**Authentication Proxy**

Table of Contents

[Objective 1](#_Toc507692637)

[Authentication Proxy Overview Diagram 1](#_Toc507692638)

[CryptoJS 2](#_Toc507692639)

[Application Overview: 2](#_Toc507692640)

[API’s on PB Authentication Proxy: 3](#_Toc507692641)

[SalesForceToken – POST Request for Getting Salesforce Authentication Token 3](#_Toc507692642)

[SalesForceUserInfo – POST Request for Getting Salesforce User Info by Authentication Token 4](#_Toc507692643)

[SalesForceLogout – GET Request for Revoking Salesforce Authentication Token for Logout 6](#_Toc507692644)

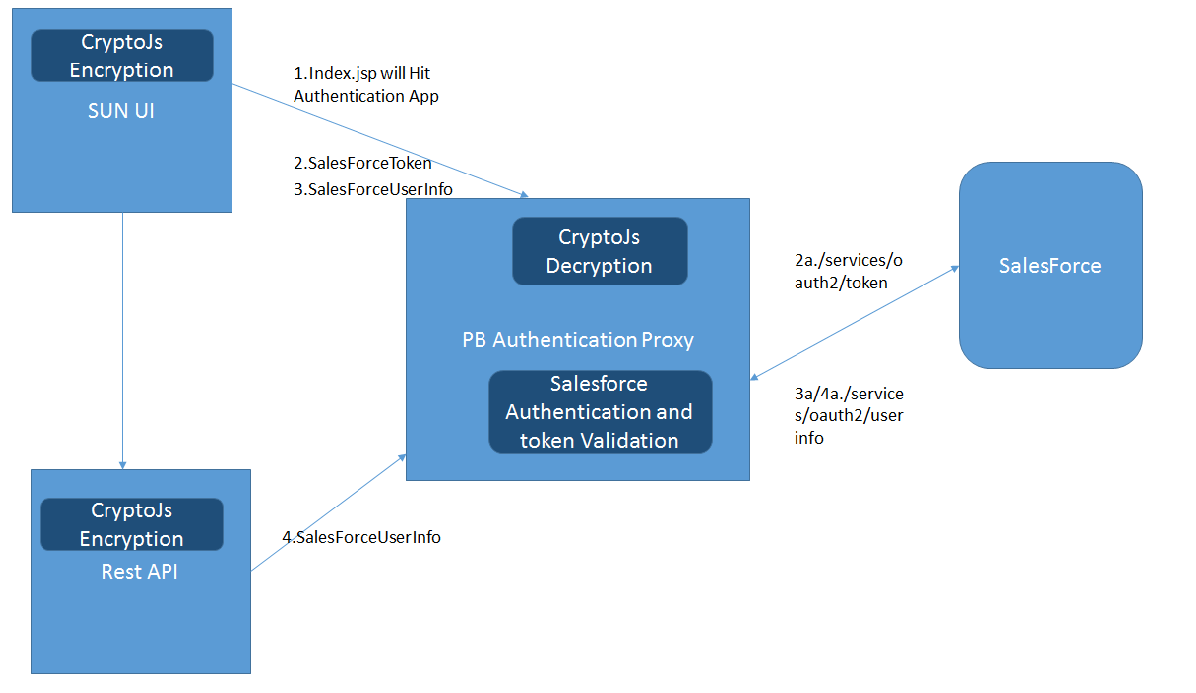
# Objective

Objective of this Application is remove multiple connected apps in CS portal and ability to avoid multiple CS Portal Environments.

# Authentication Proxy Overview Diagram

Currently we have different connected apps for different environments like sundev, sunqa etc...

By using authentication proxy we will use only one connected app and all UI Apps will hit authentication proxy it will remember the url it will give revert back to corresponding UI Application.



# CryptoJS

CryptoJS is JavaScript implementations of standard and secure cryptographic algorithms. They are fast, and they have a consistent and simple interface.

