ADSS CSP Developers Guide

ASCERTIA LTD

FEBRUARY 2023

Document Version – 8.0

TABLES

1 Introduction

1.1 Scope

This document provides information on how to integrate ADSS CSP Service in to your business application.

1.2 Intended Readership

This guide is intended for developers who are integrating business applications with ADSS CSP Service. The document assumes a reasonable knowledge of web application development, specifically RESTful Web services and ADSS Server.

1.3 Conventions

The following typographical conventions are used in this guide to help locate and identify information:

- **Bold** text identifies menu names, menu options, items you can click on the screen, file names, folder names, and keyboard keys.
- Courier New font identifies code and text that appears on the command line.
- **`Bold Courier New`** identifies commands that you are required to type in.
- `Courier New` font identifies Ajax request/response in HTTP message body.

1.4 Technical support

If Technical Support is required, Ascertia has a dedicated support team providing debugging assistance, integration assistance and general customer support. Ascertia Support can be accessed in the following ways:

Support Website	www.ascertia.com/support
Support Email	support@ascertia.com
Knowledge base	https://www.ascertia.com/products/knowledge-base/adss-server/

In addition to the free support service describe above, Ascertia provides formal support agreements with all product sales. Please contact sales@ascertia.com for more details.

A Product Support Questionnaire should be completed to provide Ascertia Support with further information about your system environment. When requesting help, it is always important to confirm:

- System Platform details.
- ADSS Server version number and build date.
- Details of specific issue and the relevant steps taken to reproduce it.
- Database version and patch level.
- Product log files

2 CSP Service Integration

ADSS CSP Service provides the capability to manage users and list their certificates. It provides the required API interfaces to manage users, certificates, handles signing requests & getting the signed hash (i.e. PKCS#1 signature) and their current statuses.

Business applications can integrate with ADSS CSP Service using its RESTful web services available to perform different operations.

The document is organised as follows:

Authentication Services:

Includes authentication interfaces like user/client authentication, Azure Active Directory authentication for Virtual CSP.

General Services:

Includes shared interfaces like CSP Service Status, password policy and CSP Service Information for business applications and Virtual CSP.

User Management API Interfaces:

Includes end-user interfaces that can be consumed by either business application or Virtual CSP.

Certificate Services:

Includes certificate interfaces that can be consumed by business application for managing user's certificates.

Signing Services:

Includes signature interfaces that can be consumed by using Virtual CSP which enables any Microsoft or third party CAPI/CNG applications to conduct remote signing seamlessly.

3 Authentication Services

It includes the following web services API:

- Authentication
- Single Sign On Authentication
- Revoke Tokens
- Authentication Settings

3.1 Authentication

CSP Service uses OAuth 2.0 to authorize client requests. The parameters required to authenticate a client application and retrieve an OAuth 2.0 access token are "grant type", “Client ID”, “Client Secret”, “User Name”, and “Password”.

The Authenticate API call returns an access token in the response which must be saved by the business client and must be provided with each subsequent API call as a bearer token for authorisation.

When invoking the refresh token option this will generate both a new access token for use, and refresh token for a subsequent refresh operation. The refresh token operation means the correct grant type must be used, i.e. refresh_token, and the client_id and client_secret parameters are required. User name and password are not required.

Exposed for: Business Applications and Virtual CSP

https://server:8779/adss/service/csp/authenticate		
HTTP Verb	POST	
Content-Type	application/x-www-form-urlencoded	
Accept	application/json	
authentication_profile_id	adss:authentication:profile:001	
Accept-Language	en-US	
Request Body	grant_type=password& client_id=samples_test_client& client_secret=jr67gj0h76gr83nf8734nj59g4he895jh87nr& username=admin@ascertia.com& password=Password@12	
Status Code	**Message**	**Response Body**

200	OK	{
		"access_token":"2YotnFZFEjr1zCsicMWpAA",
		"token_type":”Bearer”,
		"expires_in":3600,
		"refresh_token":"tGzv3JOkF0XG5Qx2TlKWIA"
		}

401	Unauthorized	{ "error": "60001", "error_description": "Detailed error message"

		}

500	Internal Server Error	{
		"error": "60001",
		"error_description": "Detailed error message"
		}

Item Details	
Name	Description
Request Parameters	
grant_type	Grant_type can be set to “password”, “client_credentials”, “refresh_token” or “active_directory”.
client_id	Client ID which is configured in ADSS CSP Console > Client Manager > CSP Service.
client_secret	Client Secret is generated and configured in ADSS Global Settings.
username	User ID of registered user in CSP Service.
password	Password the user/account will be defined in this parameter.
refresh_token(Optional)	If access token is expired then we get the new access token using refresh token.
authentication_profile_id	Get the client authentication profiles and pass the selected authentication profile ID.
Response Parameters	
access_token	OAuth user authentication access token - bearer token for subsequent authorisation to other API calls.
token_type	Type of the token returned by authorisation server. It always sets to "bearer".
expires_in	Number of seconds for which this access token is valid.
refresh_token	Refresh token returned by system to be used to regenerate access token.
error	The error code.
error_description	A description message that explains error code.

Table 1 – Authentication

3.2 Single Sign On Authentication

This API uses to authorize the client requests. Business application will call this API to get the access token of user by requesting third party, based on provided method and token. Currently Azure Active Directory is only supported.

Exposed for: Business Applications and Virtual CSP

https://server:8779/adss/service/csp/authenticate/sso		
HTTP Verb	POST	
Authorization	Bearer {user_access_token}	
Content-Type	application/json; charset=utf-8	
Accept	application/json	
authentication_profile_id	adss:authentication:profile:001	
Accept-Language	en-US	

Request Body	{
	"token":"2YotnFZFEjr1zCsicMWpAA",
	"method":"AZURE_ACTIVE_DIRECTORY"
	}	

Status Code	**Message**	**Response Body**

200	OK	{
		"access_token":"2YotnFZFEjr1zCsicMWpAA",
		"token_type":"Bearer",
		"expires_in":3600,
		"refresh_token":"tGzv3JOkF0XG5Qx2TlKWIA"
		}

400	Bad Request	{
		"error": "60001",
		"error_description": "Detailed error message"
		}

401	Unauthorized	{
		"error": "60001",
		"error_description": "Detailed error message"
		}

500	Internal Server Error	{
		"error": "60001",
		"error_description": "Detailed error message"
		}

Item Details	
Name	Description
Request Parameters	

authentication_profile_id	Authentication method will selected on this profile otherwise, default profile method will be selected.
token	(Mandatory) Token obtained from third party e.g Azure Active Directory.
method	(Mandatory) Supported method type e.g AZURE_ACTIVE_DIRECTORY.
Response Parameters	
access_token	OAuth user authentication access token - bearer token for subsequent authorisation to other API calls.
token_type	Type of the token returned by authorisation server. It always sets to "bearer".
expires_in	Number of seconds for which this access token is valid.
refresh_token	Refresh token returned by system to be used to regenerate access token.
error	The error code.
error_description	A description message that explains error code.

