Release Notes

ON THIS PAGE

Release Notes

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.

What's New

Discover new features.

Requirements

Check IG prerequisites.

Compatibility

Review incompatible changes.

Fixes

Review bug fixes, limitations, known issues.

Documentation

Track documentation changes.

Support

Get support and training.

ForgeRock® Identity Platform serves as the basis for our simple and comprehensive Identity and Access Management solution. We help our customers deepen their relationships with their customers, and improve the productivity and connectivity of their employees and partners. For more information about ForgeRock and about the platform, see https://www.forgerock.com.

The ForgeRock® Common REST API works across the platform to provide common ways to access web resources and collections of resources.

What’s New

What’s New in IG 7.1.2

▼ Support for SameSite Cookies in Standalone Mode

`sameSite` is a new subproperty of `session` in admin.json, to manage the circumstances in which a cookie is sent to the server. Use this property to reduce the risk of cross-site request forgery (CSRF) attacks when IG is in standalone mode.

▼ Functions `find` and `matchesWithRegex`

The functions `find` and `matchesWithRegex` are added to use as replacements for the deprecated function `matches`.

For more information, see Functions.

▼ Function `findGroups`

The function `findGroups` is added to use as a replacement for the deprecated function `matchingGroups`.

For more information, see Functions.

▼ Improved logging

Exception logging when looking for client certificates in ChfApplicationWebHandler has been improved.

When IG detects that `AMCtxId` is not available in a session, it now checks that notifications are enabled before logging an error. When notifications are disabled, there is no need to make `AMCtxId` available.

What’s New in IG 7.1.1

▼ Vert.x Metrics

Vert.x metrics are now available by default for IG in standalone mode, to provide metrics for HTTP, TCP, and the internal component pool. The metrics provide low-level information about requests and responses, such as the number of bytes, duration, the number of concurrent requests, and so on.

Metrics are provided at the Prometheus Scrape Endpoint and Common REST Monitoring Endpoint endpoints.

For more information, see the `vertx` object in AdminHttpApplication (`admin.json`), and Monitoring VertX Metrics.

▼ Additional Logging for a BadRequestException During Policy Evaluation Requests

To help with troubleshooting, a debug message is logged when a BadRequestException occurs during policy evaluation requests. In previous releases, the original error was not logged, IG just returned an HTTP 401 Unauthorized.

What’s New in IG 7.1

Non-Blocking Processing and Data Streaming

▼ Bi-directional Asynchronous Streaming of the HTTP Entity (HTTP/1.1 and HTTP/2)

streamingEnabled is a new property in `admin.json` for standalone mode to stream the content of HTTP requests and responses. When this property is `true`, the evaluation of runtime expressions that consume streamed content must be deferred.

This feature introduces changes that can impact your migration from a previous version of IG. For more information, see Incompatible Changes.

For more information, see AdminHttpApplication (`admin.json`) and runtime expression.

▼ Deferred Evaluation of Runtime Expressions

The evaluation of runtime expressions can be deferred until all of the content of the request or response is available. To prevent blocked threads, use deferred evaluation for runtime expressions that consume streamed content.

For more information, see runtime expression.

API Security

- ▼ Retention of URI Fragments During Authentication

  FragmentFilter is a new filter that enables URI fragments to be retained during authentication with the SingleSignOnFilter, CrossDomainSingleSignOnFilter, OAuth2ClientFilter, and PolicyEnforcementFilter. Previously, when an unauthenticated requested a resource that contained a URI fragment, the fragment was lost in the eventual redirect.

  For more information, see FragmentFilter.

- ▼ Customized Claim Checks in IdTokenValidationFilter

  Some OAuth 2.0 providers allow roles, groups, and custom properties to be defined in a JWT. The `customizer` property, previously available in the JwtValidationFilter, has been added to the IdTokenValidationFilter. Use this property to validate customized properties for a JWT, while still validating the existing constraints in the IdTokenValidationFilter.

  For more information, see IdTokenValidationFilter.

