



Links –

1. <https://dzone.com/articles/sso-login-key-benefits-and-implementation>
2. <https://codeburst.io/building-a-simple-single-sign-on-sso-server-and-solution-from-scratch-in-node-js-ea6ee5fdf340>
3. <https://support.google.com/a/answer/6262987?hl=en>

**SSO Implementation**

1. User tries to access example1.com. The Server of example1 has middleware that checks if servers knows me and am I authorized. For that it checks my cookie if it has token. If doesn’t exists, it shows that user is not logged to this server.
2. We check if token is present in the query params (if user is authenticated and is coming from auth server). If not, user needs to be redirected to auth server with the request url being passed as query params so that auth server can be redirected back to the requested page.
3. In auth server, the request is passed through middleware where it checks if incoming request from the user browser has cookie set with auth server (i.e. Global session is maintained or not) Or the token that is maintained is expired or not. If Global session doesn’t exist then the user needs to login to auth server and creates the Global session.
4. User Redirected to the login page of auth server with serviceUrl passed to all actions.
5. Once user submit the form of auth server with creds, it goes to signin method and checks if the user is valid user or not.If user exists, it creates a token with bare min detail required to be setup for auth server, and the request is redirected from the user browser to service url(requested page) with the global token set to cookie as well as passed to query params so that the underlying application can take this token and can identify get the user details for the application.
6. Again the middleware is called in the requested application after redirection, where it checks if the cookie exists b/w user and the requested service which doesn’t exists, so it checks if the incoming redirected request has token in query params, i.e. first time a global session is established and a local session needs to be established b/w user and the application using this global session token to identify user.
7. Rest call is made to auth server to get the detail of the user to identify, then u can fetch the user detail for the application using the id, creates a token and set it to cookie to create a local session b/w user and browser. The request is redirected to the page requested. If token verification is failed on token expiration, request is redirected to auth by making query param for token exp as 1 to reset the global session b/w user and auth sso server.
8. When a new application is requested, again same middleware is called and checked if local session is established, which is not. The request will be redirected to auth server where a global session is established, so redirects back to the requested page with global token in query param.
9. Now, rest call is made to auth server to get the identity of the user,and from that id, checks and sets the user details for that application b/w user and the application(local session).