Table 2 – Single Sign On Authentication

3.3 Revoke Tokens

Business application can call this API to revoke the refresh tokens generated for a user.

Exposed for: Business Applications and Virtual CSP

https://server:8779/adss/service/csp/tokens		
HTTP Verb	POST	
Authorization	Bearer {client_access_token}	
Content-Type	application/x-www-form-urlencoded	
Accept	application/json	
Accept-Language	en-US	
Request Body	refresh_token= jr67gj0h76gr83nf8734nj59g4he895jh87nr…& token_type=refresh_token	
Status Code	**Message**	**Response Body**
200	OK	

400	Bad Request	<pre>{
		"error": "60001",
		"error_description": "Detailed error message"
		}</pre>

401	Unauthorized	<pre>{
		"error": "60001",
		"error_description": "Detailed error message"
		}</pre>

500	Internal Server Error	{
		"error": "60001",
		"error_description": "Detailed error message"
		}

Item Details	
Name	Description
Request Parameters	
access_token	(Mandatory) OAuth access token obtained as a result of successful authentication via "client_credentials" grant type.
refresh_token	(Mandatory) Refresh token which has to be revoked.
token_type	(Mandatory) Type of the token e.g refresh_token.
Response Parameters	
error	The error code.
error_description	A description message that explains error code.

Table 3 – Revoke Token

3.4 Authentication Profiles

This API authenticates a client with access token and returns the supported authentication methods to client application i.e VCSP.

Exposed for: Virtual CSP

https://server:8779/adss/service/csp/v2/profiles/auth		
HTTP Verb	GET	
Authorization	Bearer {client_access_token}	
Accept	application/json	
Accept-Language	en-US	
App-Name	(Optional) Client Id of requester via CSP proxy	
Status Code	**Message**	**Response Body**

200	OK	{
		"default_authentication_profile": {
		“profile_id”: "adss:authentication:profile:001"
		“profile_name”: "Azure AD"
		“auth_type”: "AZURE_ACTIVE_DIRECTORY",
		"auth_info": {
		"app_id": "osdcnosdcn49309nn",
		"auth_url": "https://graph.microsoft.com",
		}

		"allowed_authentication_profile": [
		{
		“profile_id”: "adss:authentication:profile:001"
		“profile_name”: "Azure AD"
		“auth_type”: "AZURE_ACTIVE_DIRECTORY",
		"auth_info": {
		"app_id": "osdcnosdcn49309nn",
		"auth_url": "https://graph.microsoft.com",
		}
		},
		{
		“profile_id”: "adss:authentication:profile:002"
		“profile_name”: "Microsoft Active Directory"
		“auth_type”: "ACTIVE_DIRECTORY ",
		},
		{
		“profile_id”: "adss:authentication:profile:003"
		“profile_name”: "Web RA"
		“auth_type”: "PASSWORD ",
		},
		]
		}

400	Bad Request	{ "error": "60001", "error_description": "Detailed error message" }
401	Unauthorized	{ "error": "60001", "error_description": "Detailed error message" }
404	Not Found	{ "error": "60001", "error_description": "Detailed error message" }
500	Internal Server Error	{ "error": "60001", "error_description": "Detailed error message" }

Item Details	
Name	Description
Request Parameters	
access_token	(Mandatory) OAuth access token obtained as a result of successful authentication via "client_credentials" grant type.
Response Parameters	
default_authentication	Default authentication which is selected in default authentication profile.

allowed_authentication	Allowed authentication profiles selected by client.
auth_type	Authentication type of the user in CSP service i.e. AZURE_AD.
auth_info	User name as friendly name of the user in CSP service.
error	The error code.
error_description	A description message that explains error code.

Table 4 - Authenticate User

4 General Services

It includes the following web services API:

- Status Service
- About Service
- Password Policy
- Advance Settings

4.1 Status Service

This API returns the status of CSP Service i.e. running or not.

Exposed for: Business Applications and Virtual CSP

https://server:8779/adss/service/csp?client_id=abc&profile_id=xyz		
HTTP Verb	GET	
Accept	application/json	
Accept-Language	en-US	
Status Code	**Message**	**Response Body**
200	OK	{ "message": "Success", "message_description": "ADSS CSP Service is running" }
401	Unauthorized	{ "error": "60001", "error__escription": "Detailed error message" }
500	Internal Server Error	{ "error": "60001", "error_description": "Detailed error message" }

Item Details	
Name	Description
Request Parameters	
client_id	(Optional) Client ID which is configured in ADSS CSP Console > Client Manager
profile_id	(Optional) Profile ID which is configured in ADSS CSP Console > Client Manager > CSP Service
Response Parameters	
message	If CSP Service is running, it returns Success otherwise error response is returned.

message_description	Detailed information about CSP Service status.
error	The error code.
error_description	A description message that explains error code.

Table 5 – Status Service

4.2 About Service

This API returns the information about ADSS CSP Service.

Exposed for: Business Applications and Virtual CSP

https://server:8779/adss/service/csp/v2/about		
HTTP Verb	GET	
Authorization	Bearer {client_access_token}	
Accept	application/json	
Accept-Language	en-US	
Status Code	**Message**	**Response Body**
200	OK	{ "version": "Version", "build_number": "Build Number" }
400	Bad Request	{ "error": "60001", "error_description": "Detailed error message" }
401	Unauthorized	{ "error": "60001", "error_description": "Detailed error message" }
500	Internal Server Error	{ "error": "60001", "error_description": "Detailed error message" }

Item Details	
Name	Description
Request Parameters	
access_token	(Mandatory) OAuth access token obtained as a result of successful authentication via "client_credentials" grant type.

Response Parameters	
version	ADSS CSP Service version number.
build_number	ADSS CSP Service build number.
error	The error code.
error_description	A description message that explains error code.

Table 6 – About Service

4.3 Password Policy

This API is used to get password policy information from ADSS CSP Service for user password.

Exposed for: Business Applications and Virtual CSP

https://server:8779/adss/service/csp/v2/users/password/policy		
HTTP Verb	POST	
Authorization	Bearer {client_access_token}	
Content-Type	application/json; charset=utf-8	
Accept	application/json	
Accept-Language	en-US	

Request Body	{
	"profile_id": "ADSS CSP Profile"
	}	

Status Code	**Message**	**Response Body**

200	OK	{
		"min_pass_length": "8",
		"max_pass_length": "255",
		"upper_lower_letters": "true",
		"digits_and_special_char": "false"
		}

400	Bad Request	{
		"error": "60001",
		"error_description": "Detailed error message"
		}

401	Unauthorized	{
		"error": "60001",
		"error_description": "Detailed error message"
		}

500	Internal Server Error	{ "error": "60001", "error_description": "Detailed error message"

		}

Item Details	
Name	Description
Request Parameters	
access_token	(Mandatory) OAuth access token obtained as a result of successful authentication via "client_credentials" grant type.
profile_id	(Mandatory) Profile ID which is configured in ADSS CSP Console > Client Manager > CSP Service
Response Parameters	
min_pass_length	Minimum password length (number of characters)
max_pass_length	Maximum password length (number of characters). Supported value is 255.
upper_lowe_letters	If true, password must contain upper and lower case letters.
Digits_and_special_char	If true, password must contain digits and special characters. Supported special characters are: !"#$%&'()*+,-./:;<=>?@[\]^_`{|}~
error	The error code.
error_description	A description message that explains error code.