- ▼ JwtValidationFilter Applies Constraints for Claim Comparison and Pattern Match

  In JwtValidationFilter, the set of validation constraints for JWT claims and sub-claims now includes the following additional constraints:

  - **Claims comparisons** to check that a claim value compares to another value or the value of another claim as follows: `isGreaterOrEqualTo`, `isGreaterThan`, `isLessOrEqualTo`, or `isLessThan`.
  - **Regex** match to check that the claim value matches a specified regular expression.

  For more information, see the `customizer` property in JwtValidationFilter.

Secrets

- ▼ Support for PEM-Encoded Secrets

  PemPropertyFormat is a new format for secrets used in mappings in FileSystemSecretStore and SystemAndEnvSecretStore. Use PemPropertyFormat to read a Privacy-Enhanced Mail (PEM) file.

  For more information, see PemPropertyFormat. For examples, see Pass Runtime Data in a JWT Signed With a PEM and Pass Runtime Data in a JWT Signed and

Encrypted With a PEM.

▼ Support for SAML 2.0 Signing and Encryption With Secrets

IG can now use the Commons Secrets Service when acting as a SAML 2.0 service provider, when signing and/or encryption is enabled in the IDP or SP configuration in AM.

For more information, see SamlFederationHandler.

▼ Expose Cryptographic Keys as a JWK Set

JwkSetHandler is a new handler that exposes cryptographic keys as JWK set. Use this handler so that a downstream application can reuse the exposed keys for their assigned purpose.

For more information and an example of use, see JwkSetHandler.

▼ Support for Lease Expiry in Secret Stores

leaseExpiry is a new property for the following SecretStores, to define the time that secrets can be cached before they must be refreshed:

- SystemAndEnvSecretStore
- FileSystemSecretStore
- KeystoreSecretStore
- HsmSecretStore
- JwkSetSecretStore

For more information, see Secrets.

▼ Key ID Header Available for JwtBuilderFilter and JwtSession

The key ID header, `kid`, used to match a specific key, is now present in JWTs built by JwtBuilderFilter and JwtSession.

For information about `kid`, see "kid" (Key ID) Parameter.

Stability

▼ AmService Automatically Obtains SSO Token Header Name From AM

To reduce configuration errors, and simplify configuration, AmService no longer uses the default value, `iPlanetDirectoryPro`, for `ssoTokenHeader`. If `ssoTokenHeader` is not provided, IG queries the AM `/serverinfo/*` endpoint for the header name or cookie name of the SSO token.

▼ Filter to Rebase Requests Scheme, Host Name, and Port

The ForwardedRequestFilter has been added to rebase a request URI with a computed scheme, host name, and port. Use this filter to configure redirects when the request is forwarded by an upstream application such as a TLS offloader.

For more information, see ForwardedRequestFilter.

- ▼ Limit on Connection Attempts Prevents Stalled Requests and Timeouts

  initialConnectionAttempts is a new property in AmService to limit the number of times IG attempts to open a WebSocket connection before failing to deploy the route. Use this feature to prevent stalled requests and timeouts. For more information, see AmService.

Monitoring

- ▼ TimerDecorator Available for AccessTokenResolvers.

  The TimerDecorator can now record the time to process requests and responses as they pass through AccessTokenResolvers.

  For more information, see TimerDecorator.

- ▼ Log for Tested and Succesful Route Conditions.

  A new logger is available to log the routes for which IG evaluates a condition, and the route that matches a condition and treats a request.

  For more information, see the `condition` property of Route.

Other

- ▼ SAML 2.0 Requests Processed With Original URI Value

  useOriginalUri is a new property in SamlFederationHandler to prevent errors that occur when a `baseUri` decorator applies to the whole route. This change forces the handler to use the original URI instead of the rebased URI when validating RelayState and Assertion Consumer Location URLs.

  For more information, see SamlFederationHandler.

