/ open banking

Forgerock OBIE Production Assets

Release Notes v.1.1

Table of Contents

Table of Contents 2

1 Chapter 1. What's New 3

1.1 New Features 3

2 Chapter 2. Fixes, Limitations, and Known Issues 4

2.1 Fixed Issues 4

2.2 Limitations 5

2.3 Known Issues 5

|  |  |  |  |
| --- | --- | --- | --- |
| / **Author** | / Action | / Date | / Version |
| Daniel Coman  Marian Tiris | First version | 2019-06-07 | 1.0 |

Disclaimer of Liability

While every effort will be made to ensure that the information contained within the document is accurate and up to date, ForgeRock makes no warranty, representation or undertaking whether expressed or implied, nor does it assume any legal liability, whether direct or indirect, or responsibility for the accuracy, completeness, or usefulness of any information.

Further analysis on the detailed requirements, fine-tuning and validation of the proposed architecture is still required by the selected Systems Integrator and/or Architects in charge of architecting the IAM project.

Copyright Unless otherwise stated, copyright of all material published within this website is reserved by ForgeRock. All rights are reserved. As permitted under the provisions of the Copyright Act 1968, no part may be reproduced or re-used for any purposes whatsoever without the prior written permission of ForgeRock.

# Chapter 1. What's New

This chapter covers the new features and improvements done in the current release of ForgeRock Open Banking Accelerators.

## New Features

* **Authorize flow:** Added a new IG filter for the /authorize endpoint + a new groovy script for properly manipulating the errors (there are cases when AM doesn’t include as a fragment the error details, but through query parameters which doesn’t comply with the conformance suite)
* **IG Error Handling:** Updated the IG routes for handling the error response codes and payloads according to the Open Banking specifications
* **AM Policies:** Updated the creation of the AM authorization policies (from the custom IDM scripts) in order to include also the TPP identifier as a claim
* **IDM Managed Objects:** Updated the TPP managed object to include now as the identifier the Oauth client id generated by AM (it still includes the TPP identifier from the certificate)
* **AM Base URL Service:** Configured in AM, as the Base URL the hostname of the IG component
* **IG Custom Filters:** A new filter was added in order to check that the content of the request JWT parameter used as input for the authorization flow is compliant with the specification (mandatory information is included in the claims)

# Chapter 2. Fixes, Limitations, and Known Issues

This chapter covers the status of key issues and limitations at release 1.1.

## Fixed Issues

The following important bugs were fixed in this release:

* **IG TPP registration route:** Sample TPP registration route - should have dedicated secret for openidm-admin (not share agent secret).
  + Created new secret for the IDM user that is used in the registration route;
* **IG TPP registration route:** For registration, the IDM managed TPP object is created, then the dynamic client registration against AM so if the dynamic client registration fails, you end up with a zombie TPP object any plans to look at this
  + Changed the RegisterTppForwardFilter to be executed on response instead of on request, after the Oauth client is created in AM;
* **IG TPP registration custom filter**: TPP registration filter should have configurable client cert header
  + RegistrationJwtVerificationFilter and CertificateExtensionValidatorFilter now have the client cert header configurable in config field: "clientCertificateHeaderName" Eg: "clientCertificateHeaderName”: "ssl-client-cert";
* **IG TPP registration custom filter:** TPP registration filter expects client certificate to be URL encoded (not the case with haproxy)
  + Checking if the TPP certificate contains %. Normally certificates don't contain this. If header cert contains % will be decoded, otherwise proceed without decoding;
* **IG TPP registration custom filter:** RegisterTppForwardFilter NPE decoding openidm-admin password:
  + openIdmPassword is the name of the config field that holds the name of the context attribute holding the password. Changed the secret to be plain and now the code looks like this:

String password = (String) context.asContext(AttributesContext.class).getAttributes().get(openIdmPassword); requestToIDM.addHeader("X-OpenIDM-Password", password);

* **IG filters:** NPE when no client cert
  + Will block flow and return 401 Unauthorized if client cert is null;
* **RCS:** RCS backend changes keys at each restart
  + Configured in AM console to use for the forgercok-rcs agent the JWKS\_URI instead of plain keys value
* **RCS:** No check if the TPP identifier related to a specific consent resource is the same as the one trying to authorize it
  + A new check was implemented in RCS when retrieving the consent details from IDM in order to be sure that the TPP that is trying to authorize is the same with the one that have submitted the intent in the first phase;
* **IDM custom endpoints:** Policy creation script has hardcoded lbcookie
  + Removed the amlbcoockie from idm custom script;
* **IDM custom endpoints**: The response to account-access-consent is non conformant - specifically - POST response includes an empty "id" field, and additional "response" envelope
  + Updated IDM scripts in order to format the json response returned by IDM;
* **IDM custom endpoints:** GET includes an \_id and \_rev section, and returns no meta section,
  + Updated IDM scripts in order to format the json response returned by IDM;
* **IDM custom endpoints:** DELETE returns a 200 with the consent, but should return a 204 with no body
  + Updated IDM scripts in order to format the code and json response returned by IDM;
* **IDM custom endpoints**: TPP's can access account consents they don't own, including deletion
  + Implemented an extra check that the id of the TPP that is requesting the information is the same with the one stored in IDM for the AISP and PISP consent resource.

## Limitations

* There are no known limitations in Open Banking Accelerators 1.1

## Known Issues

* **AM PS256:** by default, in the forgeops openam image, the BouncyCastle java security provider isn’t installed and the PS256 algorithm (mandatory in the Open Banking specifications) is not working;
* **Tomcat Request Header too large:** by default, in the forgeops tomcat configuration of the images, the **maxHttpHeaderSize (**The maximum size of the request and response HTTP header, specified in bytes) is set to the default value 4096 (4 KB) and, for some browsers, this can cause problems on the authorization flow (error Request header is too large).