Table 7 – Password Policy

4.4 Advance Settings

This API is used to get CSP Advance Settings configured in ADSS Global Settings.

Exposed for: Virtual CSP

https://server:8779/adss/service/csp/v2/advancesettings		
HTTP Verb	GET	
Authorization	Bearer {user_access_token}	
Accept	application/json	
App-Name	(Optional) Client Id of requester via CSP proxy	
Accept-Language	en-US	
Status Code	**Message**	**Response Body**
200	OK	{ "otp_expiry_time": 100,

		<pre>"otp_length": 6,
		"user_auth_retrie_limit": 5,
		"user_block_period": 100,
		}</pre>

400	Bad Request	<pre>{
		"error": "60001",
		"error_description": "Detailed error message"
		}</pre>

401	Unauthorized	<pre>{
		"error": "60001",
		"error_description": "Detailed error message"
		}</pre>

412	Pre-Condition Failed	<pre>{
		"error": "60001",
		"error_description": "Detailed error message"
		}</pre>

404	Not Found	<pre>{
		"error": "60001",
		"error_description": "Detailed error message"
		}</pre>

500	Internal Server Error	<pre>{
		"error": "60001",
		"error_description": "Detailed error message"
		}</pre>

Item Details	
Name	Description
Request Parameters	
access_token	(Mandatory) OAuth access token obtained as a result of successful user authentication.
Response Parameters	
otp_expiry_time	OTP expiry time is integer value configured in seconds.
otp_length	OTP length is integer value so that business application can show the number of digits.
user_auth_retries_limit	User retires limit while authenticating the user.
user_block_period	User block period in seconds when user is blocked for a specific time.

Table 8 – Advance Settings

5 User Services

User services includes the following web services APIs:

- Register User
- Update User
- User Profile
- Delete User
- Count Users
- Get Users
- Change Password
- Reset Password
- Resend Password Reset
- Confirm Reset Password
- Change Email
- Resend Change Email
- Confirm Change Email

5.1 Register User

This API is used to register a user in CSP Service.

Exposed for: Business Applications

https://server:8779/adss/service/csp/v2/users/register		
HTTP Verb	POST	
Authorization	Bearer {client_access_token}	
Content-Type	application/json; charset=utf-8	
Accept	application/json	
Accept-Language	en-US	

Request Body	{
	"user_id": "johnDoe12",
	"user_name": "John Doe",
	"user_password": "password",
	"email": "john.doe@ascertia.com",
	"profile_id": "CSP-RegisterUser"
	}	

Status Code	**Message**	**Response Body**
201	OK	Created

400	Bad Request	{
		"error": "60001",
		"error_description": "Detailed error message"
		}

401	Unauthorized	{
		"error": "86000",
		"error_description": "Detailed error message"
		}

500	Internal Server Error	{
		"error": "60001",
		"error_description": "Detailed error message"
		}

Item Details	
Name	Description
Request Parameters	
access_token	(Mandatory) OAuth access token obtained as a result of successful authentication via "client_credentials" grant type.
user_id	(Mandatory) User ID identifying the user in CSP service
user_name	(Optional) User name as friendly name for the user in CSP service Following languages are supported for username: • Norwegian Characters • Slovenian Characters • Czech & Slovak Characters • Icelandic Characters • Arabic Characters • Latvian Characters
user_password	(Optional) User password in CSP service.
email	(Mandatory) User Email Address. It will be used to send OTP for password recovery etc.
profile_id	CSP Profile ID against which user will be registered in CSP Service
Response Parameters	
error	The error code
error_description	A description message that explains error code.

Table 9 - Register User

5.2 Update User

This API is used to update the user information in ADSS CSP Service.

Exposed for: Business Applications

https://server:8779/adss/service/csp/v2/users/{user_id}		
HTTP Verb	PUT	
Authorization	Bearer {client_access_token}	
Content-Type	application/json; charset=utf-8	
Accept	application/json	
Accept-Language	en-US	

Request Body	{
	"user_name": "John Doe",
	"status": "ACTIVE"
	}	

Status Code	**Message**	**Response Body**
200	OK	

400	Bad Request	{
		"error": "60001",
		"error_description": "Detailed error message"
		}

401	Unauthorized	{
		"error": "60001",
		"error_description": "Detailed error message"
		}

500	Internal Server Error	{
		"error": "60001",
		"error_description": "Detailed error message"
		}

Item Details	
Name	Description
Request Parameters	
access_token	(Mandatory) OAuth access token obtained as a result of successful authentication via "client_credentials" grant type.
user_name	(Optional) User friendly name for already registered user in CSP service.
status	Change the registered user status. Possible values are: ACTIVE, INACTIVE and BLOCKED.
Response Parameters	
error	The error code
error_description	A description message that explains error code.

Table 10 - Update User

5.3 Get User

This API is used to get registered user information in CSP Service identified by {user_id}.

Exposed for: Business Applications

https://server:8779/adss/service/csp/v2/users/{user_id}		
HTTP Verb	GET	
Authorization	Bearer {client_access_token}	
Accept	application/json	
Accept-Language	en-US	
Status Code	**Message**	**Response Body**

200	OK	{
		"user_name": "John Doe",
		"user_id": "johnDoe12",
		"email": "john.doe@ascertia.com",
		"status": "ACTIVE",
		"profile_id": "CSP-Profile-User"
		}

400	Bad Request	{
		"error": "60001",
		"error_description": "Detailed error message"
		}

401	Unauthorized	{
		"error": "60001",
		"error_description": "Detailed error message"
		}

404	Not Found	{
		"error": "60001",
		"error_description": "Detailed error message"
		}

500	Internal Server Error	{
		"error": "60001",
		"error_description": "Detailed error message"
		}

Item Details	
Name	Description
Request Parameters	

access_token	(Mandatory) OAuth access token obtained as a result of successful authentication via "client_credentials" grant type.
user_id	(Mandatory) User ID identifying the user in CSP service.
Response Parameters	
user_id	User ID identifying the user in CSP service.
user_name	User name as friendly name for the user in CSP service.
email	Email of the user. It will be used to send OTP for password recovery etc.
profile_id	CSP Profile ID against which the user was registered in CSP Service.
status	User status e.g. ACTIVE, INACTIVE, BLOCKED
error	The error code.
error_description	A description message that explains error code.

Table 11 - User Profile

5.4 Delete User

This API is used to delete a user in CSP Service identified by {user_id}.