- ▼ New Methods to Get and Set URL-Encoded Form Data in Scripts

  Entity.getForm() and **Entity.setForm(Form)** are new methods available for use in scripts, with the content type `application/x-www-form-urlencoded`.

- ▼ Limit on Size to Which a JWT Can be Decompressed

  org.forgerock.json.jose.jwe.compression.max.decompressed.size.bytes is a new system property to limit the maximum size to which a compressed JWT can be decompressed. This property reduces the risk of a decompressed JWT consuming too much available memory.

  For more information, see Provided Properties.

- ▼ Temporary Storage Directory

By default, the TemporaryStorage object now stores temporary files in $HOME/.openig/tmp instead of a directory defined by the system property `java.io.tmpdir`.

For more information, see TemporaryStorage.

▼ Redirection Marker Can Be Disabled or Renamed

redirectionMarker is a new property in SingleSignOnFilter to limit the number of authentication redirects.

When there is no SSO session due to, for example, SSO cookie name misconfiguration, an authentication request fails and is redirected back to IG. The scenario can result in infinite authentication redirects.

For more information, see SingleSignOnFilter.

▼ Log Entry for Number of Retries

When a runtime error occurs during the execution of a request to a remote server, IG retries the request until the allowed number of retries is reached or the execution succeeds. The retries are now logged by default.

For more information, see the `retries` property of ClientHandler.

▼ System Property to Decode Invalid Characters Without Error

org.forgerock.http.util.ignoreFormParamDecodingError is a new Java system property to ignore form encoding errors caused by invalid characters. Encoded values are used instead.

For more information, see Supported System Properties.

Requirements

IMPORTANT

ForgeRock supports customers using the versions specified here. Other versions and alternative environments might work as well. When opening a support ticket for an issue, however, make sure you can also reproduce the problem on a combination covered here.

Downloads

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

- `IG-7.1.2.zip` : For deployment in standalone mode
- `IG-7.1.2.war` : For deployment in web container mode

- `IG-sample-application-7.1.2.jar` : Web application for testing IG configurations

For information about using the Docker image provided with the product software, see the Deployment Guide.

Operating Systems

IG is tested on Windows and Linux operating systems.

Web Application Containers

In web container mode, IG runs in the following containers:

- Apache Tomcat 9
- Jetty 9
- JBoss EAP 7.3

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

Java

ForgeRock recommends that you keep your Java installation up-to-date with the latest security fixes. IG supports the following Java environments:

Vendor	Version
OpenJDK, including OpenJDK-based distributions: • AdoptOpenJDK/Eclipse Adoptium • Amazon Corretto • Azul Zulu • Red Hat OpenJDK ForgeRock tests most extensively with AdoptOpenJDK/Eclipse Adoptium. ForgeRock recommends using the HotSpot JVM.	11
Oracle Java	11

HTTP Protocol

IG supports HTTP/1.1 and HTTP/2.0.

HTTP/1.0 is not supported.

FQDNs

IG replication requires use of fully qualified domain names (FQDNs), such as `openig.example.com`.

Hostnames like `example.com` are acceptable for evaluation. In production, and when using replication across systems, you must either ensure DNS is set up correctly to provide FQDNs, or update the hosts file (`/etc/hosts` or `C:\Windows\System32\drivers\etc\hosts`) to supply unique, FQDNs.

Certificates

For secure network communications with client applications that you do not control, install a properly signed digital certificate that your client applications recognize, such as one that works with your organization's PKI, or one signed by a recognized CA.

To use the certificate during installation, the certificate must be located in a file-based keystore supported by the JVM (JKS, JCEKS, PKCS#12), or on a PKCS#11 token. To import a signed certificate into the server keystore, use the Java **`keytool`** command.

Third-Party Software for Encryption

Bouncy Castle is required for signature encryption with RSASSA-PSS or Deterministic ECDSA. For information, see The Legion of the Bouncy Castle.