In cryptojs we are using PBKDF2 Algorithm. PBKDF2 is a password-based key derivation function. In many applications of cryptography, user security is ultimately dependent on a password, and because a password usually can't be used directly as a cryptographic key, some processing is required.

UI Application and REST App will generate a new encrypted key every time while calling any API as shown in Auth Proxy over Diagram. Authentication proxy will decrypt the key is valid or not and it will check Access token is valid or not from salesforce.

# Application Overview:

The Authentication Proxy server is used for common authentication with One Salesforce Connected app using PB Application.

Previously for every PB application instance we need to connect the application to different Salesforce connected app for each PB application.

This was creating some overhead on having new connected app created on salesforce when deploying new PB application for Development or Testing and there was a significant time spent on onboarding the user to DB.

So the PB Authentication Proxy application will be deployed for mapping one salesforce connected app and making all the user in that salesforce connected app able to login in the multiple PB application connected to the PB Authentication Proxy.

By this we can have one connected app and one PB Authentication Proxy (Authdev) for entire development environment including SunDev, SunQA, PBPilotQA and SunSystem etc…. And we can have another PB Authentication Proxy for pilot and production environments with another connected app targeting only pilot and production customers.

Even though the all users belong to salesforce connected app mapped to PB Authentication proxy can login to PB application may not able to use the application unless the users are on boarded to the corresponding PB application DB.

Also the PB Authentication Proxy used for validating the salesforce token by PB application and other application securely.

# API’s on PB Authentication Proxy:

### SalesForceToken – POST Request for Getting Salesforce Authentication Token

**API URL -** /SalesForceToken?code=code\_value

**Input:**

|  |  |
| --- | --- |
| Param1 in Header - authkey | Unique Crypto Key generated for API authorization |
| Param2 in Query String - Code | Salesforce Code Received on Login using PB Authentication Proxy |

**Output:**

{    
   **"access\_token"**:"access\_token\_value",  
   **"signature"**:"signature\_value",  
   **"scope"**:"web openid api id",  
   **"id\_token"**:"id\_token\_value",  
   **"instance\_url"**:"https://xxx.my.salesforce.com",  
   **"id"**:"https://login.salesforce.com/id/xxxxxxxxx/xxxxxxxxxx",  
   **"token\_type"**:"Bearer",  
   **"issued\_at"**:"xxxxxxxxxxxxx"  
}

The output is received from Salesforce that has been sent to the PB Application for using the access\_token as OAuth token for the other API’s in PB application.

If any params are not passed API will send back **403** with ‘**Missing\_OAuth\_Token**’ message.

If authkey is invalid then API will send back **403** with ‘**Bad\_AuthKey**’ message.

If code is invalid then API will send back following response with **400**,

{    
   **"error"**:"invalid\_grant",  
   **"error\_description"**:"invalid authorization code"  
}

### SalesForceUserInfo – POST Request for Getting Salesforce User Info by Authentication Token

**API URL -** /SalesForceUserInfo

**Input:**

|  |  |
| --- | --- |
| Param1 in Header - authkey | Unique Crypto Key generated for API authorization |
| Param2 in Header - authorization | OAuth access\_token - Received on API - SalesForceToken |

The authorization header value should have in format - OAuth access\_token.

**Ex:** If received access\_token is 123456789 then the value for authorization would be **OAuth 123456789**

**Output:**

{    
   **"sub"**:" https://login.salesforce.com/id/xxxxxxxxx/xxxxxxxxxx",  
   **"user\_id"**:"xxxxxxxxxxxxx",  
   **"organization\_id"**:"xxxxxxxxxxx",  
   **"preferred\_username"**:"xxxxxxxxxxxxxxx@philips.com",  
   **"nickname"**:"PerformanceBridge User",  
   **"name"**:"PerformanceBridge User",  
   **"email"**:"xxxxxxxxxxxxxx@philips.com",  
   **"email\_verified"**:true,  
   **"given\_name"**:"PerformanceBridge",  
   **"family\_name"**:"User",  
   **"zoneinfo"**:"America/Los\_Angeles",  
   **"photos"**:{    
      **"picture"**:"https://xxx.my.salesforce.com/img/xxx.png",  
      **"thumbnail"**:"https://xxx.my.salesforce.com/img/xxx.png"  
   },  
   **"profile"**:"https://xxx.my.salesforce.com/xxxxx",  
   **"picture"**:"https://xxx.my.salesforce.com/img/xxxx.png",  
   **"address"**:{    
  
