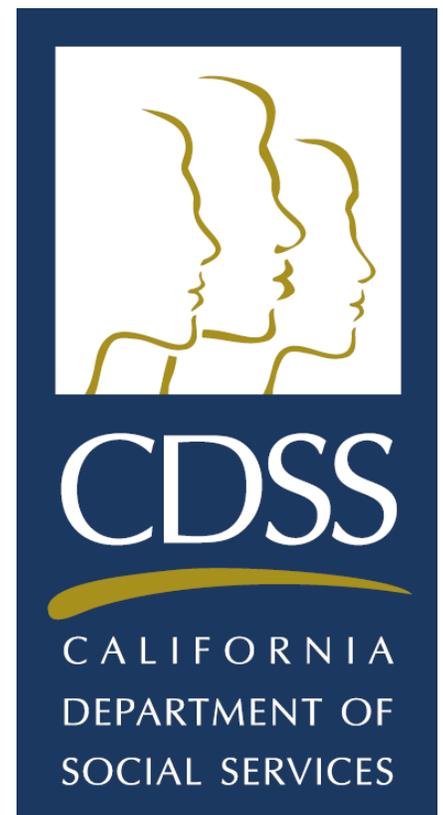


# Identity and Access Management (IdAM) Security Access Framework (SAF) for CDSS

Admin & Developer Documentation

Version 3.0.3

October 30, 2015



# Table of Contents

Revision Index .....	4
Introduction .....	6
Intended Users.....	6
Features of DSS SAF .....	6
Supported System Type .....	7
Future Enhancements.....	7
Identity and Access Management Overview .....	7
Components of SAF.....	8
Managing your Admin / Developer account.....	9
Managing your Applications .....	11
Create Application .....	11
Edit Applications .....	13
Managing Applications Properties.....	14
Request Authentication Workflow .....	16
Request Application Activation.....	17
Managing Users .....	18
Developer Instructions.....	20
Service Protocols.....	20
Service URLs.....	20
JSON and OAuth Overview .....	20
Authentication Overview .....	21
Service Integration.....	22
Response Values .....	23
Error Handling .....	25
Membership and Role Provider .....	26
Application Provider .....	28
Development and Testing.....	30
Appendix A.....	31
New Application.....	31

Application Property ..... 32

    User – Default Properties..... 34

Appendix B ..... 35

    Status Codes and Response Headers ..... 35

Appendix C ..... 36

    Definitions and Abbreviations ..... 36

Appendix D..... 37

    Password Policy & Strong Password ..... 37

## Revision Index

Version 1	05/13/2015	GP/IPV
Version 1.1	06/17/2015	GP
Version 2.0.0	07/06/2015	IPV
Version 3.0.0	08/07/2015	IPV
Version 3.0.1	08/17/2015	IPV
Version 3.0.2	10/01/2015	IPV
Version 3.0.3	10/30/2015	IPV

[PAGE INTENTIONALLY LEFT BLANK]

## Introduction

DSS SAF is a modern suite of web applications and services that provide a scalable and dynamic solution for Authentication, Authorization and Auditing (AAA). Services allow for effective management of the end-to-end lifecycle of user identities across all participating CDSS resources. Authentication mechanism is technology agnostic and can be consumed by a multitude of programming languages (C#, VB, JavaScript, Java, etc) and platforms (Web, Windows, Console, Mobile). SAF emphasizes decoupling of the authentication mechanism from an application, allowing it to focus on authorization. This reduces the standard application development commitment by as much as 20%. Provides a centralized and singular user experience (UX) for; Application User Administrators for access management, Application Developers for development and configuration, Department policy administration and enforcement.

Without SAF, multiple user accounts must be managed within each system or applications. Modification of user's account / profile data must be performed manually in each system. Audit trail/log is managed separately as well.

## Intended Users

DSS SAF provides rapid integration security, next generation of authentication methods, and it is scalable from few users to thousands of users.

- Developers
- Internal State User
- External County User
- External State User
- Public

## Features of DSS SAF

With SAF you can identify and validate 'who has access to what' and effectively enforce least-privileged access across multiple applications.

- Central control of users and security credentials.
- Multi-Factor Authentication.
- Business Intelligence.
- Offers an intuitive Web-based interface.
- Reduce the risk associated with unauthorized access and meet stringent state and federal government requirements.

## Supported System Type

DSS SAF is implemented as a Web API; it is in compliance with Service Oriented Architecture (SOA).

1. Cloud computing
2. Multi-Tier applications
3. Web, Windows and Console applications
4. Mobile applications

## Future Enhancements

SAF will be the core component of Department of Social Service's enterprise security solution. With future enhancements SAF will address the following:

1. Single Sign On (SSO)
2. Easily integrates with existing mobile applications
3. Visibility of Audit Reports per User and Role
4. Visibility of Audit Report per System / Application
5. Visibility of Comprehensive audit trail of all requests and approval activities to ensure compliance with state and federal standards.

## Identity and Access Management Overview

DSS SAF addresses some of the fundamental security requirements; Identity and Access Management. It is developed and maintained by the California Department of Social Services (*Information System Division / Technical Service Branch / Application & Production Services Bureau*) and provides the easiest way to integrate security controls with your application.

We're continuously enhancing the DSS SAF services. If you have something you'd like to see, or provide us any recommendation, please let us know [ISDSecuritySupport@dss.ca.gov](mailto:ISDSecuritySupport@dss.ca.gov)

## Components of SAF

**SecurityUI:** Internally accessible Web Application where; CDSS Developers can create, configure and manage applications settings and CDSS State Administrators can create, edit and delete applications users

**Security\_Lite:** Externally accessible Web Application variant of SecurityUI, where; Delegated Administrators (County, Other Entities) can create, edit and delete applications users (where applicable).

**AuthService:** Restful API for generation of token. An applications custom authentication workflow determines incoming/outgoing parameters and return objects.

**DataService:** Restful API for consumption of SAF Data. Client Membership and Client Role extensions provide real time user information lookup capabilities.

**IdentityPortal:** Externally accessible Web Application where Application users can manage each of their application profiles. Users can; Activate accounts, unlock accounts, update user profile information, change passwords and resets passwords as well as manage Multifactor authentication information.

## Managing your Admin / Developer account

Utilization of SAF requires a developer to create and activate their developer account. It allows a developer to be assigned to existing app(s), create new app(s), add application properties and lets the developer manage users for their app. Without this account you won't be able to register your application to utilize these services. Please use the following steps to request a developer account.

**Note:** Multiple applications can be registered under a single developer account. If you already have an developer account but need to register a new application, please skip the steps below and proceed to the 'Managing your application' section; on page 7.