Third-Party Software

ForgeRock provides support for using the following third-party software when logging ForgeRock Common Audit events:

Software	Version
Java Message Service (JMS)	2.0 API
MySQL JDBC Driver Connector/J	8 (at least 8.0.19)
Splunk	8.0 (at least 8.0.2)

TIP

TIP

Elasticsearch and Splunk have native or third-party tools to collect, transform, and route logs. Examples include Logstash and Fluentd.

ForgeRock recommends that you consider these alternatives. These tools have advanced, specialized features focused on getting log data into the target system. They decouple the solution from the ForgeRock Identity Platform systems and version, and provide inherent persistence and reliability. You can configure the tools to avoid losing audit messages if a ForgeRock Identity Platform service goes offline, or delivery issues occur.

These tools can work with ForgeRock Common Audit logging:

- Configure the server to log messages to standard output, and route from there.
- Configure the server to log to files, and use log collection and routing for the log files.

ForgeRock provides support for using the following third-party software when monitoring ForgeRock servers:

Software	Version
Grafana	5 (at least 5.0.2)
Graphite	1
Prometheus	2.0

For hardware security module (HSM) support, ForgeRock software requires a client library that conforms to the PKCS#11 standard v2.20 or later.

Studio Browser

ForgeRock has tested many browsers with Studio, including:

- Chrome, latest stable version
- Firefox, latest stable version

Features Using ForgeRock Access Management

Feature	Supported in AM Version
Support for refresh of idle sessions when the SingleSignOnFilter is used for authentication with AM. For more information, see the `sessionIdleRefresh` property of AmService.	AM 6.5.3 and later versions.
Eviction of revoked OAuth 2.0 access_tokens from the cache. For more information, see CacheAccessTokenResolver, and the `cache` property of OAuth2ResourceServerFilter.	AM 6.5.3 and later versions.
Support for OAuth 2.0 Mutual TLS (mTLS). For more information, see ConfirmationKeyVerifierAccessTokenResolver, and Validate Certificate-Bound Access Tokens.	AM 6.5.1 and later versions.
Eviction of entries from the AmService `sessionCache`, using WebSocket notifications from AM. For more information, see AmService.	AM 5.5 when the user manually safelists the `AMCtxId` session property, and with AM 6 and later versions.
AM password capture and replay, as described in Get Login Credentials From AM.	AM 5 and later versions, and AM 6 and later versions when the `AES` keyType is used to decrypt the password.
AM policy enforcement, as described in Enforce Policy Decisions From AM.	AM 5 and later versions
OpenID Connect dynamic registration and discovery, as described in Discover and Dynamically Register With OpenID Connect Providers.	OpenAM 13.5, and AM 5 and later versions
Token transformation, as described in Transform OpenID Connect ID Tokens Into SAML Assertions.	OpenAM 13.5, and AM 5 and later versions
User Managed Access 2.x, for IG 5.5, as described in Support UMA Resource Servers.	AM 5.5 and later versions

Feature	Supported in AM Version
User Managed Access 1.x, for IG 5 and earlier versions.	AM 5.1 and earlier versions
Single sign-on, as described in Single Sign-On and Cross-Domain Single Sign-On.	AM 5 and later versions
Cross-domain single sign-on, as described in Authenticate With CDSSO.	AM 5.5 and later versions
Capture and storage of AM session information, as described in SessionInfoFilter.	AM 6 and later versions
Capture and storage of AM user profile attributes, as described in UserProfileFilter.	AM 5 and later
Support for transactional authorization, as described in Harden Authorization With Advice From AM.	AM 5.5 and later versions
Validation of stateless access_tokens, as described in Validate Stateless Access_Tokens With the StatelessAccessTokenResolver.	OpenAM 13.5, and AM 5 and later versions
Retrieval of specified session properties or all session properties from AM, without relying on AM’s Session Properties Whitelist. Described in AmService.	AM 5.1.2 and later versions

Incompatible Changes

Incompatible Changes in IG 7.1.2

The following change introduced in this release can impact your migration from IG 7.1.1:

▼ Logback Upgrade

IG has upgraded the version of Logback, used for the logging framework. The Logback update introduces changes that can affect your existing deployment. For more information about changes in Logback, see the Logback website.

