**

  The SamlFederationHandler does not support filtering. Do not use a SamlFederationHandler as the handler for a Chain.

  More generally, do not use this handler when its use depends on something in the response. The response can be handled independently of IG, and can be `null` when control returns to IG. For example, do not use this handler in a `SequenceHandler` where the `postcondition` depends on the response.

- **IG Scripts Can Access Anything in Their Environment (OPENIG-3274)**

  IG scripts are not sandboxed, but instead have access to anything in their environment. You must make sure that the scripts that IG loads are safe.

- **Persists UMA Shares (OPENIG-3273)**

  Shared resources cannot be persisted when IG restarts. They must be shared each time that IG restarts. For more information, see "*Supporting UMA Resource Servers*" in the *Gateway Guide*.

- **Proxy WebSocket Traffic (OPENIG-3248)**

  When IG is running in the Jetty application container, it cannot proxy WebSocket traffic.

  For more information, see "*Proxying WebSocket Traffic*" in the *Gateway Guide*, and the `websocket` property of ClientHandler(5) in the *Configuration Reference* or ReverseProxyHandler(5) in the *Configuration Reference*.

- **Blocked ClientHandler With Asynchronous HTTP Clients (OPENIG-2417)**

  IG processes responses from asynchronous HTTP clients by using two thread pools of the same size:

  - the first thread pool receive the response headers,
  - the second thread pool completes the promise by to executing the callback and writing the response content to the stream. Reading and writing to the stream are synchronous, blocking operations

  When there are a lot of clients, or when responses are big, the synchronous operation can cause routes to declare a blocked ClientHandler.

  To recover from blocking, restart the route, or, if the route is `config.json`, restart the server. To prevent blocking, increase the number of worker threads.

- **Cannot Use Custom `config.json` in Studio (OPENIG-1557)**

  When a customized `config.json` is configured in Studio, Studio cannot deploy routes.

- **Log File of Audit Events Can be Overwritten (OPENIG-813)**

  The log file of audit events can be overwritten when the log file is rotated.

  When `CsvAuditEventHandler` is used to log audit events, the log file is overwritten if it is rotated before the file suffix, `rotationFileSuffix`, changes. By default, `rotationFileSuffix` is defined as a date in the format `_yyyy-MM-dd`.

  Log files are rotated when one of the following limits is reached: `maxFileSize`, `rotationInterval`, or `rotationTimes`.

  Set the log rotation parameters so that the log is not likely to rotate before `rotationFileSuffix` changes.

- **Cannot Use SAML With AM Policy Agent (OPENIG-291)**

  When SAML is used with an AM policy agent, class cast exceptions occur.

- **SAML Fails With Incorrect User-Defined Mapping (OPENIG-234)**

  When the user defined mapping is incorrectly set, missing SAML assertions produce an infinite loop during authentication attempts.

- **For Mutual Authentication in HTTPS Cannot Specify Which Certificate to Present (OPENIG-221)**

  IG can check server certificates for HTTPS. However, for mutual authentication, the client certificate must be the first certificate in the KeyStore.

4.3. Known Issues

Known Issues in IG 6.5.4

- OPENIG-3821: ResourceHandler should create redirect to a relative URI when requests don't end in /
- OPENIG-3755: IG's decodeBase64 function returns null on JWTs generated by IG or AM
- OPENIG-3579: NullPointerException when calling org.forgerock.openig.handler.router.DirectoryMonitor#createFileChangeSet
- OPENIG-3488: IG fails to stop when started with a config.json with invalid json syntax.

Known Issues in IG 6.5.3

- OPENIG-659: CryptoHeaderFilter - error on handling header value with incorrect length
- OPENIG-3488: IG fails to stop when started with a config.json with invalid json syntax
- OPENIG-3755: IG's decodeBase64 function returns null on JWTs generated by IG or AM

Known Issues in IG 6.5.2

- OPENIG-3488: IG fails to stop when started with a config.json with invalid json syntax
- OPENIG-3755: IG's decodeBase64 function returns null on JWTs generated by IG or AM
- OPENIG-3783: ClassCastException in scriptable access token resolver occurs when invalid token is returned

Known Issues in IG 6.5.1

- There are no new known issues in IG 6.5.1, other than those identified in Known Issues in IG 6.5.0.