To request a new developer's account, please proceed to the following URL:

[https://----Will\\_be\\_provided\\_in\\_future\\_revision\\_of\\_this\\_doc.\\_----/SecurityUI](https://----Will_be_provided_in_future_revision_of_this_doc._----/SecurityUI)

- a. Note: If you already have an account, same URL would be used to sign-in into your account.

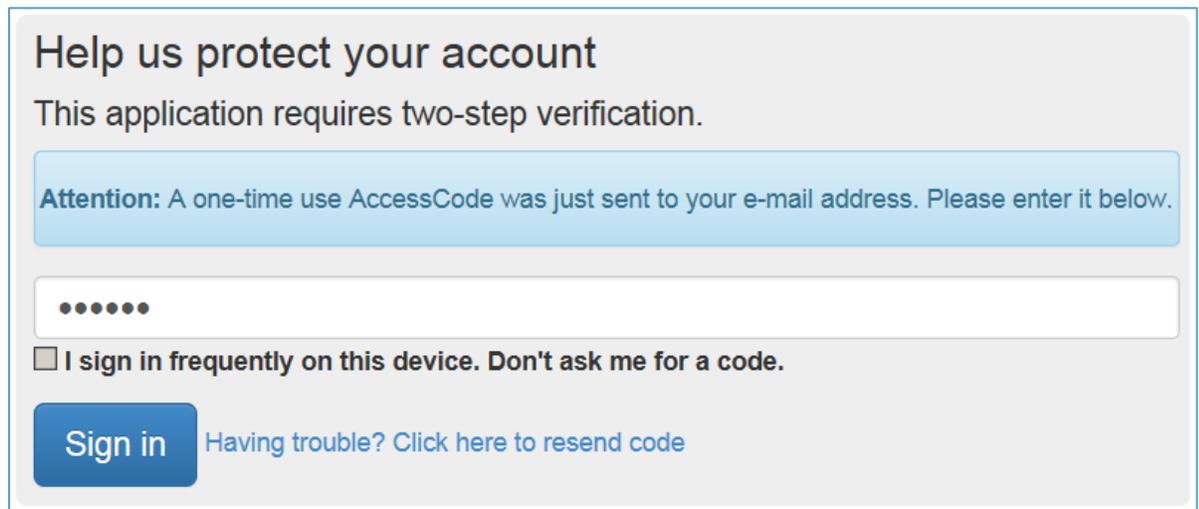


- b. To create a new account click on 'Register' button.
- c. In the next window (Register for Access to SecurityUI), Enter your username and an email address.

- d. To submit the user name and work email address ([first.last@dss.ca.gov](mailto:first.last@dss.ca.gov)) click the Register button.



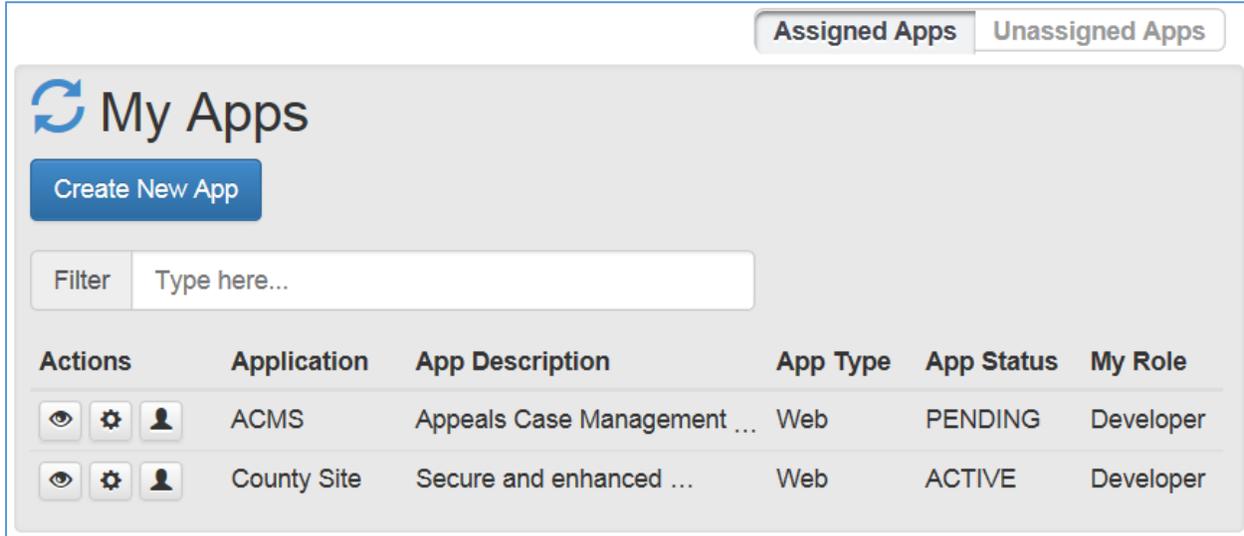
- e. If successful, you will be redirected to LOGIN.
- f. Enter your Username (from Registration page)
- g. You will receive a onetime use access code in your email address (provided in registration).
- h. Once you receive an email, follow the instruction and enter the onetime access code, as shown the following screenshot.
- i. Upon clicking on 'Sign In' button you will be redirected to homepage of DSS SAF.



## Managing your Applications

Under a single developer, multiple applications can be authorized to utilize the security controls provided by the DSS SAF. As posted in the screenshot below:

Note: For a new account the App list will be empty.



The screenshot shows the 'My Apps' interface with two tabs: 'Assigned Apps' and 'Unassigned Apps'. The 'Assigned Apps' tab is active. Below the tabs is a 'Create New App' button and a search filter box labeled 'Filter' with the text 'Type here...'. Below the filter is a table with the following data:

Actions	Application	App Description	App Type	App Status	My Role
  	ACMS	Appeals Case Management ...	Web	PENDING	Developer
  	County Site	Secure and enhanced ...	Web	ACTIVE	Developer