Incompatible Changes in IG 7.1.1

The following change introduced in this release can impact your migration from IG 7.1:

▼ Proxying WebSocket Traffic in Standalone Mode

When IG is in standalone mode, proxying Websocket traffic can produce errors where requested subprotocols not supported. To prevent these error, you must now list the subprotocols that are proxied by IG in the `vertx` property of admin.json.

Incompatible Changes in IG 7.1

The following changes introduced in this release can impact your migration from IG 7.0:

▼ Name of TimerDecorator in Prometheus Output

In the Prometheus output, information for the default TimerDecorator is always included as `name="gateway.timer"`.

In previous releases, information is included in the Prometheus output as follows:

- When a default TimerDecorator **is not** declared in `config.json`, information is included as `name="timer"`.
- When a default TimerDecorator **is** declared in `config.json`, information is included as `name="gateway.timer"`.

For more information, see TimerDecorator.

▼ Runtime expressions that consume streamed content written with a #

To prevent IG from blocking executing threads, write runtime expressions that consume streamed content with `#` instead of `$`. This ensures that IG does a deferred evaluation.

For IG in standalone mode, when the new `streamingEnabled` property in `admin.json` is `true`, expressions that consume streamed content **must** be written with `#` instead of `$`.

For more information, see runtime expression.

▼ New methods for asynchronous execution of scripts

NOTE

This change is required if the new `streaming` property in `admin.json` is `true`. Otherwise, the change is recommended but not required.

APIs that read the entity content have been updated to execute scripts asynchronously.

▼ Username of an AM Identity is Now subname

Before AM 7.1, the `sub` claim of OAuth 2.0 access_tokens and id_tokens contained only the username. From AM 7.1, the username is contained in the `subname` claim. The `sub` claim includes additional information.

Update scripts and expressions in IG that use the `sub` claim.

▼ Secrets From Secret Stores Expire by Default

Secrets from FileSystemSecretStore, HsmSecretStore, KeyStoreSecretStore, and SystemAndEnvSecretStore, now expire after a default of five minutes, or after the time specified in the property `leaseExpiry`. In the previous release, secrets from these secret stores never expired or had other expiry times.

▼ Entity.toString() Function Does Not Return Content

The `Entity.toString()` function no longer returns the entity content as a string. Instead, it returns only metadata. This change prevents buffering of the entity content during logging, which, when the entity is big, can impede asynchronous operation.

To return the entity content as a string, replace `request.entity.toString()` and `response.entity.toString()` functions with `request.entity.string` and `response.entity.string`.

▼ Capture and Logging of Entity

To faciltate asynchronous processing in this release, when the CaptureDecorator property `captureEntity` is `false`, the decorator does not capture the message entity, and writes nothing to the logs.

In previous releases, when `captureEntity` was `false`, the decorator wrote `[entity]` in the log to show that there was an entity but that capture was not configured.

▼ RSA Keys MUST be at Least 2048 Bits

For security, RFC 7518 - Digital Signature with RSASSA-PKCS1-v1_5 requires that RSA keys must be 2048 bits or larger. Smaller keys are now rejected.

▼ 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.

To redirect a request to a site that does not meet the goto URL criteria, change the original URI by using a ForwardedRequestFilter.

For more information, see OAuth2ClientFilter and ForwardedRequestFilter.

Deprecation

WARNING

IG logs a warning message each time it evaluates a call to a deprecated function. Under high loads, logging high numbers of messages can reduce performance. Consider the impact on performance if you decide to continue to use deprecated functions in your deployment.

Deprecation is defined in ForgeRock Product Stability Labels.

Deprecated Functionality in IG 7.1.2

The following additional properties are deprecated in this release:

Object	Deprecated Settings	Replacement Settings
Functions	`matches`	Replaced by `matchesWithRegex` or `find`.
	`matchingGroups`	Replaced by `findGroups`.