Exposed for: Business Applications

https://server:8779/adss/service/csp/v2/users/{user_id}		
HTTP Verb	DELETE	
Authorization	Bearer {client_access_token}	
Accept	application/json	
Accept-Language	en-US	
Status Code	**Message**	**Response Body**
200	OK	
400	Bad Request	{ "error": "60001", "error_description": "Detailed error message" }
401	Unauthorized	{ "error": "60001", "error_description": "Detailed error message" }
404	Not Found	{ "error": "60001", "error_description": "Detailed error message" }
500	Internal Server Error	{ "error": "60001", "error_description": "Detailed error message"

		}

Item Details	
Name	Description
Request Parameters	
access_token	(Mandatory) OAuth access token obtained as a result of successful authentication via "client_credentials" grant type.
user_id	(Mandatory) User ID identifying the user in CSP service.
Response Parameters	
error	The error code.
error_description	A description message that explains error code.

Table 12 - Delete User

5.5 User Count

This API is used to get the users count registered in CSP Service by a Client.

Exposed for: Business Applications

https://server:8779/adss/service/csp/v2/users/count		
HTTP Verb	GET	
Authorization	Bearer {client_access_token}	
Accept	application/json	
Accept-Language	en-US	
Status Code	**Message**	**Response Body**
200	OK	{ "user_count": "99", }
400	Bad Request	{ "error": "60001", "error_description": "Detailed error message" }
401	Unauthorized	{ "error": "60001", "error_description": "Detailed error message" }
404	Not Found	{ "error": "60001", "error_description": "Detailed error message" }

500	Internal Server Error	{
		"error": "60001",
		"error_description": "Detailed error message"
		}

Item Details	
Name	Description
Request Parameters	
access_token	(Mandatory) OAuth access token obtained as a result of successful authentication via "client_credentials" grant type.
Response Parameters	
user_count	Number of users registered with the specified {clientId}.
error	The error code.
error_description	A description message that explains error code.

Table 13 – Count Users

5.6 Get Users

This API is used to get the list of registered users in CSP Service against specific client. Records can be divided based on the value or startPointer and fetchSize parameters. If these parameters (startPointer & fetchSize) are not provided, then all the users for specified client are returned.

Exposed for: Business Applications

https://server:8779/adss/service/csp/v2/users/{start_pointer}/{fetch_size}		
HTTP Verb	GET	
Authorization	Bearer {client_access_token}	
Accept	application/json	
Accept-Language	en-US	
Response Headers		
x-total-records	2	
Status Code	**Message**	**Response Body**

200	OK	{
		"users": [{
		"user_name": "Peter Doe",
		"user_id": "PeterDoe12",
		"email": "peter.doe@ascertia.com",
		"status": "ACTIVE",
		"profile_id": "profile-001"
		},{
		"user_name": "John Doe",
		"user_id": "johnDoe12",
		"user_email": "john.doe@ascertia.com",

		<pre>"status": "ACTIVE",
		"profile_id": "profile-001"
		}]
		}</pre>

400	Bad Request	<pre>{
		"error": "60001",
		"error_description": "Detailed error message"
		}</pre>

401	Unauthorized	<pre>{
		"error": "60001",
		"error_description": "Detailed error message"
		}</pre>

500	Internal Server Error	<pre>{
		"error": "60001",
		"error_description": "Detailed error message"
		}</pre>

Item Details	
Name	Description
Request Parameters	
access_token	(Mandatory) OAuth access token obtained as a result of successful authentication via "client_credentials" grant type.
start_pointer	Start index of the user records.
fetch_size	Number of records to be returned in response.
Response Parameters	
users	Array of users containing user information.
user_id	User ID identifying the user in CSP service.
user_name	User name as friendly name of the user in CSP service.
email	Email for the user in CSP service. It will be used to send OTP for password recovery etc.
profile_id	CSP Profile ID against which user is registered in CSP Service.
status	User status e.g. ACTIVE, INACTIVE, BLOCKED
error	The error code.
error_description	A description message that explains error code.

Table 14 - Get Users

5.7 Change Password

This API is used to change the password of a registered user. The user provides the old password and new password in the request. The CSP Service verifies the old password and after successful verification, it will change the old password with the new one.

Exposed for: Virtual CSP

https://server:8779/adss/service/csp/v2/users/password/change		
HTTP Verb	POST	
Authorization	Bearer {user_access_token}	
Content-Type	application/json; charset=utf-8	
Accept	application/json	
Accept-Language	en-US	
App-Name	(Optional) Client Id of requester via CSP proxy	

Request Body	{
	"old_password": "password"
	"new_password": "password"
	}	

Status Code	**Message**	**Response Body**
200	OK	

400	Bad Request	{
		"error": "60001",
		"error_description": "Detailed error message"
		}

401	Unauthorized	{
		"error": "60001",
		"error_description": "Detailed error message"
		}

404	Not Found	{
		"error": "60001",
		"error_description": "Detailed error message"
		}

500	Internal Server Error	{
		"error": "60001",
		"error_description": "Detailed error message"
		}

Item Details	
Name	Description
Request Parameters	
access_token	(Mandatory) OAuth access token obtained as a result of successful user authentication.
old_password	(Mandatory) Current password of the user.

new_password	(Mandatory) New user password set for the user.
Response Parameters	
error	The error code.
error_description	A description message that explains error code.

Table 15 - Change Password

5.8 Reset Password (No Auth)

If a user forgets his/her password, this API is used to recover/reset password. Client application sends the user ID with user password using client authentication and user password will be updated.

Exposed for: Virtual CSP

https://server:8779/adss/service/csp/v2/users/password/reset/noauth		
HTTP Verb	POST	
Authorization	Bearer {client_access_token}	
Content-Type	application/json; charset=utf-8	
Accept	application/json	
Accept-Language	en-US	
App-Name	(Optional) Client Id of requester via CSP proxy	

Request Body	{
	"user_id": "john.doe",
	"user_password": "password"
	}	

Status Code	**Message**	**Response Body**
200	OK	

400	Bad Request	{
		"error": "60001",
		"error_description": "Detailed error message"
		}

401	Unauthorized	{
		"error": "60001",
		"error_description": "Detailed error message"
		}

404	Not Found	{
		"error": "60001",
		"error_description": "Detailed error message"
		}

500	Internal Server Error	{
		"error": "60001",
		"error_description": "Detailed error message"
		}

Item Details	
Name	Description
Request Parameters	
access_token	(Mandatory) OAuth access token obtained as a result of successful authentication via "client_credentials" grant type.
user_id	(Mandatory) User ID identifying the user in CSP service.
user_password	(Mandatory) User password in CSP service.
Response Parameters	
error	The error code.
error_description	A description message that explains error code.

Table 16 – Reset Password(NoAuth)

5.9 Reset Password

If a user forgets his/her password, this API is used to recover/reset password. Password recovery is two steps process; first the business application calls this interface, CSP Service sends OTP to user email address and then call Confirm Reset Password interface to provide new password for the user.

