

# Release 17.0 Upgrade Instructions

Before you upgrade make sure to read the [General Upgrade instructions](#) as well as the Upgrade instructions for previous versions.



The changes are color coded. **Orange** elements have been changed, **Green** elements have been added and **Red** elements have been removed compared to the 16.x release.

## Table of contents

- [Blueriq SDK](#)
  - [REST request parsing](#)
- [OAuth2 Rest Service client properties](#)
- [OpenAPI specification version](#)
- [Legacy properties](#)
- [Security-enabled property](#)
- [Known issues](#)

## Blueriq SDK

### REST request parsing

The method

```
ICommunicationEngine#parseRestRequest(IRestRequestMessage request, InferenceContext target, String serviceName, ICommunicationEngineHandler handler, IDebugger debugger)
```

has been updated to

```
ICommunicationEngine#parseRestRequest(IRestRequestMessage message, InferenceContext target, String serviceName, String operationName, ICommunicationEngineHandler handler, IDebugger debugger)
```

i.e. the `operationName` now needs to be provided up-front. Previously it was computed internally, but this resulted in duplicating the work as the caller also needs to know the operation. The corresponding operation for the REST message can be determined using `ICommunicationEngine#matchOperation(IRestRequestMessage request, String serviceName, ICommunicationEngineHandler handler)`, of which the resulting `IRestOperationDefinition#getName()` can be passed as `operationName`.

#### Before

```
engine.parseRestRequest(request, target, serviceName, handler, debugger);
```

#### After

```
IRestOperationDefinition operation = engine.matchOperation(request, serviceName, handler);  
engine.parseRestRequest(request, target, serviceName, operation.getName(), handler, debugger);
```

## OAuth2 Rest Service client properties

We changed the way you configure an OAuth2 client on REST connections. Now you need to define OAuth2 clients separately and reference them from multiple connections.

You need to migrate the connection properties if you are using OAuth2 for rest service clients.

In the old situation you would need to do something like this:

## Before

```
blueriq:
  connection:
    my-connection:
      http:
        url: https://some.domain.com/resource
        authentication: oauth2
        oauth2-client-id: my-client-id
        oauth2-client-secret: secret-password-text
        oauth2-grant-type: client_credentials
        oauth2-token-endpoint: https://identity.provider.com/token
```

In the new situation you can define OAuth2 clients separately and reference them from multiple connection if you want:

## After

```
spring:
  security:
    oauth2:
      client:
        registration:
          my-oauth2-client:
            provider: my-auth-server
            client-id: my-client-id
            client-authentication-method: client_secret_basic
            client-secret: secret-password-text
            authorization-grant-type: client_credentials
        provider:
          my-auth-server:
            token-uri: https://identity.provider.com/token
blueriq:
  connection:
    my-connection1:
      http:
        url: https://some.domain.com/resource1
        authentication: oauth2
        oauth2-client-registration: my-oauth2-client
    my-connection2:
      http:
        url: https://some.domain.com/resource2
        authentication: oauth2
        oauth2-client-registration: my-oauth2-client
```

## OpenAPI specification version

Blueriq 17 introduces support for OpenAPI v3.1, in addition to v3.0. The default version has been changed to 3.1, so if v3.0 of the specification is desired then `spec=3.0` needs to be included when requesting the OpenAPI docs.

## Legacy properties

The following legacy properties have been removed:

- `blueriq.legacy.mapping-to-same-module`
- `blueriq.legacy.legacy-jwt-claims-path-expression`
- `blueriq.legacy.legacy-rest-validation`

## Security-enabled property

The `securityEnabled` property of flow and function shortcuts has been removed.

Authentication and authorization now have to be achieved by modelling roles on exposed flows and exposed decision trees. From now on at least one role is required to be set to enable authentication for the following:

- Blueriq as a Services (BAAS)
- Blueriq as a Rest Services (BAARS)
- Exposed function flows
- Decision trees
  - Roles can now be modelled for exposed decision trees.

It is recommended for the `securityEnabled` property to be removed from all configuration; each remaining `securityEnabled` property will be reported as a WARN in the log.

The BAAS response when authentication or authorization is insufficient or missing has changed from status code 401 or 403 with a JSON response to a status code 500 with a SOAP fault as response body, to better match SOAP semantics.

## Known issues

For an overview of known issue please refer to: [Known issues](#).