JWT / Révocation de jetons

(V14.4.2)

La révocation de jetons JWT par le serveur de ressources (application exposant les webapis REST) est un élément de sécurisation.
Un jeton dont l'utilité est devenue caduque alors que sa durée de validité n'est pas révolue doit être révoqué. Ainsi il ne peut être utilisé pour s'authentifier malicieusement auprès d'un service.
L'opération de révocation est réalisée par le biais d'une requête explicite à une webapi dédiée.

Note/mise en garde : l'implémentation de la révocation des jetons par le serveur de ressources casse le paradigme d'une application stateless scalable.
Les informations de révocation ne sont pas partageables entre deux instances de l'application.

1. Configuration de la gestion de révocation

La gestion de la révocation n'est pas active par défaut. Pour la rendre opérationnelle, il faut déclarer - dans le fichier *Beans.xml* - un bean de type *JwtRevokeConfiguration*

jwtRevokeConfiguration

```
<bean id="jwtTokensRevocation" class="com.hardis.adelia.webservice.JwtRevokeConfiguration" init-method="
loadTokensMap" destroy-method="saveTokensMap" />

ou

<bean id="jwtTokensRevocation" class="com.hardis.adelia.webservice.JwtRevokeConfiguration" init-method="
loadTokensMap" destroy-method="saveTokensMap">
      <property name="jwtClaimId"               value="???"/> <!-- default value: jti -->
      <property name="jwtPurgeFrequencyTime"    value="???"/> <!-- default value: 3600 -->
</bean>
```

- jwtClaimId : nom du claim identifiant le jeton JWT. Valeur par défaut : "jti'. Cette information est obligatoire : lorsque la gestion de la révocation est activée, la validation d'un jeton - identifié par son id - est tributaire de sa non révocation.
  Note : si le jeton ne possède pas le claim désigné par la propriété *jwtClaimId* alors la validation du jeton échouera ; une erreur 401 sera alors retournée par le serveur.
- jwtPurgeFrequencyTime : Fréquence (en s) de purge de la table des jetons révoqués. Valeur par défaut : 3600. A adapter en rapport à la durée de la validité des jetons.

  Qui a le droit de révoquer un jeton ?
  Le jeton peut être révoqué uniquement par une requête authentifiée par ce même jeton.

  Persistence de la table des jetons révoqués
  La table des jetons révoqués subsiste à un redémarrage du serveur de ressources grâce aux méthodes d'initialisation et de terminaison du bean *JwtRevokeConfiguration* :
  - init-method : loadTokenMaps
  - destroy-method : saveTokensMaps

  Purge de la table des jetons révoqués
  La table des jetons révoqués est purgée à une fréquence donnée (par défaut toutes les heures, Cf. jwtPurgeFrequencyTime). Les jetons révoqués purgés sont les jetons révoqués arrivés à expiration.

2. Webapis

Purpose	Api Endpoint	HTTPVerb	Response	Status
Revocation	/<ContextPath> /<CXFServletPath> **/tokens /revocation**	@DELETE	Media-type : text/plain Values : true | false	200 401 404

Is a token revoked ?	/<ContextPath>/<CXFServletPath>/**tokens/revocation/{jwtId}**	@GET	Media-type : text/plain Values : true | false	200 401 404
Revoked tokens list	/<ContextPath>/<CXFServletPath>/**tokens/revocation/list**	@GET	Media-type : application/json [{ "jwtId": "TokenId__FE706DC72E90060E9E88FB887ACB72E1_25_1619685311434", "revokedBy": "userName", "revocationRequestDate": "2021-04-29T08:35Z", "expirationDate": 1619688911 }, { "jwtId": "TokenId__FE706DC72E90060E9E88FB887ACB72E1_28_1619685265807", "revokedBy": "userName", "revocationRequestDate": "2021-04-29T08:35Z", "expirationDate": 1619688865 }]	200 401 404

HTTP Code :

- Si l'opération aboutit : HTTP Code = 200
- Si l'opération échoue car la requête de révocation n'est pas autorisée : HTTP Code = 401
- Si le bean JwtRevokeConfiguration n'est pas déclaré (= révocation non active) ou si le jeton interrogé n'est pas trouvé : HTTP Code = 404.

(V14.9.0)

Ajout d'une implémentation "distribuée" de la révocation des jetons JWT à l'aide de NATS/JetStream.
Si au moins un serveur Nats est défini et qu'une connexion à celui-ci est établie alors le mode distribué est activé (sous condition de la création réussie ou de l'existence du *stream* et du *sujet*).
Dans ce mode chaque demande de révocation émet un message dans le *sujet* ${jwtNatsSubject} du *stream* ${jwtNatsStreamName}. Les messages sont conservés dans le *stream* pour la durée définie par ${jwtNatsStreamMaxAge}.
Le message est une chaine au format UTF-8 composée de 4 champs délimités par le caractère **;** Exemple : <JwtId>;<RevokedBy>;<RevocationRequestDate>;<TokenExpirationDate>