## Create Application

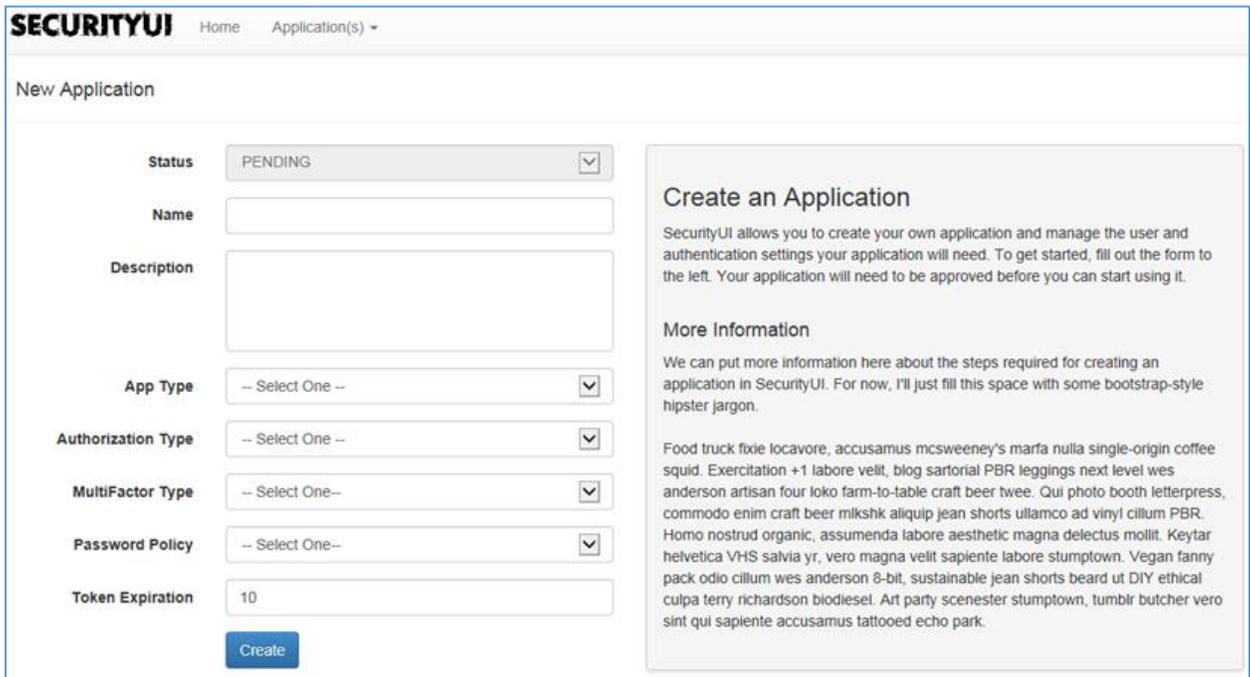
- a. To Create a new Application, click Create New App on the home window.

Note: In the 'New Application' window, please fill all the columns and submit new application for review. The 'Status' column is already pre-populated with 'Pending' status. Upon submitting your application, it will be reviewed and approved by DSS Information System Division (ISD). Upon approval, the status of the application will be changed to 'Active' and you will be notified via email.

Follow the steps below to register the new application. Enter the data into following required fields. For additional details regarding these columns, please refer to *Appendix A: New Application* of this document.

- b. *Name*: Enter the name of the application.
- c. *Description*: Enter the description of your application.
- d. *App Type*: Select the application type from the drop-down menu.
- e. *Authorization Type*: Select the authorization type from the drop-down menu.
- f. *Multifactor Type*: Select Multi-Factor from the drop-down menu.
- g. *Password Policy*: Select Password Policy from the drop-down menu.

- h. *Token Expiration*: Select your Token Expiration from the drop-down menu.
- i. Click the Create Button.



**SECURITYUI** Home Application(s) ▾

New Application

Status: PENDING ▾

Name:

Description:

App Type: -- Select One -- ▾

Authorization Type: -- Select One -- ▾

MultiFactor Type: -- Select One -- ▾

Password Policy: -- Select One -- ▾

Token Expiration: 10

[Create](#)

**Create an Application**

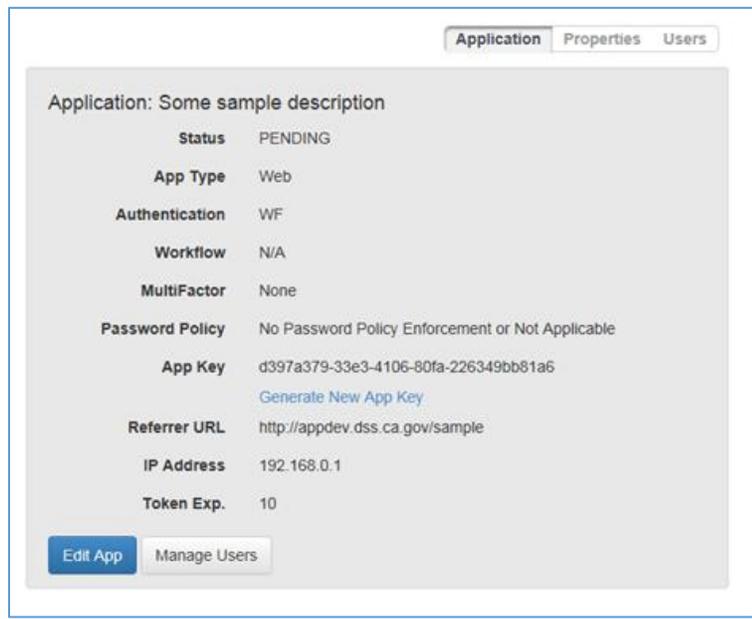
SecurityUI allows you to create your own application and manage the user and authentication settings your application will need. To get started, fill out the form to the left. Your application will need to be approved before you can start using it.

**More Information**

We can put more information here about the steps required for creating an application in SecurityUI. For now, I'll just fill this space with some bootstrap-style hipster jargon.

Food truck fixie locavore, accusamus mcsweeney's marfa nulla single-origin coffee squid. Exercitation +1 labore velit, blog sartorial PBR leggings next level wes anderson artisan four loko farm-to-table craft beer twee. Qui photo booth letterpress, commodo enim craft beer milkshk aliquip jean shorts ullamco ad vinyl cillum PBR. Homo nostrud organic, assumenda labore aesthetic magna delectus mollit. Keytar helvetica VHS salvia yr, vero magna velit sapiente labore stumptown. Vegan fanny pack odio cillum wes anderson 8-bit, sustainable jean shorts beard ut DIY ethical culpa terry richardson biodiesel. Art party scenester stumptown, tumblr butcher vero sint qui sapiente accusamus tattooed echo park.

- j. Upon clicking the on Create you will be redirected to your Application summary page (as shown below). This page will also display the connection details for your application.



Application Properties Users

Application: Some sample description

Status	PENDING
App Type	Web
Authentication	WF
Workflow	N/A
MultiFactor	None
Password Policy	No Password Policy Enforcement or Not Applicable
App Key	d397a379-33e3-4106-80fa-226349bb81a6 <a href="#">Generate New App Key</a>
Referrer URL	http://appdev.dss.ca.gov/sample
IP Address	192.168.0.1
Token Exp.	10

[Edit App](#) [Manage Users](#)

## Edit Applications

To edit the configurations of existing applications, click on the “Edit App” button as shown in the above screenshot.