   },  
   **"urls"**:{    
      **"enterprise"**:"https://xxxx.my.salesforce.com/services/xxxx",  
      **"metadata"**:"https://xxx.my.salesforce.com/services/xxx",  
      **"partner"**:"https://xxx.my.salesforce.com/services/Soap/xxx",  
      **"rest"**:"https://xxx.my.salesforce.com/services/xxxxx",  
      **"sobjects"**:"https://xxx.my.salesforce.com/services/data/xxx",  
      **"search"**:"https://xxx.my.salesforce.com/services/data/xxx",  
      **"query"**:"https://xxx.my.salesforce.com/services/data/xxx",  
      **"recent"**:"https://xxx.my.salesforce.com/services/data/xxx",  
      **"tooling\_soap"**:"https://xxx.my.salesforce.com/xxx",  
      **"tooling\_rest"**:"https://xxx.my.salesforce.com/services/data/xxx",  
      **"profile"**:"https://xxx.my.salesforce.com/00528000005DqyrAAC",  
      **"feeds"**:"https://xxx.my.salesforce.com/services/data/xxx",  
      **"groups"**:"https://xxx.my.salesforce.com/services/data/xxx",  
      **"users"**:"https://xxx.my.salesforce.com/services/data/xxx",  
      **"feed\_items"**:"https://xxx.my.salesforce.com/services/data/xxx",  
      **"feed\_elements"**:"https://xxx.my.salesforce.com/services/data/xxx",  
      **"custom\_domain"**:"https://xxx.my.salesforce.com"  
   },  
   **"active"**:true,  
   **"user\_type"**:"xxxxx",  
   **"language"**:"en\_US",  
   **"locale"**:"en\_US",  
   **"utcOffset"**:-28800000,  
   **"updated\_at"**:"2018-02-05T06:56:10.000+0000",  
   **"is\_app\_installed"**:true  
}

The output is received from Salesforce that has been sent to the PB Application for showing the user details on the application and PB Applications API’s are using this data for validating the user while PB Applications receive any request using the Salesforce OAuth token.

If any params are not passed API will send back **403** with ‘**Missing\_OAuth\_Token**’ message.

If authkey is invalid then API will send back **403** with ‘**Bad\_AuthKey**’ message.

If OAuth Authorization token is invalid then API will send back ‘**Bad\_OAuth\_Token**’ response with **403**,

### SalesForceLogout – GET Request for Revoking Salesforce Authentication Token for Logout

**API URL -** /SalesForceLogout

**Input:**

|  |  |
| --- | --- |
| Param1 in Header - authkey | Unique Crypto Key generated for API authorization |
| Param2 in Header - authorization | OAuth access\_token - Received on API - SalesForceToken |

The authorization header value should have in format - OAuth access\_token.

**Ex:** If received access\_token is 123456789 then the value for authorization would be **OAuth 123456789**

**Output:**

If valid access\_token passed then Empty Response with 200 Status Code will be sent back.

If already expired access\_token is sent then Empty Response with 400 Status Code will be sent back.

The response is received from Salesforce that has been sent to the PB Application for proceeding with user logout even the PB application receive any response on this API since if we get 200 or 400 in both case token has been revoked or expired.

If any params are not passed API will send back **403** with ‘**Missing\_OAuth\_Token**’ message.

If authkey is invalid then API will send back **403** with ‘**Bad\_AuthKey**’ message.