Deprecated Functionality in IG 7.1.1

No additional functionality was deprecated in this release.

Deprecated Functionality in IG 7.1

The following features and properties are deprecated:

▼ Delivery of IG war file

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

▼ Methods to read or set query and form parameters

The `request.form` method used in scripts to read or set query and form parameters is deprecated. Use the following methods instead:

- `Request.getQueryParams()` to read query parameters.
- `Entity.getForm()` to read form parameters.
- `Entity.setForm()` to set form parameters.

▼ LdapClient class and 'ldap' script binding

The LdapClient class and the `ldap` script binding are deprecated.

Object	Deprecated Settings	Replacement Settings
AmService	`password`	Replaced by `passwordSecretId`. If the deprecated and replacement properties are both provided, the replacement property takes precedence.
AuditService	`event-handlers`	Replaced by `eventHandlers`.
CapturedUserPasswordFilter	`key`	Replaced by `keySecretId`. If the deprecated and replacement properties are both provided, the replacement property takes precedence.

Object	Deprecated Settings	Replacement Settings
ClientHandler	`proxy` subproperty `password`	Replaced by `passwordSecretId`. If the deprecated and replacement properties are both provided, the replacement property takes precedence.
	- `keyManager` - `sslCipherSuites` - `sslContextAlgorithm` - `sslEnabledProtocols` - `trustManager`	Replaced by the ClientTlsOptions object. For more information, see ClientTlsOptions.
	`websocket` subproperties: - `keyManager` - `sslCipherSuites` - `sslContextAlgorithm` - `sslEnabledProtocols` - `trustManager`	Replaced by the ClientTlsOptions object. For more information, see ClientTlsOptions.

Object	Deprecated Settings	Replacement Settings
ClientRegistration	- `keystore` - `privateKeyJwtAlias` - `privateKeyJwtPassword`	Replaced by `privateKeyJwtSecretId`. If the deprecated and replacement properties are both provided, the replacement property takes precedence.
	`name`, when used to identify a `registration`	Replaced by `clientId`. For information, see ClientRegistration, and the example route in Use Multiple OpenID Connect Providers.
	`clientSecret`	Replaced by `clientSecretId`. If the deprecated and replacement properties are both provided, the replacement property takes precedence.
CorsFilter	`origins`	Replaced by `acceptedOrigins`. For information, see CorsFilter.
CryptoHeaderFilter	Whole object	Not replaced. For information, see CryptoHeaderFilter.
DesKeyGenHandler	Whole object	Not replaced. For information, see DesKeyGenHandler.
ElasticsearchAuditEventHandler	Whole object	Not replaced.

Object	Deprecated Settings	Replacement Settings
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.

Object	Deprecated Settings	Replacement Settings
JwtSession	`encryptionSecretId` and `signatureSecretId`	Replaced by `authenticatedEncryptionSecretId` and `encryptionMethod` .
	`cookieName` and `cookieDomain`	Replaced by `cookie` , and its subproperties `name` , `domain` , `path` , `secure` , `httpOnly` . If the deprecated and replacement properties are both provided, the replacement property takes precedence.
	`password`	Replaced by `passwordSecretId` . If the deprecated and replacement properties are both provided, the replacement property takes precedence.
	Combination of `password` , `alias` , and `keystore` Combination of `passwordSecretId` , `alias` , and `keystore`	Replaced by `encryptionSecretId` . 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.

Object	Deprecated Settings	Replacement Settings
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.
OpenAmAccessTokenResolver	Whole object	Not replaced. For information, see OpenAmAccessTokenResolver.
ReverseProxyHandler	• `keyManager` • `sslCipherSuites` • `sslContextAlgorithm` • `sslEnabledProtocols` • `trustManager`	Replaced by the ClientTlsOptions object. For more information, see ClientTlsOptions.
	`websocket` subproperties: • `keyManager` • `sslCipherSuites` • `sslContextAlgorithm` • `sslEnabledProtocols` • `trustManager`	Replaced by the ClientTlsOptions object. For more information, see ClientTlsOptions.