Upon clicking on this button, it will redirect to the following page (screenshot below). This is where the application configuration can be edited and saved.

<b>Status</b>	ACTIVE ▾
<b>Name</b>	<input type="text"/>
<b>Description</b>	<div style="border: 1px solid #ccc; height: 40px;"></div>
<b>App Key</b>	d284aa76-80e4-443f-a02f-52d626d81d33
<b>App Type</b>	Web ▾
<b>Authorization Type</b>	AD ▾
<b>MultiFactor Type</b>	None ▾
<b>Password Policy</b>	None ▾
<b>Token Expiration</b>	1D
<b>Authorization Workflow</b>	Not yet available.

### Edit an Application

SecurityUI allows you to edit your own application and manage the user and authentication settings your application will need. To get started, fill out the form to the left. Your application will need to be approved before you can start using it.

**More Information**

We can put more information here about the steps required for creating an application in SecurityUI. For now, I'll just fill this space with some bootstrap-style hipster jargon.

Food truck fixie locavore, accusamus mcsweeney's marfa nulla single-origin coffee squid. Exercitation +1 labore velit, blog sartorial PBR leggings next level wes anderson artisan four loko farm-to-table craft beer twee. Qui photo booth letterpress, commodo enim craft beer mlkshk aliquip jean shorts ullamco ad vinyl cillum PBR. Homo nostrud organic, assumenda labore aesthetic magna delectus mollit. Keytar helvetica VHS salvia yr, vero magna velit sapiente labore stumptown. Vegan fanny pack odio cillum wes anderson 8-bit, sustainable jean shorts beard ut DIY ethical culpa terry richardson biodiesel. Art party scenester stumptown, tumblr butcher vero sint qui sapiente accusamus tattooed echo park.

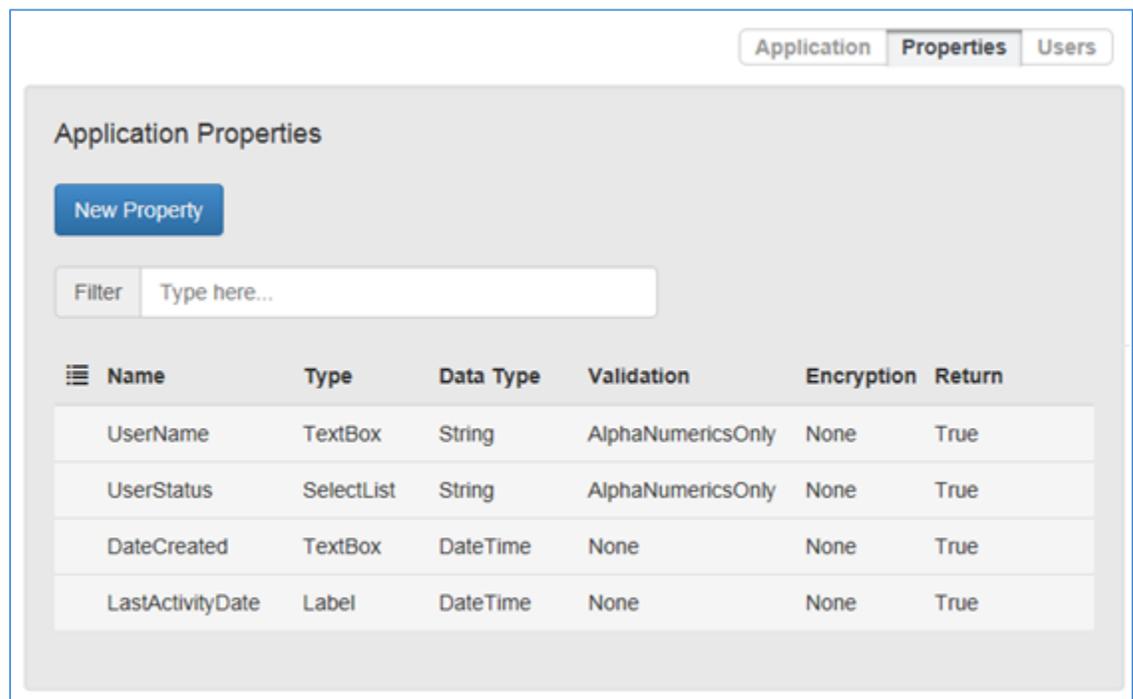
## Managing Applications Properties

DSS security SAF allows custom properties to be configured for a specific application. Application properties are utilized for defining the data elements of a specific application’s needs for the security.

These properties will be available for all users (where applicable) associated with a specific application. In order to configure a new property, follow the steps listed below.

- a. To add new application properties click on the ‘New Property’ button.

Note: All applications have the following (screenshot below) properties configured by default. These properties are read-only and cannot be modified.



The screenshot shows the 'Application Properties' section of a web application. It includes a 'New Property' button, a search filter, and a table of default properties.

Name	Type	Data Type	Validation	Encryption	Return
UserName	TextBox	String	AlphaNumericsOnly	None	True
UserStatus	SelectList	String	AlphaNumericsOnly	None	True
DateCreated	TextBox	DateTime	None	None	True
LastActivityDate	Label	DateTime	None	None	True

Follow the steps below to configure new properties for your application. Enter the data into following required fields. For additional details regarding these columns, please refer to *Appendix A: [Application Property](#)* of this document.

- b. *Status*: Select the property status from the drop-down menu.
- c. *Name*: Enter the name of the new property

- d. *Description:* Enter the description of the property.
- e. *Type:* Select the property type from the drop-down menu.
- f. *Data Type:* Select the data type from the drop-down menu.
- g. *Display Type:* Select Display-type option from the drop-down menu.
- h. *Validation:* Select Validation option from the drop-down menu.
- i. *Encryption:* Select the Encryption option from the drop-down menu.
- j. *Return:* Check the Return box to return the value during authentication.
- k. *Viewable:* Check this to allow users to view this specific property.
- l. *Editable:* Check this to allow to edit this specific property.
- m. Click the 'Create Property' Button.

**New Property**

<b>Status</b>	ACTIVE <input type="button" value="v"/>	<p><b>Creating a Property</b></p> <p>SecurityUI allows you to extend your application's properties and manage the values and relationships associated with those properties. To get started, fill out the form to the left.</p>
<b>Name</b>	<input type="text" value="Property Name"/>	
<b>Description</b>	<input type="text" value="Property Description"/>	
<b>Type</b>	<input type="button" value="v"/>	
<b>Data Type</b>	<input type="button" value="v"/>	
<b>Display Type</b>	<input type="button" value="v"/>	
<b>Validation</b>	<input type="button" value="v"/>	
<b>Encryption</b>	<input type="button" value="v"/>	
<b>Return</b>	<input type="checkbox"/> Return During Authentication	
<b>Viewable</b>	<input type="checkbox"/> Authenticated User Can See Property	
<b>Editable</b>	<input type="checkbox"/> Authenticated User Can Edit Property	
<input type="button" value="Create Property"/>		

## Request Authentication Workflow

At this time, your Applications custom authentication workflow must still be created by a SecurityUI System Administrator.