Exposed for: Virtual CSP

https://server:8779/adss/service/csp/v2/users/password/reset		
HTTP Verb	POST	
Authorization	Bearer {client_access_token}	
Content-Type	application/json; charset=utf-8	
Accept	application/json	
Accept-Language	en-US	
App-Name	(Optional) Client Id of requester via CSP proxy	

Request Body	{
	"user_id": "john.doe "
	}	

Status Code	**Message**	**Response Body**
200	OK	

400	Bad Request	{
		"error": "60001",
		"error_description": "Detailed error message"
		}

401	Unauthorized	{ "error": "60001", "error_description": "Detailed error message"

		}

404	Not Found	{
		"error": "60001",
		"error_description": "Detailed error message"
		}

500	Internal Server Error	{
		"error": "60001",
		"error_description": "Detailed error message"
		}

Item Details	
Name	Description
Request Parameters	
access_token	(Mandatory) OAuth access token obtained as a result of successful authentication via "client_credentials" grant type.
user_id	(Mandatory) User ID identifying the user in CSP service.
Response Parameters	
error	The error code.
error_description	A description message that explains error code.

Table 17 – Reset Password

5.10 Confirm Reset Password

This API verifies the provided OTP which was returned in Reset Password interface. Once verified it sets the new password for the user.

Exposed for: Virtual CSP

https://server:8779/adss/service/csp/v2/users/password/reset/confirm		
HTTP Verb	POST	
Authorization	Bearer {client_access_token}	
Content-Type	application/json	
Accept	application/json	
Accept-Language	en-US	
App-Name	(Optional) Client Id of requester via CSP proxy	

Request Body	{
	"user_id": "john.doe",
	"new_password": "password",
	"otp": "46548"
	}	

Status Code	**Message**	**Response Body**

200	OK	

400	Bad Request	{
		"error": "60001",
		"error_description": "Detailed error message"
		}

401	Unauthorized	{
		"error": "60001",
		"error_description": "Detailed error message"
		}

404	Not Found	{
		"error": "60001",
		"error_description": "Detailed error message"
		}

500	Internal Server Error	{
		"error": "60001",
		"error_description": "Detailed error message"
		}

Item Details	
Name	Description
Request Parameters	
access_token	(Mandatory) OAuth access token obtained as a result of successful authentication via "client_credentials" grant type.
user_id	(Mandatory) User ID identifying the user in CSP service.
new_password	(Mandatory) New password for the user.
otp	(Mandatory) OTP value to validate request.
Response Parameters	
error	The error code.
error_description	A description message that explains error code.

Table 18 – Resend Password Recovery

5.11 Change Email

When user wants to change his/her email, this API is used. Email change is two steps process; first the business application calls this interface, CSP Service sends OTPs to user's new and old email addresses.

The user provides these OTPs in Confirm Change Email API.

Exposed for: Virtual CSP

https://server:8779/adss/service/csp/v2/users/email/change	
HTTP Verb	POST
Authorization	Bearer {user_access_token}

Content-Type	application/json; charset=utf-8	
Accept	application/json	
Accept-Language	en-US	
App-Name	(Optional) Client Id of requester via CSP proxy.	

Request Body	<pre>{
	"old_email": "jhon@ascertia.com"
	"new_email": "john.doe2@ascertia.com"
	}</pre>	

Status Code	**Message**	**Response Body**
200	OK	

400	Bad Request	<pre>{
		"error": "60001",
		"error_description": "Detailed error message"
		}</pre>

401	Unauthorized	<pre>{
		"error": "60001",
		"error_description": "Detailed error message"
		}</pre>

404	Not Found	<pre>{
		"error": "60001",
		"error_description": "Detailed error message"
		}</pre>

500	Internal Server Error	<pre>{
		"error": "60001",
		"error_description": "Detailed error message"
		}</pre>

Item Details	
Name	Description
Request Parameters	
access_token	(Mandatory) OAuth access token obtained as a result of successful user authentication.
old_email	(Mandatory) User old email address.
new_email	(Mandatory) User new email address.
Response Parameters	
error	The error code.
error_description	A description message that explains error code.

Table 19 – Change Email

5.12 Resend Change Email

This API is used to resend OTPs to user's old and new email addresses.

Exposed for: Virtual CSP

https://server:8779/adss/service/csp/v2/users/email/change/resend		
HTTP Verb	GET	
Authorization	Bearer {user_access_token}	
Accept	application/json	
Accept-Language	en-US	
App-Name	(Optional) Client Id of requester via CSP proxy.	
Status Code	**Message**	**Response Body**
200	OK	
400	Bad Request	{ "error": "60001", "error_description": "Detailed error message" }
401	Unauthorized	{ "error": "60001", "error_description": "Detailed error message" }
404	Not Found	{ "error": "60001", "error_description": "Detailed error message" }
500	Internal Server Error	{ "error": "60001", "error_description": "Detailed error message" }

Item Details	
Name	Description
Request Parameters	
access_token	(Mandatory) OAuth access token obtained as a result of successful user authentication.
Response Parameters	
error	The error code.
error_description	A description message that explains error code.

Table 20 – Resend Email Change

5.13 Confirm Change Email

This API is used to change the user email after verifying the provided OTPs.

Exposed for: Virtual CSP

https://server:8779/adss/service/csp/v2/users/email/change/confirm		
HTTP Verb	POST	
Authorization	Bearer {user_access_token}	
Content-Type	application/json; charset=utf-8	
Accept	application/json	
Accept-Language	en-US	
App-Name	(Optional) Client Id of requester via CSP proxy.	

Request Body	{
	"otp": "4565468",
	"new_email_otp": "4565468"
	}	

Status Code	**Message**	**Response Body**
200	OK	

400	Bad Request	{
		"error": "60001",
		"error_description": "Detailed error message"
		}

401	Unauthorized	{
		"error": "60001",
		"error_description": "Detailed error message"
		}

404	Not Found	{
		"error": "60001",
		"error_description": "Detailed error message"
		}

500	Internal Server Error	{
		"error": "60001",
		"error_description": "Detailed error message"
		}

Item Details	
Name	Description
Request Parameters	
access_token	(Mandatory) OAuth access token obtained as a result of successful user authentication.
otp	(Mandatory) User OTP sent to old email address.

new_email_otp	(Mandatory) User OTP sent to new email address.
Response Parameters	
error	The error code
error_description	A description message that explains error code.

Table 21 – Confirm Change Email

6 Certificate Services

Certificate services includes the following APIs:

- Add User Certificates
- Delete User Certificates
- Update User Certificate
- Update User Certificates
- List User Certificates
- Get User Certificates

6.1 Add User Certificates

This API is used to add the certificate(s) for a user which are used for signing via Virtual CSP.