Object	Deprecated Settings	Replacement Settings
Route	`monitor`	Replaced by the Prometheus Scrape Endpoint and Common REST Monitoring Endpoint. For information, see Monitoring Endpoints.
SingleSignOnFilter	`logoutEndpoint`	Replaced by `logoutExpression`.
SplunkAuditEventHandler	Whole object	Not replaced.
SqlAttributesFilter	`dataSource` as a JNDI lookup name	Replaced by `dataSource` as a `JdbcDataSource` configuration object.
StatelessAccessTokenResolver	`signatureSecretId`	Replaced by `verificationSecretId`.
	`encryptionSecretId`	Replaced by `decryptionSecretId`.
UserProfileFilter	`ssoToken`	Replaced by `username in` UserProfileFilter.
	`amService and profileAttributes`	Replaced `amService and profileAttributes`, as sub-properties of `userProfileService`

Object	Deprecated Settings	Replacement Settings
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.

Removed

Removed is defined in ForgeRock Product Stability Labels.

Removed Functionality in IG 7.1.2

No functionality was removed in this release.

Removed Functionality in IG 7.1.1

No functionality was removed in this release.

Removed Functionality in IG 7.1

The following feature was removed in this release.

▼ IG route monitoring endpoint

The IG Route Monitoring Endpoint is removed. As a replacement, IG provides Prometheus Scrape Endpoint and Common REST Monitoring Endpoint.

For more information, see Monitoring at the Prometheus Scrape Endpoint, and Monitoring the Common REST Monitoring Endpoint,

Fixes

For information about security issues fixed in this release, see Security Advisories.

Fixes in IG 7.1.2

- OPENIG-6394: Stack traces are printed twice in the log files
- OPENIG-6206: When checking for peer certificates in a request, validate that the SSLSession is available
- OPENIG-5872: Stop Tyrus WebSocket connection retry when Websocket Client is closed
- OPENIG-5793: Unexpected behaviour of EL function matches

Fixes in IG 7.1.1

- OPENIG-4956: Inbound WebSocket connection is not closed when outbound connection is closed abruptly
- OPENIG-5539: The ForwardedRequestFilter should not change original URI parameter values when rebasing
- OPENIG-5540: PEM secret format fails to decode some EC private keys
- OPENIG-5610: Null Pointer Exception when using ForwardedRequestFilter with ResourceHandler
- OPENIG-5683: HTTP/2 : set max connections
- OPENIG-5743: Standalone: Possible OOME for large requests
- OPENIG-5778: sessionInfo requests can lead to a build up of agent tokens being created
- OPENIG-5805: The notification service should attempt to refresh the caller token when receiving a 401 on WebSocket connections
- OPENIG-5868: WebSocketClientHandshakeException: Invalid subprotocol seen when using IG standalone to proxy WebSocket requests

Fixes in IG 7.1

- OPENIG-4034: AuditService does not delete old files when maxDiskSpaceToUse is reached
- OPENIG-4900: AMService cannot connect to AM via TLS with Standalone
- OPENIG-5084: WebSocket connections are not being proxied when baseURI scheme is wss

- OPENIG-5219: Vert.x HTTP Client does not replicate current CHF behaviour when request fails and headers have been received
- OPENIG-5258: IG Standalone must populate the originalUri.port from Host header
- OPENIG-5401: Retries on a ReverseProxyHandler not being triggered

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*.

Limitations

Limitations are inherent to the design, not bugs to be fixed:

▼ Pre-exisiting fragment cookie overwrites the current fragment cookie during authentication

OPENIG-6288

When a user has a pre-exisiting fragment cookie during authentication, for example, from a previous, incomplete authentication attempt, the pre-exisiting fragment overwrites the current fragment.

To minimize the impact of this limitation, the FragmentFilter cookie has a new property `maxAge` to configure the maximum duration for which it can remain valid.