Determine application workflow:

- a. Single Pass (No multifactor authentication)
  1. AppKey
  2. Username
  3. Password (if required)
  4. Returns AToken
- b. Multi Pass (No multifactor authentication)
  1. Pass 1
    - i. AppKey
    - ii. Username
    - iii. Returns UToken
  2. Pass 2
    - i. UToken
    - ii. Password
    - iii. Return AToken
- c. Multi Pass (PicturePass Multifactor Authentication)
  1. Pass 1
    - i. AppKey
    - ii. UserName
    - iii. Returns UToken, MF Image, MF PassPhrase
  2. Pass 2
    - i. UToken
    - ii. Password
    - iii. Returns Atoken
- d. Multi Pass (Access Code)
  1. Pass 1
    - i. AppKey
    - ii. Username
    - iii. Password (If required)
    - iv. Returns UToken
  2. Pass 2
    - i. UToken
    - ii. AccessCode
    - iii. Returns AToken

## Request Application Activation

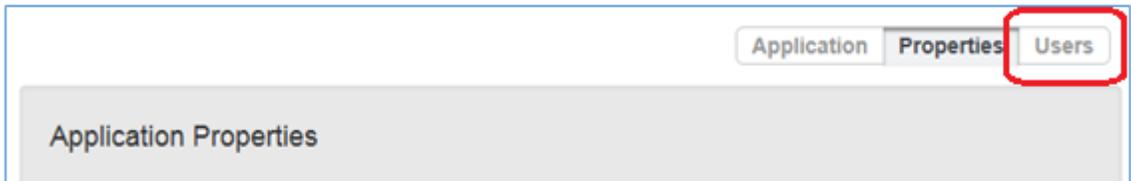
Once you have completed configuration of your Application and have determined an authentication workflow, and are ready to begin authenticating, you must request activation. Your Application will be in Pending status until this time. Any Authentication requests will fail until your Application is in an Active status.

- a. To request activation of your application, send an e-mail to [ISDSecuritySupport@dss.ca.gov](mailto:ISDSecuritySupport@dss.ca.gov).
  - a. Include
    - i. Application Name
    - ii. Authentication Workflow
  - b. Your request will be reviewed by a SecurityUI System Administrator and notification will be provided once approved.

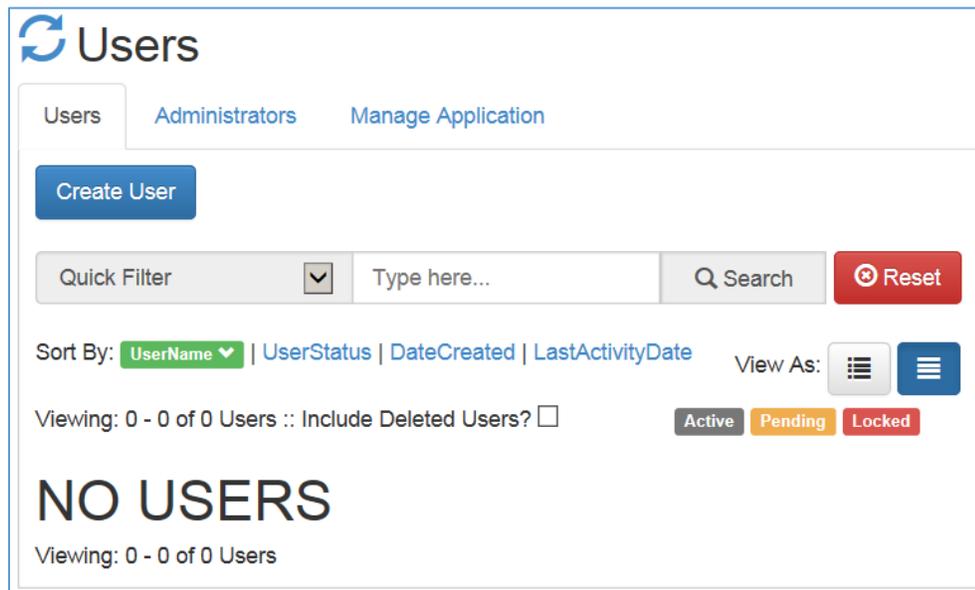
## Managing Users

DSS SAF requires users to be created for each application.

- a. To view the list of users or to add a new user click on the 'Users' tab; as shown in the screenshot.



- b. Upon clicking, it will redirect to the application 'Users' screen.



- c. To create a new user, Click on 'Create User' button. It will present a popup window. Enter the data into the required fields.
- d. *UserName*: Enter the user name (user name is an email work email address).
- e. *UserStatus*: Select the user status from the drop-down menu.

Create User
✕

**UserName**

**UserStatus**  ▼

**DateCreated**   
[Edit](#)

**LastActivityDate**

**EmailAddress**   
[Edit](#)

**E-Mail User**

Don't Send User E-Mail  
 Send User E-Mail immediately  
 Add User E-Mail to QUEUE

- f. *E-Mail User*: Select one of the radio buttons. The selected action will be performed upon creation of a new user.

# Developer Instructions

## Service Protocols

DSS SAF utilizes the OAuth 2.x Web API for authentication and authorization. SAF authentication service can be consumed by client-side, web server, installed, mobile applications. Resource Server access require Access Token (“Bearer Token”) to be provided upon every API call to SAF. For data exchange SAF is built upon open standard formats such as HTTP/HTTPS and JSON.

Note: It is required that the **token** provided by SAF is protected within your application.

Before an application can utilize SAF for user login, your application must be configured and authorized, as described in the ‘Managing your Applications’ of this document. In addition new users must be created as described in the “Managing User”.

