VERSION LOG OUT

User: amAdmin Server: sp.example.com

FORGEROCK

Save Reset Back

- General Configuration
- Deployment Configuration
- Issued SAML2 Token Configuration
- OpenIdConnect Token Configuration

General Configuration

Persist Issued Tokens in Core Token Store: ☑

Necessary to support token validation and cancellation

Supported Token Transforms:

UNT->SAML2;invalidate interim OpenAM session
UNT->SAML2;don't invalidate interim OpenAM session
OPENIDCONNECT->SAML2;invalidate interim OpenAM session
OPENIDCONNECT->SAML2;don't invalidate interim OpenAM session

Supported token transformations

Custom Token Validators (optional):

Current Values Remove

New Value Add

If validator of a custom token type is desired, specify the name of the custom token here, followed by '|', followed by the class name of the org.forgerock.openam.sts.rest.token.validator.RestTokenTransformValidator implementation which will be invoked to validate the custom tokens.

Custom Token Providers (optional):

Current Values Remove

New Value Add

If a rest-sts instance is to produce a custom token, specify the name of the custom token here, followed by '|', followed by the class name of the org.forgerock.openam.sts.rest.token.provider.RestTokenProvider implementation which will be invoked to produce an instance of the custom token.

Custom Token Transforms (optional):

Current Values Remove

New Value Add

If either custom token validators or providers are specified, they must also be specified in a custom rest-sts token transformation. These input or output tokens can be specified in a transformation with standard, or other custom, tokens.

Back to top

Deployment Configuration

Deployment Url Element: test-sts

STS endpoint Url will be composed of rest-sts/realm/urlElement

Authentication Target Mappings:

Current Values

USERNAME|service|ldapService
OPENIDCONNECT|module|oidc|oidc_id_token_auth_target_header_key=oidc_id_token
X509|module|cert_module|x509_token_auth_target_header_key=client_cert

Remove

New Value Add

Configuration of consumption of OpenAM's rest-authN

Client Certificate Header Key:

TLS-offload host certificate header key

Trusted Remote Hosts:

Current Values

Remove

New Value Add

IP addresses of TLS-offload hosts

Back to top

Issued SAML2 Token Configuration

The SAML2 issuer Id: sp.example.com

Service Provider Entity Id: http://sp.example.com:8

Values will be used to populate the Audiences of the AudienceRestriction element of the Conditions element. This value is required when issuing Bearer assertions. See section 4.1.4.2 of Profiles for the OASIS Security Assertion Markup Language (SAML) V2.0 for details.

Service Provider Assertion Consumer Service Url:

When issuing bearer assertions, the recipient attribute of the SubjectConfirmation element must be set to the Service Provider Assertion Consumer Service Url. See section 4.1.4.2 of Profiles for the OASIS Security Assertion Markup Language (SAML) V2.0 for details. Value required when issuing Bearer assertions.

NameIdFormat: urn:oasis:names:tc:SAML:1.0:nameid-format:unspecified

Token Lifetime (Seconds): 60000

Custom Conditions Provider Class Name (optional):

If the Conditions of the issued SAML2 assertion need to be customized, implement the `org.forgerock.openam.sts.tokengeneration.saml2.statements.ConditionsProvider` interface, and specify the class name of the implementation here.

Customs Subject Provider Class Name (optional):

If the Subject of the issued SAML2 assertion needs to be customized, implement the `org.forgerock.openam.sts.tokengeneration.saml2.statements.SubjectProvider` interface, and specify the class name of the implementation here.

Custom

AuthenticationStatements Class Name (optional):

If the AuthenticationStatements of the issued SAML2 assertion need to be customized, implement the `org.forgerock.openam.sts.tokengeneration.saml2.statements.AuthenticationStatementsProvider` interface, and specify the class name of the implementation here.

Custom AttributeStatements Class Name (optional):

If the AttributeStatements of the issued SAML2 assertion need to be customized, implement the `org.forgerock.openam.sts.tokengeneration.saml2.statements.AttributeStatementsProvider` interface, and specify the class name of the implementation here.