Exposed for: Business Applications

https://server:8779/adss/service/csp/v2/users/certs/{user_id}		
HTTP Verb	POST	
Authorization	Bearer {client_access_token}	
Content-Type	application/json; charset=utf-8	
Accept	application/json	
App-Name	(Optional) Client Id of requester via CSP proxy	
Accept-Language	en-US	

Request Body	{
	"certificates": [{
	"cert_alias": "o5eaqqdi9hte17vj6s63ai4j4chlpqt",
	"sam_user_id": "john.doe@ascertia.com",
	"auth_pass": "TRUE",
	"certificate": "MIAGCSqGSI[...]AAAAAAAAA="
	"certificate_chain": "MIAGCSqGSI[...]AAAAAAAAA="
	},{
	"cert_alias": "o5eaqqdi9hsdf63ai4j4chlpqt",
	"sam_user_id": "john.doe@ascertia.com",
	"auth_pass": "FALSE",
	"certificate": "MIAGCSqGSI[...]AAAAAAAAA="
	}]
	}	

Status Code	**Message**	**Response Body**
200	OK	

400	Bad Request	{
		"error": "60001",
		"error_description": "Detailed error message"
		}

401	Unauthorized	{
		"error": "60001",
		"error_description": "Detailed error message"
		}

404	Not Found	{
		"error": "60001",
		"error_description": "Detailed error message"
		}

500	Internal Server Error	{
		"error": "60001",
		"error_description": "Detailed error message"
		}

Item Details	
Name	Description
Request Parameters	
access_token	Mandatory) OAuth access token obtained as a result of successful authentication via "client_credentials" grant type.
certificates	Array of user certificates.
cert_alias	(Mandatory) Certificate Alias of the certificate.
sam_user_id	(Optional) Specifies SAM User ID when this certificate is to be used for authorise remote signing.
auth_pass	(Mandatory) Specifies password based authentication is applied on this certificate.
certificate	(Mandatory) Certificate Base64 String.
certificate_chain	(Optional) Certificate chain Base64 String.
Response Parameters	
error	The error code.
error_description	A description message that explains error code.

Table 22 –Add User Certificates

6.2 Delete User Certificate

This API deletes the certificates for specified user with certificate alias.

Exposed for: Business Applications

https://server:8779/adss/service/csp/v2/users/certs/{user_id}/{cert_alias}		
HTTP Verb	DELETE	
Authorization	Bearer {client_access_token}	
Accept	application/json	
App-Name	(Optional) Client Id of requester via CSP proxy	
Accept-Language	en-US	
Status Code	**Message**	**Response Body**
200	OK	

400	Bad Request	{
		"error": "60001",
		"error_description": "Detailed error message"
		}

401	Unauthorized	{
		"error": "60001",
		"error_description": "Detailed error message"
		}

404	Not Found	{
		"error": "60001",
		"error_description": "Detailed error message"
		}

500	Internal Server Error	{
		"error": "60001",
		"error_description": "Detailed error message"
		}

Item Details	
Name	Description
Request Parameters	
access_token	Mandatory) OAuth access token obtained as a result of successful authentication via "client_credentials" grant type.
Response Parameters	
error	The error code.
error_description	A description message that explains error code.

Table 23 – Delete User Certificates

6.3 Update User Certificate

This API is used to update the specified certificate for a user used for signing via VCSP.

Exposed for: Business Applications

https://server:8779/adss/service/csp/v2/users/certs/{user_id}	
HTTP Verb	PUT
Authorization	Bearer {client_access_token}
Content-Type	application/json; charset=utf-8
Accept	application/json
App-Name	(Optional) Client Id of requester via CSP proxy

Accept-Language	en-US	

Request Body	<pre>{
	"cert_alias": "o5eaqqdi9hsdf63ai4j4chlpqt",
	"sam_user_id": "john.doe@ascertia.com",
	"auth_pass": "FALSE",
	"certificate": "MIAGCSqGSI[...]AAAAAAAAA="
	"certificate_chain": "MIAGCSqGSI[...]AAAAAAAAA="
	}</pre>	

Status Code	**Message**	**Response Body**
200	OK	

400	Bad Request	<pre>{
		"error": "60001",
		"error_description": "Detailed error message"
		}</pre>

401	Unauthorized	<pre>{
		"error": "60001",
		"error_description": "Detailed error message"
		}</pre>

404	Not Found	<pre>{
		"error": "60001",
		"error_description": "Detailed error message"
		}</pre>

500	Internal Server Error	<pre>{
		"error": "60001",
		"error_description": "Detailed error message"
		}</pre>

Item Details	
Name	Description
Request Parameters	
access_token	Mandatory) OAuth access token obtained as a result of successful authentication via "client_credentials" grant type.
user_id	(Mandatory) User ID identifying the user in CSP service.
cert_alias	(Mandatory) Certificate Alias of the certificate.
sam_user_id	(Optional) Specifies SAM User ID when this certificate is to be used for authorise remote signing.
auth_pass	(Mandatory) Specifies password based authentication is applied on this certificate.
certificate	(Mandatory) Certificate Base64 String.
certificate_chain	(Optional) Certificate chain Base64 String.
Response Parameters	
error	The error code.
error_description	A description message that explains error code.

Table 24 – Update User Certificate

6.4 Update User Certificates

This API is used to set the certificate(s) for a user which are used for signing via Virtual CSP.

Exposed for: Business Applications

https://server:8779/adss/service/csp/v2/users/certs/update		
HTTP Verb	POST	
Authorization	Bearer {client_access_token}	
Content-Type	application/json; charset=utf-8	
Accept	application/json	
App-Name	(Optional) Client Id of requester via CSP proxy	
Accept-Language	en-US	

Request Body	{
	"user_id": "johnDoe12",
	"certificates": [{
	"cert_alias": "o5eaqqdi9hte17vj6s63ai4j4chlpqt",
	"sam_user_id": "john.doe@ascertia.com",
	"auth_pass": "TRUE",
	"certificate": "MIAGCSqGSI[...]AAAAAAAAA="
	"certificate_chain": "MIAGCSqGSI[...]AAAAAAAAA="
	},{
	"cert_alias": "o5eaqqdi9hsdf63ai4j4chlpqt",
	"sam_user_id": "john.doe@ascertia.com",
	"auth_pass": "FALSE",
	"certificate": "MIAGCSqGSI[...]AAAAAAAAA="
	"certificate_chain": "MIAGCSqGSI[...]AAAAAAAAA="
	}]
	}	

Status Code	**Message**	**Response Body**
200	OK	

400	Bad Request	{
		"error": "60001",
		"error_description": "Detailed error message"
		}

401	Unauthorized	{
		"error": "60001",
		"error_description": "Detailed error message"
		}

404	Not Found	{
		"error": "60001",
		"error_description": "Detailed error message"
		}

500	Internal Server Error	{ "error": "60001", "error_description": "Detailed error message"

		}

Item Details	
Name	Description
Request Parameters	
access_token	(Mandatory) OAuth access token obtained as a result of successful authentication via "client_credentials" grant type.
user_id	(Mandatory) User ID identifying the user in CSP service.
certificates	Array of user certificates.
cert_alias	(Mandatory) Certificate Alias of the certificate.
sam_user_id	(Optional) Specifies SAM User ID when this certificate is to be used for authorise remote signing.
auth_pass	(Mandatory) Specifies password based authentication is applied on this certificate.
certificate	(Mandatory) Certificate Base64 String.
certificate_chain	(Optional) Certificate chain Base64 String.
Response Parameters	
error	The error code.
error_description	A description message that explains error code.