- <JwtId> : String> Id du jeton
- <RevokedBy> : String Nom de l'utilisateur à l'initiative de la révocation ('sub' du jeton utilisé lors de la révocation)
- <RevocationRequestDate> : String ; ISO-8601 Date de la demande révocation.
- <TokenExpirationDate> : Long Date (unix time) d'expiration prévue du jeton.

jwtRevokeConfiguration - Nats

```
<bean id="jwtTokensRevocation" class="com.hardis.adelia.webservice.JwtRevokeConfiguration" init-method="
loadTokensMap" destroy-method="saveTokensMap">
        <property name="jwtNatsServers"                value="localhost:4222"/> <!-- si un ou plusieurs
serveurs sont définis alors le mode partagé est activé sous condition d'une connexion réussie à l'un des
serveurs -->
        <property name="jwtNatsStreamName"               value="ADELIASTREAM"/> <!-- Nom du stream dédié ;
défaut : "ADELIASTREAM" -->
        <property name="jwtNatsSubject"                   value="streaming.adelia.internal.jwt.revoke"/> <!--
Nom du sujet ; défaut : "streaming.adelia.internal.jwt.revoke" -->
        <property name="jwtNatsStreamMaxAge"         value="12"/> <!--en heures, âge maximal des messages dans
le stream - valeur à fixer au regard de la durée de vie des jetons ; défaut: "24"-->

    <!-- Nats authentication : nkey -->
    <property name="jwtNatsAuthNKeySeed" value="CRYPT
(00D133D32D285115D308A5E89F9CEA80D227FDB3304CBD18AD596D09AA40359D7F3DA0F74CCDE20B8A6BD53687A2584851712091A26207A
7F7D790DD719753E7D7)"/>
</bean>
```

- *jwtNatsServers*: Nom du serveur Nats ou noms des serveurs Nats (séparés par un ; exemple: srvnats1:4222;srvnats2:4222).
- *jwtNatsStreamName*: Nom du stream dédié [défaut : ADELIA_JWT_TOKEN_STREAM].
- *jwtNatsSubject* : Nom du sujet utilisé pour la révocation des jetons [défaut : streaming.adelia.internal.jwt.revoke].
- *jwtNatsStreamMaxAge* : Age maximal des messages dans le stream dédié. Cet âge doit être supérieur à la durée de vie des jetons ainsi un nouveau client/abonné prend connaissance de tous les jetons révoqués possiblement actifs

Authentification au serveur Nats :

- User/Password
  jwtNatsAuthUser : Nom de l'utilisateur
  jwtNatsAuthPwd : Mot de passe. Pour chiffrer le mot de passe utilisez la commande "java EncryptPassword" du runtime Adélia, et indiquez le mot de passe entre parenthèses préfixé par "CRYPT".
  exemples :

  ```
  <property name="jwtNatsAuthUser" value="bob"/>
  <property name="jwtNatsAuthPwd" value="s3cr3t"/>

  <property name="jwtNatsAuthUser" value="bob"/>
  <property name="jwtNatsAuthPwd" value="CRYPT(0040B783FDB007E032)"/>
  ```

- Token
  jwtNatsAuthToken : Pour chiffrer le token utilisez la commande "java EncryptPassword" du runtime Adélia, et indiquez le mot de passe entre parenthèses préfixé par "CRYPT"
  exemples :
  ```
  <property name="jwtNatsAuthToken" value="s3cr3t"/>
  <property name="jwtNatsAuthToken" value="CRYPT(0040B783FDB007E032)"/>
  ```

- Nkey
  jwtNatsAuthNKeySeed : clé privée. Pour chiffrer la clé utilisez la commande "java EncryptPassword" du runtime Adélia, et indiquez le mot de passe entre parenthèses préfixé par "CRYPT"
  exemples :
  ```
  <property name="jwtNatsAuthNKeySeed" value="SUAFGCEVJB6DTYFPEQIC4DSLGBRSD75BMQPO2HS6SX4FNPVZDTHMQYYN24"/>
  <property name="jwtNatsAuthNKeySeed" value="CRYPT
  (00D133D32D285115D308A5E89F9CEA80D227FDB3304CBD18AD596D09AA40359D7F3DA0F74CCDE20B8A6BD53687A2584851712091A2
  6207A7F7D790DD719753E7D7)"/>
  ```

- Credentials file

  jwtNatsAuthCredsFile : emplacement du fichier contenant les informations d'authentification. Cf/ https://docs.nats.io/using-nats/developer/connecting/creds

  exemple :
  ```
  <property name="jwtNatsAuthCredsFile" value="nats.creds"/>
  ```

(V14.10.0)

Possibilité de préciser une liste de noms de claim pour la propriété *jwtClaimId* : utiliser le ; comme séparateur.
Le jeton est identifié par le premier nom de claim de la liste valide pour le jeton.

jwtClaimId

```
<bean id="jwtTokensRevocation" class="com.hardis.adelia.webservice.JwtRevokeConfiguration" init-method="
loadTokensMap" destroy-method="saveTokensMap">
...
      <property name="jwtClaimId"              value="jti;tid"/> <!-- liste des noms de claim pouvant
identifier le jeton -->
...
</bean>
```