## Service URLs

### DEVELOPMENT

[https://---Will\\_be\\_provided\\_in\\_future\\_revision\\_of\\_this\\_doc.\\_----/SecurityUI](https://---Will_be_provided_in_future_revision_of_this_doc._----/SecurityUI)  
[https:// --- Will\\_be\\_provided\\_in\\_future\\_revision\\_of\\_this\\_doc.\\_ ----/AuthService/](https:// --- Will_be_provided_in_future_revision_of_this_doc._ ----/AuthService/)

### TEST

[https:// ---- Will\\_be\\_provided\\_in\\_future\\_revision\\_of\\_this\\_doc.\\_ -----/SecurityUI](https:// ---- Will_be_provided_in_future_revision_of_this_doc._ -----/SecurityUI)  
[https:// ---- Will\\_be\\_provided\\_in\\_future\\_revision\\_of\\_this\\_doc.\\_ -----/AuthService/](https:// ---- Will_be_provided_in_future_revision_of_this_doc._ -----/AuthService/)

### PROD

[https:// ---- Will\\_be\\_provided\\_in\\_future\\_revision\\_of\\_this\\_doc.\\_ -----/SecurityUI](https:// ---- Will_be_provided_in_future_revision_of_this_doc._ -----/SecurityUI)  
[https:// ---- Will\\_be\\_provided\\_in\\_future\\_revision\\_of\\_this\\_doc.\\_ -----/AuthService/](https:// ---- Will_be_provided_in_future_revision_of_this_doc._ -----/AuthService/)

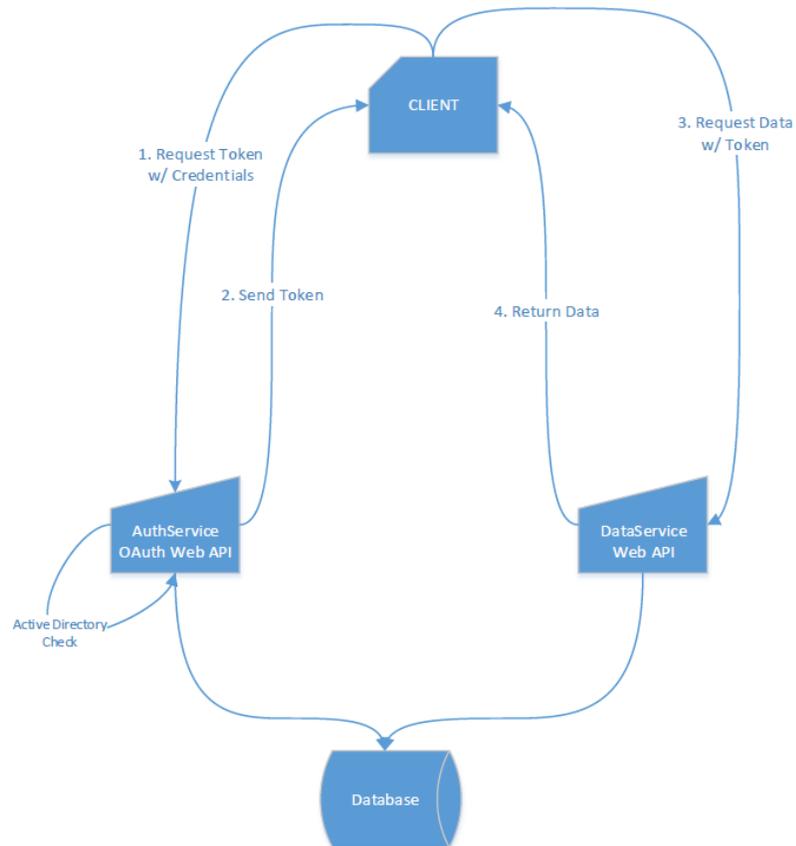
## JSON and OAuth Overview

The main technologies that are utilized by SAF are JSON and The OAuth 2.0 Authorization Framework. Additional details of these technologies is available in the following link.

JSON: <http://json.org/> or <http://en.wikipedia.org/wiki/JSON>

OAuth 2.0: <http://self-issued.info/docs/draft-ietf-oauth-v2-bearer.html>

# Authentication Overview



## Service Integration

Request parameters (Resource Owner Password Credentials Grant).

An authentication request with 'app key' is the start of a transaction that concludes when the response is returned as a JSON string. These application parameters are configured as 'Application properties'.

Type	Parameter	Values
POST	username	string
POST	password	string
POST	app key	string
POST	grant_type	string

Authentication requests are stateless; no URL rewrites are used during the exchange. A C# example of this would look as such:

Example C# code: Request

```
HttpClient client = new HttpClient();
List<KeyValuePair<string, string>> postData = new List<KeyValuePair<string, string>>();
    postData.Add(new KeyValuePair<string, string>("username", username));
    postData.Add(new KeyValuePair<string, string>("password", password));
    postData.Add(new KeyValuePair<string, string>("app key", _appKey));
    postData.Add(new KeyValuePair<string, string>("grant_type", "password"));
```

All requests are passed as JSON Data (name and value pair). JSON data is case sensitive and misspelled field name (in double quotes) will generate errors.

## Response Values

The returned content is JSON data string and token. Response message also contains the header and HTTP status code. All possible status codes are listed in the [Appendix B](#) of this document.

### Example C# Code: Response Message

```
HttpResponseMessage responseMessage = await client.PostAsync(_serviceUrl + "token",
new FormUrlEncodedContent(postData));
```

Standard response for successful status code is 200. Success codes are in two hundreds (2xx), client-side error codes are in four hundreds (4xx) and server-side error codes are in five hundreds (5xx). All possible http status codes are listed in the [Appendix B](#) of this document.

A C# example of this would look as such:

### Example C# Code: Validation of Status Code

```
if (responseMessage.IsSuccessStatusCode) {
    string _AToken = responseMessage.Content.ReadAsStringAsync().Result;
    var avalues = responseMessage.Headers.Contains("DSS-Security-Values") ?
    JsonConvert.DeserializeObject<Dictionary<string,string>>
(responseMessage.Headers.GetValues("DSS-Security-Values").FirstOrDefault()) : null;
} else {
    // code for else statement; Any Error could be handled here...
}
```

In addition to basic security authentication, SAF can also return the value of any other field configured as application property and marked as return during authentication. In the example below the "UserStatus" was defined as application property in the 'Managing Applications Properties' section of this document and the value of this property is populated in the 'Managing Users' section of this document. As indicated in this example each user will have its own 'UserStatus' value.

#### Example C# Code: Response Message

```
if (avalues.ContainsKey("UserStatus"))
{
    string status = avalues["UserStatus"];
    if(status.ToUpper().Equals("ACTIVE"))
    {
        //Enter your function code here; Example set declared Boolean to true or false.
    }
}
```

## Error Handling

Errors are handled via a response message header called “DSS-Security-Errors”. Upon receiving an error, a standard HTTP status code of 400 or 500 is returned.

There is not a collection of error messages or states that is available. The response header has the identified problem and should be used to troubleshoot the issue. Problems can arise even after thoroughly testing an application because of updates to SAF service.