Table 25 – Update User Certificates

6.5 List User Certificates

This API returns the list of certificates for specified user.

Exposed for: Virtual CSP

https://server:8779/adss/service/csp/v2/users/certs		
HTTP Verb	GET	
Authorization	Bearer {user_access_token}	
Accept	application/json	
App-Name	(Optional) Client Id of requester via CSP proxy.	
Accept-Language	en-US	
Status Code	**Message**	**Response Body**

200	OK	{
		"certificates": [{
		"cert_alias": "o5eaqqdiai4j4chlpqt",
		"sam_user_id": "john.doe@ascertia.com",
		"auth_pass": "TRUE",

		"certificate": "MIAGCSqGSI[...]AAAAA="
		"certificate_chain": "MIAGsdGSI[...]AA="
		},{
		"cert_alias": "o5eaqadhlpqt",
		"sam_user_id": "john.doe@ascertia.com",
		"auth_pass": "FALSE",
		"certificate": "MIAGCSqGSI[...]AAA="
		"certificate_chain": "MIAGsdGSI[...]AA="
		}]
		}

400	Bad Request	{
		"error": "60001",
		"error_description": "Detailed error message"
		}

401	Unauthorized	{
		"error": "60001",
		"error_description": "Detailed error message"
		}

404	Not Found	{
		"error": "60001",
		"error_description": "Detailed error message"
		}

500	Internal Server Error	{
		"error": "60001",
		"error_description": "Detailed error message"
		}

Item Details	
Name	Description
Request Parameters	
access_token	(Mandatory) OAuth access token obtained as a result of successful user authentication.
Response Parameters	
certificates	Array of certificates for a user.
cert_alias	(Mandatory) Certificate Alias of the certificate.
sam_user_id	(Optional) Specifies SAM User ID when this certificate is to be used for authorise remote signing.
auth_pass	(Mandatory) Specifies password based authentication is applied on this certificate.

certificate	(Mandatory) Certificate Base64 String.
certificate_chain	(Optional) Certificate chain Base64 String.
error	The error code.
error_description	A description message that explains error code.

Table 26 – List User Certificates

6.6 Get User Certificates

This API returns the list of certificates for specified user.

Exposed for: Business Application

https://server:8779/adss/service/csp/v2/users/certs/{user_id}		
HTTP Verb	GET	
Authorization	Bearer {client_access_token}	
Accept	application/json	
App-Name	(Optional) Client Id of requester via CSP proxy	
Accept-Language	en-US	
Status Code	**Message**	**Response Body**

200	OK	{
		"certificates": [{
		"cert_alias": "o5eaqqdiai4j4chlpqt",
		"sam_user_id": "john.doe@ascertia.com",
		"auth_pass": "TRUE",
		"certificate": "MIAGCSqGSI[...]AAAAA="
		"certificate_chain": "MIAGsdGSI[...]AA="
		},{
		"cert_alias": "o5eaqadhlpqt",
		"sam_user_id": "john.doe@ascertia.com",
		"auth_pass": "FALSE",
		"certificate": "MIAGCSqGSI[...]AAA="
		"certificate_chain": "MIAGsdGSI[...]AA="
		}]
		}

400	Bad Request	{
		"error": "60001",
		"error_description": "Detailed error message"
		}

401	Unauthorized	{ "error": "60001", "error_description": "Detailed error message"

		}

404	Not Found	{
		"error": "60001",
		"error_description": "Detailed error message"
		}

500	Internal Server Error	{
		"error": "60001",
		"error_description": "Detailed error message"
		}

Item Details	
Name	Description
Request Parameters	
access_token	(Mandatory) OAuth access token obtained as a result of successful authentication via "client_credentials" grant type.
Response Parameters	
certificates	Array of certificates for a user.
cert_alias	(Mandatory) Certificate Alias of the certificate.
sam_user_id	(Optional) Specifies SAM User ID when this certificate is to be used for authorise remote signing.
auth_pass	(Mandatory) Specifies password based authentication is applied on this certificate.
certificate	(Mandatory) Certificate Base64 String.
certificate_chain	(Optional) Certificate chain Base64 String.
error	The error code.
error_description	A description message that explains error code.

Table 27 – Get User Certificates

7 Signing Services

Signing Services includes API interfaces used by the Virtual CSP.

- Signature Request
- Signature Status

7.1 Signature Request

This API is used to initiate a signing request to the Signing Service.

Signing Service allows to perform authorise remote signing. For more information about Signing Profile configuration see:

Step 4 - Configuring Signing Profile (ascertia.com)

SAM profile configured in Signing Profile must have “Compute Hash at Signing Time” unchecked. For more information, see:

Step 2 - Configuring SAM Profile (ascertia.com)

Exposed for: Virtual CSP

https://server:8779/adss/service/csp/v2/sign/request		
HTTP Verb	POST	
Authorization	Bearer {user_access_token}	
Content-Type	application/json; charset=utf-8	
Accept	application/json	
App-Name	(Optional) Client Id of requester via CSP proxy	
Accept-Language	en-US	

Request Body	{
	"profile_id": "CSP Service Profile",
	"sam_user_id": "john.doe@ascertia.com",
	"cert_alias": "o5eaqqdi9hte17vj6s63ai4j4chlpqt",
	"cert_password": "password",
	"request_description": "Display Text",
	"docs_to_sign_info": [{
	"doc_alias": "DWORD.EXE",
	"hash_algo": "2.16.840.1.101.3.4.2.1",
	"hash_to_sign_base64": "35456465465GCSqGSAAAAAAAAA="
	},{
	"doc_alias": "DWORD.EXE",
	"hash_algo": "2.16.840.1.101.3.4.2.1",
	"hash_to_sign_base64": "35456465465GCSqGSAAAAAAAAA="
	}]
	}	

Status Code	**Message**	**Response Body**
200	OK	{ "request_id": "1613630666", "status": "SUCCESS", "signatures": [{

		<pre>"signature_base64":"35456465465AAAA="
		},
		{
		"signature_base64":"35456465465AAAA="
		}
		]
		}</pre>

400	Bad Request	<pre>{
		"error": "60001",
		"error_description": "Detailed error message"
		}</pre>

401	Unauthorized	<pre>{
		"error": "60001",
		"error_description": "Detailed error message"
		}</pre>

412	Pre-Condition Failed	<pre>{
		"error": "60001",
		"error_description": "Detailed error message"
		}</pre>

404	Not Found	<pre>{
		"error": "60001",
		"error_description": "Detailed error message"
		}</pre>

500	Internal Server Error	<pre>{
		"error": "60001",
		"error_description": "Detailed error message"
		}</pre>

Item Details	
Name	Description
Request Parameters	
access_token	(Mandatory) OAuth access token obtained as a result of successful user authentication.
profile_id	CSP Profile passed in the request message.
sam_user_id	(Optional) Specifies SAM User ID when this certificate is to be used for authorise remote signing.
cert_alias	(Mandatory) Certificate Alias of the certificate used for signing.
cert_password	(Optional) If certificate is password protected.
request_description	(Optional) This is Base64 encoded or plane text that will be displayed on the mobile device when user authorises the transaction. Business application needs to sends the Base64 encoded test to show the Latvian/special characters.
docs_to_sign_info	Array of documents/hashes to be signed.
doc_alias	(Optional) Document name to be signed.
hash_algo	(Mandatory) Hash algorithm OID used to calculate the hash of the document.

hash_to_sign_base64	(Mandatory) Document hash.
Response Parameters	
status	PENDING/TIMEOUT/DECLINED/FAILED/SUCCESS
request_id	(Conditional) Generated by the Signing Server which is later used to check the status of the signature operation.
signature_base64	(Conditional) Array of Signature values when status value is SUCCESS and simple remote signing is performed.
error	The error code.
error_description	A description message that explains error code.