Known Issues in IG 6.5.0

- OPENIG-3235: Support UTF-8 encoded password values for agent's credentials
- OPENIG-3221: OpenIG is decoding special character ' while sending to the backend which is causing issues

- OPENIG-659: CryptoHeaderFilter - error on handling header value with incorrect length

Chapter 5
Documentation Changes

Documentation Change Log

Date	Description
March 2022	A list of audit fields was added to AuditService(5) in the *Configuration Reference*
March 2021	Release of IG 6.5.4.
September 2020	Release of IG 6.5.3: • Information about tuning deployments is added in a new chapter, "*Tuning Performance*" in the *Gateway Guide*.
2020-07-01	(For AM 6.5.3 and later versions.) Procedures that use AM to redirect the request to the sample app have an additional step to configure an AM Validation Service.
2020-05-04	Update to "Configuring IG for HTTPS (Server-Side) in Jetty" in the *Gateway Guide*.
2020-04-15	Correction to the `AMCtxId` property name. Correction in SAML routes with multiple service providers.
2019-12-02	Release of IG 6.5.2. The following documentation updates were made: • Added a new property information that can request policy decisions from AM using a configurable resource URL. For more information see the `resourceUriProvider` property of PolicyEnforcementFilter(5) in the *Configuration Reference*. • Added `secure` and `httpOnly` options to the `JwtCookieSession` cookie. For more information, see CrossDomainSingleSignOnFilter(5) in the *Configuration Reference*. • Added a new property called `sameSite` for `CrossDomainSingleSignOnFilter` and `JwtSession` to manage the risk of cross-site request forgery (CSRF) attacks. For information, see the `authCookie` property of CrossDomainSingleSignOnFilter(5) in the *Configuration Reference*, or the `cookie` property of JwtSession(5) in the *Configuration Reference*. • Added infinite loop prevention for single sign-on. For more information, see SingleSignOnFilter(5) in the *Configuration Reference* and CrossDomainSingleSignOnFilter(5) in the *Configuration Reference*.
2019-05-28	Minor corrections in OAuth2ResourceServerFilter and SingleSignOnFilter.
2019-03-10	Release of IG 6.5.1 maintenance release.

Date	Description
2018-11-30	Release of IG 6.5.0 release. The following changes were made to the documentation: • The default configuration of IG, provided by when your configuration does not include a custom `config.json` file, is now described in the Examples section of GatewayHttpApplication(5) in the *Configuration Reference*. • Information about session upgrade has moved from "*Enforcing Policy Decisions From AM*" in the *Gateway Guide* to the new chapter "*Hardening Authorization With Advice From AM*" in the *Gateway Guide*. • A description of the `readWithCharset` function has been added to Functions(5) in the *Configuration Reference*. • The description of available access_token resolvers has moved from the `accessTokenResolvers` property of OAuth2ResourceServerFilter(5) in the *Configuration Reference* to the dedicated section Access Token Resolvers in the *Configuration Reference*. • The examples in "*Throttling the Rate of Requests to Protected Applications*" in the *Gateway Guide* have been changed to take the grouping policy and rate policy from fields in the OAuth2Context. • Documentation for the deprecated IG Route Monitoring Endpoint is removed in this release.

Appendix A. Release Levels and Interface Stability

This appendix includes ForgeRock definitions for product release levels and interface stability.

A.1. ForgeRock Product Release Levels

ForgeRock defines Major, Minor, Maintenance, and Patch product release levels. The release level is reflected in the version number. The release level tells you what sort of compatibility changes to expect.

