# Oauth2

Af: Nikolaj

Oauth er et verificeringsframework, der giver en applikation begrænset adgang til bruger data fra en HTTP service, som fx Facebook, Github osv.   
Frameworket virker ved at overføre bruger verifikationen, til den service hvorpå bruger informationerne ligger. Herefter skal brugeren give applikationen tilladelse til at tilgå deres information.  
  
Der er 4 roller i Oauth:

1. Bruger
2. Klienten
3. Autentificering server
4. Ressource server

## Bruger

Brugeren også kaldet ressource ejer, er den bruger som giver en applikation tilladelse til at tilgå deres information.

## Klienten

Klienten er den applikation der gerne vil tilgå brugerens informationer. Før den kan dette skal den have tilladelse fra brugeren, og valideres af HTTP servicen.

## Ressource / Autentificering server

Ressourceserveren er der hvor brugerens information er lageret, og Autentificering serveren verificere identiteten af brugeren og returnere en access token til klienten.

# Oauth flow



1. Applikationen laver et login request til brugeren
2. Hvis brugeren giver sine logininformationer til applikationen sender han en godkendelse.
3. Applikationen requester en access token fra autentificering serveren, ved at sende sin egen identitet med og godkendelsen fra brugeren.
4. Hvis applikationens identitet verificeres og bruger godkendelsen er valid, sender autentificeringsserveren en access token til applikationen.
5. Applikationen sender et request til ressourceserveren, og vedhæfter sin access token.
6. Hvis det er en valid access token returnere ressourceserveren de informationer applikationen requester.

# Applikation registrering

Før man kan bruge Oauth på sin applikation skal den registreres på den service hvor man vil hente data fra fx Xena.

Servicen skal bruge nogle informationer om applikationen:

* Applikations navn
* Applikations website
* Callback URL

Callback URL er der hvor man bliver overført til når autentificeringen er godkendt.

# Vores valg

Vi har valgt at bruge Oauth på vores applikation, da det giver en høj sikkerhed og giver os adgang til alle de data vi skal bruge fra Xenas ressource server.  
Vi har i et tidligere projekt arbejdet direkte med Xena omkring Oauth, og derfor er det et klart valg at vi udnytter vores viden omkring denne verificerings service i dette projekt.

## Andre muligheder

Vi havde en anden mulighed for verificering ved Xena. Dette var API-Keys, API-Keys er mindre sikkert end Oauth, da det er krævet at der bliver sendt en key med i hvert request til Xena APIen, Xena og vores applikation kan derved heller ikke være sikker på hvem der sender disse requests, da det kan falsificeres hvis man får fingrene i api-keyen. Hvorimod med Oauth tildeles man en access token, der er unik for den enkelte bruger så man altid kan identificere hvem der laver hvilke requests. Oauth giver også brugeren mere tryghed, da det er noget man kan genkende fra andre applikationer.