▼ Multiple spaces in unquoted cookie values are changed to a single space in JBoss

OPENIG-4395

In JBoss, multiple spaces in unquoted cookie values are reduced to one space. For example:

```
testCookieName=cookie    value
```

is changed to

```
testCookieName=cookie value
```

▼ No access to common time related functions in expressions

OPENIG-4201

The value of `System.currentTimeMillis()` cannot currently be used in filters, such as JwtBuilderFilter, for claims such as `exp` and `iat`.

▼ 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.

▼ Persist 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 Support UMA Resource Servers.

▼ Proxy WebSocket traffic when running in Jetty

OPENIG-3248

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

For more information, see Proxy WebSocket Traffic, and the `websocket` property of ClientHandler or ReverseProxyHandler.

▼ 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.

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.

Known Issues

IG issues are tracked at https://bugster.forgerock.org/jira/browse/OPENIG.

Known Issues in IG 7.1.2

No additional issues were introduced in this release.

Known Issues in IG 7.1.1

No additional issues were introduced in this release.

Known Issues in IG 7.1

- OPENIG-5913: Route configuration lost sometime after un-deploy from route list
- OPENIG-5872: Incorrect URL for Groovy Inlined Scripts
- OPENIG-5725: Add SNI configuration
- OPENIG-5425: JwkSetHandler: No error displayed when using an invalid configuration such as a public key exported -as jwk- for decryption usage
- OPENIG-4817: Host information not forwarded for HTTP/2 requests

Documentation

Date	Description
July 2022	A section on upgrade has been added to the Installation Guide, and information about migrating from web container mode to standalone mode has been taken from the Release Notes.
April 2022	Update to include information and examples for deprecated properties `matches` and `matchingGroups`.
	Initial release of IG 7.1.2 software.
September 2021	Initial release of IG 7.1.1 software.

Date	Description
May 2021	Initial release of IG 7.1 software. In addition to the changes described elsewhere in these notes, the following important changes were made to the documentation: ***New documents*** • Installation Guide • Security Guide ***Best practices*** • Information about how to keep log files clean and readable, and to prevent log flow attacks has been added to Limit Repetitive Log Messages. ***Reorganization*** • Examples using a JwtBuilderFilter and HeaderFilter to pass identity and other runtime information downstream have been moved from JwtBuilderFilter to Pass Identity and Other Runtime Data Downstream. • Information has about the `session` property has been added to AdminHttpApplication (admin.json). • The following objects have been moved from Miscellaneous Configuration Objects to Secrets: PemPropertyFormat, SecretKeyPropertyFormat, SecretsKeyManager, SecretsProvider, and SecretsTrustManager. ***New example*** • An example of token revocation has been added to Cache Access Tokens. • Examples have been added to UriPathRewriteFilter.

Appendix A: Release Levels and Interface Stability

ForgeRock Product Release Levels

ForgeRock defines Major, Minor, Maintenance, and Patch product release levels. The version number reflects release level. 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. • 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.

ForgeRock Product Stability Labels

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

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

ForgeRock Stability Label Definitions

Stability Label	Definition
Stable	This documented feature or 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 feature or 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.
Legacy	This feature or interface has been replaced with an improved version, and is no longer receiving development effort from ForgeRock. You should migrate to the newer version, however the existing functionality will remain. Legacy features or interfaces will be marked as *Deprecated* if they are scheduled to be removed from the product.
Deprecated	This feature or interface is deprecated, and likely to be removed in a future release. For previously stable features or interfaces, the change was likely announced in a previous release. Deprecated features or interfaces will be removed from ForgeRock products.
Removed	This feature or interface was deprecated in a previous release, and has now been removed from the product.

Stability Label	Definition
Technology Preview	Technology previews provide access to new features that are considered as new technology that is not yet supported. Technology preview features may be functionally incomplete, and the function as implemented is subject to 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 features or interfaces can change without notice. If you depend on one of these features or interfaces, contact ForgeRock support or email info@forgerock.com to discuss your needs.

Getting Support

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.

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 everyone, 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.