Custom Authorization Decision Statements Class Name (optional):

If the AuthorizationDecisionStatements of the issued SAML2 assertion need to be customized, implement the `org.forgerock.openam.sts.tokengeneration.saml2.statements.AuthzDecisionStatementsProvider` interface, and specify the class name of the implementation here.

Custom Attribute Mapper Class Name (optional):

If the class implementing attribute mapping for attributes contained in the issued SAML2 assertion needs to be customized, implement the `org.forgerock.openam.sts.tokengeneration.saml2.statements.AttributeMapper` interface, and specify the class name of the implementation here.

Custom Authentication Context Class Name (optional):

If the AuthnContext mapping implemented by the `org.forgerock.openam.sts.token.provider.AuthnContextMapperImpl` class, implement the `org.forgerock.openam.sts.token.provider.AuthnContextMapper` interface, and specify the name of the implementation here.

Attribute Mappings:

Current Values Remove

New Value Add

Contains the mapping of assertion attribute names (Map keys) to local OpenAM attributes (Map values) in configured data stores. Format: assertion_attr_name=ldap_attr_name

Sign Assertion:

Encrypt Assertion:

Check this box if the entire assertion should be encrypted. If this box is checked, the Encrypt NameID and Encrypt Attributes boxes cannot be checked.

Encrypt Attributes:

Check this box if the assertion Attributes should be encrypted. If this box is checked, the Encrypt Assertion box cannot be checked.

Encrypt NameID:

Check this box if the assertion NameID should be encrypted. If this box is checked, the Encrypt Assertion box cannot be checked.

Encryption Algorithm: http://www.w3.org/2001/04/xmlenc#aes128-cbc

Algorithm used to encrypt generated assertions.

KeystorePath:

Path to keystore

Keystore Password:

Confirm Keystore Password:

Encryption Key Alias:

This alias corresponds to the SP's x509 Certificate identified by the SP Entity ID for this rest-sts instance. Not necessary unless assertions are to be encrypted.

Signature Key Alias:

Corresponds to the private key of the IdP. Will be used to sign assertions. Value can remain unspecified unless assertions are signed.

Signature Key Password:

Confirm Signature Key Password:

Back to top

OpenIdConnect Token Configuration

The id of the OpenIdConnect Token Provider: http://sp.example.com:8

Token Lifetime (Seconds): 60000

Token signature algorithm: HMAC SHA 256

Algorithm used to sign issued OIDC tokens

Public key reference type: NONE

For tokens signed with RSA, how should corresponding public key be referenced in the issued jwt

KeyStore Location:

For RSA-signed tokens, the filesystem or classpath location of the KeyStore containing singing key entry

KeyStore password:

Confirm KeyStore password:

KeyStore signing key alias:

For RSA-signed tokens, corresponds to the private key of the OIDC OP. Will be used to sign assertions.

Signature key password:

Confirm signature key password:

Client secret: ••••••••••••••••••••••••••••••

For HMAC-signed tokens, the client secret used as the HMAC key

Confirm client secret: ••••••••••••••••••••••••••••••

The audience for issued tokens:

Current Values http://sp.example.com Remove

New Value Add

Contents will be set in the aud claim

The authorized party (optional):

Optional. Will be set in the azp claim

Claim map:

Current Values Remove

New Value Add

Contains the mapping of OIDC token claim names (Map keys) to local OpenAM attributes (Map values) in configured data stores.Format: claim_name=attribute_name

Custom claim mapper class (optional):

If the class implementing attribute mapping for attributes contained in issued OpenIdConnect tokens needs to be customized, implement the `org.forgerock.openam.sts.tokengeneration.oidc.OpenIdConnectTokenClaimMapper` interface, and specify the class name of the implementation here.

Custom authn

context mapper class (optional):

If issued OIDC tokens are to contain acr claims, implement the `org.forgerock.openam.sts.rest.token.provider.oidc.OpenIdConnectTokenAuthnContextMapper` interface, and specify the class name of the implementation here.

Custom authn methods references mapper class (optional):

If issued OIDC tokens are to contain amr claims, implement the `org.forgerock.openam.sts.rest.token.provider.oidc.OpenIdConnectTokenAuthMethodReferencesMapper` interface, and specify the class name of the implementation here.

Back to top

Save Reset Back