### Example C# Code: Response Message – Error Handling

```
{
    Dictionary<string, string> returnErrors;
    IEnumerable<string> returnErrors2;
    responseMessage.Headers.TryGetValues("DSS-Security-Errors", out returnErrors2);
    returnErrors = JsonConvert.DeserializeObject
        <Dictionary<string, string>>(returnErrors2.FirstOrDefault());
    //Enter your function code here
}
```

## Membership and Role Provider

The DataService has been extended to provide Client Membership and Client Role provider information to the client applications. Requests are protected and only available to a valid and authenticated user. A valid AToken must be attached to the Authorization header of any DataService call by the client. Additionally, the Client ID must be provided in the Endpoint Route and it must match the Client which is encrypted in the AToken.

Example C# Code: Authorization Header AToken

```
using (HttpClient client = new HttpClient())
    client.DefaultRequestHeaders.TryAddWithoutValidation("Authorization", AToken);
```

Endpoint: DataServiceUrl + "/api/Client/{app\_id}/Membership/" **<--MUST INCLUDE APP\_ID in ROUTE**

Client Membership Methods:

*Skip = pageIndex, Take = PageSize*

```
GetAllUsers(int skip, int take)
//returns Membership_Response object, includes List of 1 or more Client_User

FindUsersByEmail(string emailToMatch, int skip, int take)
//returns Membership_Response object, includes List of 1 or more Client_User

FindUsersByName(string nameToMatch, int skip, int take)
//returns Membership_Response object, includes List of 1 or more Client_User

FindUserByUsername(string username)
//return Membership_Response object, includes List of 1 Client_User
```

Endpoint: DataServiceUrl + "/api/Client/{app\_id}/Role/" **<--MUST INCLUDE APP\_ID in ROUTE**

Client Role Methods:

*Skip = pageIndex, Take = PageSize*

```
GetAllRoles(string nameOfRoleProperty)
//returns List<string>

FindUsersInRole(string nameOfRoleProperty, string roleToMatch, int skip, int take)
//returns Membership_Response object, includes List of 1 or more Client_User

GetRolesForUser(string userName, string nameOfRoleProperty)
//returns List<string>
```

See below for the Client\_User object and the parent Membership\_Response object, which is returned in several of the Membership and Role Provider methods. Otherwise a List<string> object is returned and can be used as such.

Client\_User Class:

```
public class Client_User
{
    public Dictionary<string, dynamic> User_Properties { get; set; }
}
```

Membership\_Response Class:

```
public class Membership_Response
{
    public int returnedUsers { get; set; }
    public int totalUsers { get; set; }
    public List<Client_User> clientUserList { get; set; }
}
```

## Application Provider

The DataService has been extended to provide Client Application provider information to the client applications. Requests are protected and only available to a valid and authenticated user. A valid AToken must be attached to the Authorization header of any DataService call by the client. Additionally, the Client ID must be provided in the Endpoint Route and it must match the Client which is encrypted in the AToken.

Example C# Code: Authorization Header AToken

```
using (HttpClient client = new HttpClient())
    client.DefaultRequestHeaders.TryAddWithoutValidation("Authorization", AToken);
```

Endpoint: DataServiceUrl + "/api/Client/{app\_id}/Application/"<--MUST INCLUDE APP\_ID in ROUTE

Client Application Methods:

```
GetAppProperty(string propName)
//returns App_Properties object

GetUserProperty(string userName, string propName, bool includeAppProperty = false)
//returns Application_Response object
```

See next page for the Application\_Response and App\_Properties objects.

Application\_Response Class:

```
public class Application_Response
{
    string _userPropertyValue;
    App_Properties _appProperty;
}
```

App\_Properties Class:

```
public class Application_Response
{
    public int App_Prop_ID { get; set; }
    public int Application_ID_FK { get; set; }
    public string App_Prop_Name { get; set; }
    public string App_Prop_Description { get; set; }
    public int App_Properties_Status_ID_FK { get; set; }
    public int Prop_Type_ID_FK { get; set; }
    public int Prop_Data_Type_ID_FK { get; set; }
    public int Prop_Type_Class_ID_FK { get; set; }
    public int App_Prop_Section { get; set; }
    public int App_Prop_Order { get; set; }
    public int Validation_Type_ID_FK { get; set; }
    public int Encryption_Type_ID_FK { get; set; }
    public int Prop_Display_Type_ID_FK { get; set; }
    public bool Return_During_Auth { get; set; }
    public bool Allow_User_To_View { get; set; }
    public bool Allow_User_To_Edit { get; set; }
    public Validation_Type Validation_Type { get; set; }
    public App_Properties_Status App_Properties_Status { get; set; }
    public App_Properties_Data_Type App_Properties_Data_Type { get; set; }
    public App_Properties_Display_Type App_Properties_Display_Type { get; set; }
    public Encryption_Type Encryption_Type { get; set; }
    public Prop_Type Prop_Type { get; set; }
    public List<App_Properties_Values> App_Properties_Values { get; set; }
}
```

## Development and Testing

It is strongly recommended that during the development and testing that your application is pointed to referring to development or test URL only. Application designed for external/public users must be tested against the externally accessible links/services of SAF.

**Note:** *Throttling and Threat Detection prevents Capacity or Stress Testing against **PRODUCTION** DSS SAF Services. If Capacity or Stress Testing DSS SAF Services are required by your application, please target the **TEST** DSS SAF environment and make sure to coordinate your testing with a DSS SAF Administrator to ensure other apps are not testing at the same time.*

It is highly recommended that your application is tested against the latest release of SAF; updates may generate errors that didn't occur previously. All updates to SAF will initially be applied to development then test account. These changes may need to be communicated with your admins and end users.

Appropriate documentation will be provided in a distributable manner when end users are affected by changes. Otherwise, internal only distributable documentation will be provided to developers and customers.

## Appendix A

### New Application

Details of defining new application:

Column	Type	Values
Status	Drop-Down	<b>Pending</b> Pending is the default value selected for new Applications.
Name	Text Box	[Enter the Name of Application]
Description	Text Box	[Enter the Description of Application]
App Key	Text Box	It is auto generated for specific application
App Type	Drop-Down	<b>Web</b> Select this option if you're creating a web-based application. <b>Windows</b> Select this option if you're creating a Windows desktop application. <b>Console</b> Select this option if you're creating a console application. <b>Mobile</b> Select this option if you're creating a native mobile application
Authorization Type	Drop-Down	<b>AD (Active Directory)</b> Authenticate using Active Directory <b>WF (WebForms)</b> Authenticate using a combination of Username and Password <b>MyCDSS</b> Authenticate using MyCDSS
MultiFactor Type	Drop-Down	<b>None</b> No multifactor authentication is required. <b>Email Code</b> Send a code via email for the authenticating user to enter. <b>Hardware Token</b> Validate a user using a hardware token. <b>PicturePass</b> An image is selected and presented for the user to validate on login. <b>SMS Code</b> Send a code via SMS for the authenticating user to enter.
Password Policy	Drop-Down	<b>None</b> No password policy. <b>CDSS Standard</b> This is the departments default ISO standard password policy.

