SAML v2

Dépendance dans cas/build.gradle:

$home/etc/cas/build.gradle

```
dependencies {
    ...
    compile "org.apereo.cas:cas-server-support-saml:${project.'cas.version'}",
    compile "org.apereo.cas:cas-server-support-saml-idp:${project.'cas.version'}"
    ...
}
```

Afin d'utiliser ce protocole, il vous sera nécessaire d'avoir un client SAML2. Vous pouvez utiliser le projet cas-implicit-grant

Configuration minimal de CAS

$home/etc/cas/config/cas.properties

```
# SAML IDP
# See metadata https://my-cas/cas/idp/metadata
cas.authn.samlIdp.entityId=https://my-cas.fr/cas/idp
cas.authn.samlIdp.scope=my-cas.fr
cas.authn.samlIdp.metadata.location=file:/home/ubuntu/workspace/etc/cas/config/saml
```

Faites attention que certaines valeurs soit bien renseignées dans le fichier de configuration CAS:

$home/etc/cas/config/cas.properties

```
cas.server.name=https://my-cas.fr
cas.server.prefix=https://my-cas.fr/cas
```

Dans le dossier “/home/ubuntu/workspace/etc/cas/config/saml”, des fichiers “crt” and “key” ainsi que le fichier de metadata.xml seront automatiquement générés par CAS.

Cette configuration s'accompagne d'un fichier de service à placer dans cas/src/main/resources/my_services

my_services/saml.json

```
{
  "@class" : "org.apereo.cas.support.saml.services.SamlRegisteredService",
  "serviceId" : "passport-saml", doit correspondre a l'option "issuer" du client
  "name" : "SAMLService",
  "id" : 9996,
  "evaluationOrder" : 10,
  "metadataLocation" : "https://my-service.fr/metadata.xml"
}
```

Configuration du client

Dans notre cas nous utilisons un client javascript cas-implicit-grant.

config.js

```
var casUrl = 'https://my-cas.fr/cas';
var loginHost = 'my-service.fr';
var loginUrl = 'https://' + loginHost;

// Cette clef doit etre remplacer par une clef provenant du serveur CAS, provenant du champs *<KeyDescriptor
use="signing">*
var idpCert = 'My-Key';

module.exports = {
  development: {
    app: {
      name: 'Cas test login platform protocol',
      port: process.env.PORT || 8080
    },
    passport: {
      serverBaseURL: loginUrl,
      casLogoutUrl: casUrl + '/logout',
      cas2: {
        confName: 'cas2',
        ssoBaseURL: casUrl,
        serverBaseURL: loginUrl

      },
      cas3: {
        confName: 'cas3',
        version: 'CAS3.0',
        ssoBaseURL: casUrl,
        serverBaseURL: loginUrl
      },
      saml1: {
        confName: 'saml1',
        version: 'CAS3.0',
        ssoBaseURL: casUrl,
        serverBaseURL: loginUrl,
        useSaml: true
      },
      saml2: {
        confName: 'saml',
        host: loginHost,
        path: process.env.SAML_PATH || '/assert',
        // Cette option peut etre remplacer par la methode désirée (voir metadata de CAS), il faudra changer
logoutUrl de la même manière
        entryPoint: process.env.SAML_ENTRY_POINT || casUrl + '/idp/profile/SAML2/Redirect/SSO',
        issuer: 'passport-saml', doit correspondre a l'option "entity_id" du serveur
        cert: process.env.SAML_CERT || idpCert,
        protocol: 'https://',
        decryptionPvk : idpCert,
        // disable expiration
        acceptedClockSkewMs: -1,
        logoutUrl: casUrl + '/idp/profile/SAML2/Redirect/SLO'
      }
    }
  }
};
```