Table 28 – Signature Request

7.2 Signature Status

This API is used to get status of a pending signature request from CSP Service.

Exposed for: Virtual CSP

https://server:8779/adss/service/csp/v2/sign/status		
HTTP Verb	POST	
Authorization	Bearer {user_access_token}	
Content-Type	application/json; charset=utf-8	
Accept	application/json	
App-Name	(Optional) Client Id of requester via CSP proxy.	
Accept-Language	en-US	

Request Body	{
	"profile_id": "CSP Service Profile",
	"request_id": "85963A58745"
	}	

Status Code	**Message**	**Response Body**

200	OK	{
		"status": "SUCCESS",
		"signatures": [
		{
		"signature_base64":"35456465465AAAA="
		},
		{
		"signature_base64":"35456465465AAAA="
		}
		]
		}

400	Bad Request	<pre>{
		"error": "60001",
		"error_description": "Detailed error message"
		}</pre>

401	Unauthorized	<pre>{
		"error": "60001",
		"error_description": "Detailed error message"
		}</pre>

412	Pre-Condition Failed	<pre>{
		"error": "60001",
		"error_description": "Detailed error message"
		}</pre>

404	Not Found	<pre>{
		"error": "60001",
		"error_description": "Detailed error message"
		}</pre>

500	Internal Server Error	<pre>{
		"error": "60001",
		"error_description": "Detailed error message"
		}</pre>

Item Details	
Name	Description
Request Parameters	
access_token	(Mandatory) OAuth access token obtained as a result of successful user authentication.
profile_id	CSP Profile passed in the request message.
requestId	(Mandatory) Specifies the parameter which was provided by the Signing Server in Signature Request API response.
Response Parameters	
status	PENDING/TIMEOUT/DECLINED/FAILED/SUCCESS
signature_base64	(Conditional) Array of Signature values when status value is SUCCESS and authorise remote signing is performed.
error	The error code.
error_description	A description message that explains error code.

Table 29 – Signature Status

8 Error Codes

Errors	
Error Code	Error Description
60001	An internal error occurred while processing the request - see the CSP service debug logs for details
60002	Failed to process request - CSP service is stopped
60003	Failed to process request - CSP service not enabled in license
60004	Failed to process request - CSP service license has expired
60005	Failed to process request - CSP service not enabled in system
60006	Failed to process request - Registered User is not found
60007	Failed to process request - Client secret is missing in request
60008	Failed to process request - User ID or password is not valid
60009	Failed to process request - User is not active
60010	Failed to process request - Client ID is missing in request
60011	Failed to process request - User ID is missing in request
60012	Failed to process request - User password is missing in request
60013	Failed to process request - User email is not valid
60014	Failed to process request - User new password is missing in request
60015	Failed to process request - Request ID is missing in request
60016	Failed to process request - OTP is not valid
60017	Failed to process request - Email OTP is missing in request
60018	Failed to process request - Email address is missing in request
60019	Failed to process request - CSP Profile does not exist
60020	Failed to process request - CSP Profile ID is missing in request
60021	Failed to process request - User certificate is not valid
60022	Failed to process request - Certificate alias is missing in request
60023	Failed to process request - Certificate auth_pass is missing in request
60024	Failed to process request - Certificate is missing in the request
60025	Failed to process request - Certificate list is empty
60026	Failed to process request - OTP for Password Change is missing in request
60027	Failed to process request - Profile ID is missing in the request
60028	Failed to process request - User is already registered
60029	Failed to process request - User is blocked
60030	Failed to process request - Certificate is not found
60031	Failed to process request - Status is not valid

60032	Failed to process request - Password is less than minimum password length
60033	Failed to process request - Password must contains digits and special characters
60034	Failed to process request - Password must contains upper and lower case letters
60035	Failed to process request - Fetch size should be greater than 0
60036	Failed to process request - Start pointer should not be negative
60037	Failed to process request - Certificate entry does not exist
60038	Failed to process request - Certificate entry already exists
60039	Failed to process request - Invalid User ID
60040	Failed to process request - Invalid Certificate chain
60041	Failed to process request - Invalid Certificate chain
60042	Failed to process request - HMAC Key not configured in CSP service manager
60043	Failed to process request - The request is missing a required parameter, includes an invalid parameter value, includes a parameter more than once, or is otherwise malformed
60044	Failed to process request - Authentication type mismatch with profile's authentication
60045	Failed to process request - Invalid or expired token
60046	Failed to process request - Existing email mismatch
60047	Failed to process request - Refresh token is missing
60048	Failed to process request - Hashing algo is missing
60049	Failed to process request - Data to be signed is missing
60050	Failed to process request - Authentication profile is missing
60051	Failed to process request - Single Sign On token is missing
60052	Failed to process request - Authentication method is missing
60053	Failed to process request - Authentication profile is not allowed to client
60054	Failed to process request - Invalid grant type
60055	Failed to process request - Invalid hash algorithm or data
60056	Failed to process request - Invalid method type
Errors related to the access control	
86000	Failed to authenticate client - TLS client authentication certificate has expired
86001	Failed to authenticate client - TLS certificate CN does not match with client ID
86002	Failed to authenticate client - TLS client certificate is revoked
86003	Failed to authenticate client - revocation status for TLS client certificate is unknown
86004	Failed to authenticate client - client ID does not match with the client identified by TLS client certificate
86005	Failed to authenticate client - TLS client certificate does not match with the configured client certificate

86011	Failed to authenticate client - Client ID does not exist
86012	An error occurred while communicating with database - see the service debug logs for details
86015	Failed to authenticate client - Client ID is not found in the request
86017	Failed to authenticate client - client is marked inactive
86018	Failed to authorise client - service is not allowed to this client.
86019	Failed to authorise client - service profile does not exist.
86020	Failed to authorise client - service profile is inactive
86021	Failed to authorise client - profile is not allowed to this client
86022	Failed to authorise client - default profile not configured and neither found in request
86023	Failed to authorise client - default profile is inactive
86024	Failed to authorise client – client secret is invalid

Table 30 – Error Codes

*** End of Document ***