		<b>CDSS Strict</b> This is a stricter policy than the departments default ISO standard password policy.
Token Expiration	Text Box	[Enter the Time in Minutes]
Authorization Workflow	N/A	[Will be available in Future Enhancement]

### Application Property

Details of configuring new application properties:

Column	Type	Values
Status	Drop-Down	<b>Active</b> Property is ACTIVE. <b>Pending</b> Property is Pending. Waiting for Approval <b>Disabled</b> Property has been disabled. <b>Deleted</b> Property is no longer in use, marked as Deleted.
Name	Text Box	[Enter the Name of Property]. SPACES NOT ALLOWED, MUST BE UNIQUE NAME WITHIN YOUR APPLICATION (NO DUPLICATE NAMES)
Description	Text Box	[Enter the Description of Property]
Type	Drop-Down	<b>SelectList</b> A dropdown list with a one-to-one relationship. <b>MultiList</b> A multi-select list with a one-to-many relationship. <b>RadioList</b> A radio list with a one-to-one relationship. <b>Textbox</b> A standard textbox. <b>Image</b> A standard image. <b>Virtual</b> A virtual property for use within another app property. <b>Dictionary</b> A collection of virtual app properties. <b>Password</b> A standard password field. <b>Label</b>

		A standard label
Data Type	Drop-Down	<b>String</b> <b>DateTime</b> <b>Currency</b> <b>Binary</b> <b>Image</b> <b>XML</b> <b>Int</b> <b>API_URL</b> <b>JSON_Dictionary</b> <b>None</b>
Display Type	Drop-Down	<b>Do Not Display</b> Do Not Display Anywhere <b>Create</b> Show on Create Page Only. <b>Edit</b> Show on Edit Page Only. <b>Delete</b> Show on Delete Page Only. <b>CreateEdit</b> Show on Both the Create and Edit Pages. <b>CreateEditDelete</b> Show on All the Create, Edit and Delete Pages. <b>Admin</b> Show on Admin Sections Only. <b>Reporting</b> Show on Reports Only. <b>Anywhere</b> Show everywhere.
Validation	Drop-Down	<b>None</b> No validation. <b>AlphaNumericOnly</b> This field only allows alphanumeric and will be validated as such. <b>Required</b> This required field will be validated on screen as such.
Encryption	Drop-Down	<b>None</b> No Encryption. <b>Password</b> Will employ hashing to password before storing in database. Hashing will prevent passwords from being recovered, only reset.
Return	Checkbox	If selected, it will return the property value during Authentication.
Viewable	Checkbox	If selected, it will allow authenticated users to <b>view</b> the property.
Editable	Checkbox	If selected, it will allow authenticated users to <b>edit</b> the property.

## User – Default Properties

Details of configuring application properties for each user. Listed below are the mandatory properties for each user.

Column	Type	Values
UserName	Text Box	[Enter the email address]
UserStatus	Drop-Down	<p><b>Active</b> User is ACTIVE.</p> <p><b>Pending</b> User is Pending. Waiting for Approval</p> <p><b>Disabled</b> User has been disabled.</p> <p><b>Deleted</b> User is no longer in use, marked as Deleted.</p> <p><b>Locked</b> User has been disabled.</p>
DateCreated	Text Box	Displays Today's date; date when the user was created. [auto populated]
LastActivityDate	Label	Displays the latest date of any activity performed by the user. [auto populated]
EmailAddress	Text Box	Displays the email address of the user

## Appendix B

### Status Codes and Response Headers

DSS SAF utilizes the standard HTTP status codes. All HTTP status codes that are utilized in SAF are listed below.

2xx – Success codes

4xx – Client side error

5xx – Server side error

Status Code	Response Headers	Values
200	<pre>{   "DSS-Security-Values" : } {   "DSS-Security-Redirect" : }</pre>	string
400	<pre>{   "DSS-Security-Errors" : }</pre>	string
500	<pre>{   "DSS-Security-Errors" : }</pre>	string

## Appendix C

### Definitions and Abbreviations

**API Key** – form of Bearer Token, it is a string that a client must present on every authentication and authorization call.

**Bearer Token** – A security token with the property that any party in possession of the token (a "bearer") can use the token in any way that any other party in possession of it can. <http://self-issued.info/docs/draft-ietf-oauth-v2-bearer.html>

**JSON (JavaScript Object Notation)** – Is an open standard format that uses human-readable text to transmit data objects consisting of attribute–value pairs. It is used primarily to transmit data between a server and web application, as an alternative to XML. <http://en.wikipedia.org/wiki/JSON>

**Single Pass:** – Single factor authentication. In the application property Multifactor Authentication is set to 'None'.

**Multi Pass:** – Multi-Factor authentication which is defined by a developer. It is configured on application properties window. More than two factor can be defined. Multifactor could include: 'RSA Token', 'Security Picture', etc.

## Appendix D

### Password Policy & Strong Password

Strong passwords must be used to ensure CDSS information assets are only accessed by authorized persons and to preserve the integrity of the systems by ensuring that no user can impersonate another on a system.

#### Protection of Passwords

1. All passwords must be protected from unauthorized discovery and use.
2. Passwords associated with personally assigned user IDs must not be shared with anyone.
3. Passwords must not be written down and stored on or near the system they are used to access.

#### Format

1. All passwords must contain a minimum of eight (8) characters.
2. All passwords must contain at least one (1) character from each of three (3) of the following four (4) password character classes:
  - Upper case letters – A B C
  - Lower case letters – a b c
  - Numerals – 1 2 3
  - Special characters – \$ & \* ! “ # @ ( ) { } . etc.
- ~~3. New passwords must be different than previous passwords by at least 3 characters. (Passwords are hashed when stored and cannot be un-hashed for this type of comparison)~~
4. See the Strong Password Guideline for more information on creating strong passwords.

#### Frequency

1. Passwords must be changed at least every 60 days.
2. Passwords must be changed whenever the password is suspected to have been discovered or compromised.
3. A password must not be re-used until at least 4 other unique passwords have been used.
  - a. Passwords can only be changed once every 7 days

#### Strong Password

The Strong Password Guideline provides suggestions for avoiding weak passwords and tips for creating a strong, complex password.

#### To create strong passwords, avoid the use of:

1. Words from a dictionary in any language.
2. Common misspellings of words.
3. Words spelled backward.
4. Abbreviations.
5. Repeated characters (111aaabbb) or sequences of characters (123456 or abcdef).
6. Characters that are adjacent to each other on the keyboard (qwerty or lkjhgf).
7. Personal information