Release Level Definitions

Release Label	Version Numbers	Characteristics
Major	Version: x[.0.0] (trailing 0s are optional)	• Bring major new features, minor features, and bug fixes • Can include changes even to Stable interfaces • Can remove previously Deprecated functionality, and in rare cases remove Evolving functionality that has not been explicitly Deprecated • Include changes present in previous Minor and Maintenance releases
Minor	Version: x.y[.0] (trailing 0s are optional)	• Bring minor features, and bug fixes

Release Label	Version Numbers	Characteristics
		• Can include backwards-compatible changes to Stable interfaces in the same Major release, and incompatible changes to Evolving interfaces • Can remove previously Deprecated functionality • Include changes present in previous Minor and Maintenance releases
Maintenance, Patch	Version: x.y.z[.p] The optional .p reflects a Patch version.	• Bring bug fixes • Are intended to be fully compatible with previous versions from the same Minor release

A.2. ForgeRock Product Interface Stability

ForgeRock products support many protocols, APIs, GUIs, and command-line interfaces. Some of these interfaces are standard and very stable. Others offer new functionality that is continuing to evolve.

ForgeRock acknowledges that you invest in these interfaces, and therefore must know when and how ForgeRock expects them to change. For that reason, ForgeRock defines interface stability labels and uses these definitions in ForgeRock products.

Interface Stability Definitions

Stability Label	Definition
Stable	This documented interface is expected to undergo backwards-compatible changes only for major releases. Changes may be announced at least one minor release before they take effect.
Evolving	This documented interface is continuing to evolve and so is expected to change, potentially in backwards-incompatible ways even in a minor release. Changes are documented at the time of product release. While new protocols and APIs are still in the process of standardization, they are Evolving. This applies for example to recent Internet-Draft implementations, and also to newly developed functionality.
Deprecated	This interface is deprecated and likely to be removed in a future release. For previously stable interfaces, the change was likely announced in a previous release. Deprecated interfaces will be removed from ForgeRock products.
Removed	This interface was deprecated in a previous release and has now been removed from the product.
Technology Preview	Technology previews provide access to new features that are evolving new technology that are not yet supported. Technology preview features may be functionally incomplete and the function as implemented is subject to

Stability Label	Definition
	change without notice. DO NOT DEPLOY A TECHNOLOGY PREVIEW INTO A PRODUCTION ENVIRONMENT. Customers are encouraged to test drive the technology preview features in a non-production environment and are welcome to make comments and suggestions about the features in the associated forums. ForgeRock does not guarantee that a technology preview feature will be present in future releases, the final complete version of the feature is liable to change between preview and the final version. Once a technology preview moves into the completed version, said feature will become part of the ForgeRock platform. Technology previews are provided on an “AS-IS” basis for evaluation purposes only and ForgeRock accepts no liability or obligations for the use thereof.
Internal/Undocumented	Internal and undocumented interfaces can change without notice. If you depend on one of these interfaces, contact ForgeRock support or email info@forgerock.com to discuss your needs.

Appendix B. Getting Support

This chapter includes information and resources for IG and ForgeRock support.

B.1. Accessing Documentation Online

ForgeRock publishes comprehensive documentation online:

- The ForgeRock Knowledge Base offers a large and increasing number of up-to-date, practical articles that help you deploy and manage ForgeRock software.

  While many articles are visible to community members, ForgeRock customers have access to much more, including advanced information for customers using ForgeRock software in a mission-critical capacity.

- ForgeRock product documentation, such as this document, aims to be technically accurate and complete with respect to the software documented. It is visible to everyone and covers all product features and examples of how to use them.

B.2. How to Report Problems or Provide Feedback

If you find issues or reproducible bugs, report them in https://bugster.forgerock.org.

When requesting help with a problem, include the following information:

- Description of the problem, including when the problem occurs and its impact on your operation
- Description of the environment, including the following information:

  - Machine type
  - Operating system and version
  - Web server or container and version
  - Java version
  - Patches or other software that might affect the problem
- Steps to reproduce the problem
- Relevant access and error logs, stack traces, and core dumps

B.3. Getting Support and Contacting ForgeRock

ForgeRock provides support services, professional services, training through ForgeRock University, and partner services to assist you in setting up and maintaining your deployments. For a general overview of these services, see https://www.forgerock.com.

ForgeRock has staff members around the globe who support our international customers and partners. For details on ForgeRock's support offering, including support plans and service level agreements (SLAs), visit https://www.forgerock.com/support.