



JOB AID: CSD Integration Testing

Central Supplier Database

Document Statistics

Property	Value
Created by:	CSD Team
Version:	1.1
Date Created:	03 February 2016
Last updated:	15 February 2016
Location:	C:\Data\CSD\Design and arch\Consumers\JOB AID_CSD Integration Testing v1.1_20160215.docx



Table of Content

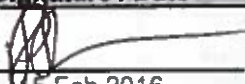
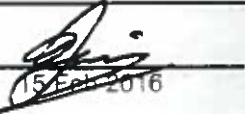
1. APPROVAL	3
2. INTRODUCTION.....	4
3. PURPOSE	4
4. AUDIENCE	4
5. INTEGRATION TESTING.....	5
5.1. OVERVIEW.....	5
5.2. PRINCIPLES	5
5.3. ORGAN OF STATE REQUESTS ACCESS	5
5.4. CONFIRM TEST ENVIRONMENT CONNECTIVITY	5
5.5. CONFIRM TEST ENVIRONMENT ACCESS	5
5.6. INVOKE SERVICES	7
5.7. TEST DATA	7
6. OPEN ITEMS	9
7. OUTSTANDING ISSUES.....	9
8. DECISION LOG	9
9. REVISION HISTORY.....	9
10. REFERENCES.....	9
11. ABBREVIATIONS.....	10
12. APPENDIX A: AUTHENTICATION AND AUTHORISATION	11
12.1. OVERVIEW	11
12.2. AUTHENTICATION REQUEST	11
12.2.1. SAMPLE XML.....	11
12.3. AUTHENTICATION RESPONSE.....	11
12.3.1. SAMPLE XML.....	12
12.4. BUSINESS RULES	12
12.4.1. SAMPLE XML.....	13
13. APPENDIX B: SERVICE ERRORS.....	14
13.1. OVERVIEW	14
13.2. HTTP ERRORS	14
13.3. CSD BUSINESS ERRORS	14
14. APPENDIX A: REST AND HTTP	15
14.1. HTTP REQUEST FORMAT	15
14.2. HTTP RESPONSE FORMAT	15



1. APPROVAL

Project	Central Supplier Database		
Document Title	Job Aid: Integration Testing		
Status	Published	Document Revision	1.1

The signatories hereof, being duly authorised thereto, by their signatures hereto authorise the execution of the work detailed herein, or confirm their acceptance of the contents hereof and authorise the implementation/adoption thereof, as the case may be, for and on behalf of the parties represented by them.

Designation	Name	Signature / Date
Director: Central Supplier Database Office of the Chief Procurement Officer	Tumelo Ntlaba	 15 Feb 2016
Central Supplier Database Architect Accenture	CSD Team	 15 Feb 2016



2. INTRODUCTION

The Central Supplier Database (CSD) system serves as a central master of supplier information for Organs of State (OoS):

- National departments;
- Provincial departments;
- Local government; and
- State Owned Entities (SOE's).

In order for OoS to make use of the supplier information, it is made available using three mechanisms:

1. Online functionality whereby users can search for suppliers;
2. Real time (RESTfull) services whereby systems can consume supplier information; and
3. Bulk data (XML format) whereby systems can consume large volumes of supplier information.

The remainder of this document describes the available environment for integration testing purposes of the Real Time services as well as the Bulk data services.

3. PURPOSE

The purpose of this document is to describe the integration testing approach and available environment.

4. AUDIENCE

The audience of this document includes the teams responsible for preparing organ of state systems to consume CSD supplier information in bulk.



5. INTEGRATION TESTING

5.1. OVERVIEW

An integration testing environment is available to OoS to perform integration testing to the Real Time (refer 5) and Bulk data (refer 4) services. For the remainder of this document it is assumed that the reader is familiar both as covered in reference 3 and 4.

5.2. PRINCIPLES

The following principles are noted:

- It is the responsibility of the OoS to conduct integration testing using the provided environment. The CSD team only supports OoS if needed and does not require any formal sign off;
- A vendor cannot request access on behalf of an Organ of State;
- The on-boarding process only applies to the CSD production environment (refer 3); and
- Using the UAT environment, an OoS creates and maintains suppliers which functions as test data that can be used for API integration testing purposes. This is described in more detail in section 5.7 below.

5.3. ORGAN OF STATE REQUESTS ACCESS

An OoS requests access to the integration testing environment (CSD UAT) as follows:

- An Organ of State requests access to the testing environment by emailing technical.support@csd.gov.za ;
- OCPO approves the request and CSD support creates a system account and returns the details to the requestor; and
- The system account is used to invoke the Real Time or Bulk data services.

5.4. CONFIRM TEST ENVIRONMENT CONNECTIVITY

An OoS can confirm access and connectivity to the test environment as follows:

- To confirm connectivity to the test environment, navigate to the following link: <https://uatapi.csd.gov.za/api/MasterData/Province>;
- The above service returns a JSON formatted list of provinces. If the above link is accessible, the browser attempts to download a file containing the list of provinces. This indicates that connectivity to the CSD integration environment is established;
- If the above services is not accessible, please contact technical.support@csd.gov.za.

5.5. CONFIRM TEST ENVIRONMENT ACCESS

In order to access any of the Real Time or Bulk services, an OoS system needs to authenticate to obtain a session token. Each subsequent Application Programming Interface (API) call requires the token to authorise the request. Section 12 below describes the Authentication API method in further detail.

If an OoS prefers to confirm access to the CSD UAT environment prior to development, a pre-configured Google Postman collection is available to invoke CSD API methods.



Figure 2 – Header

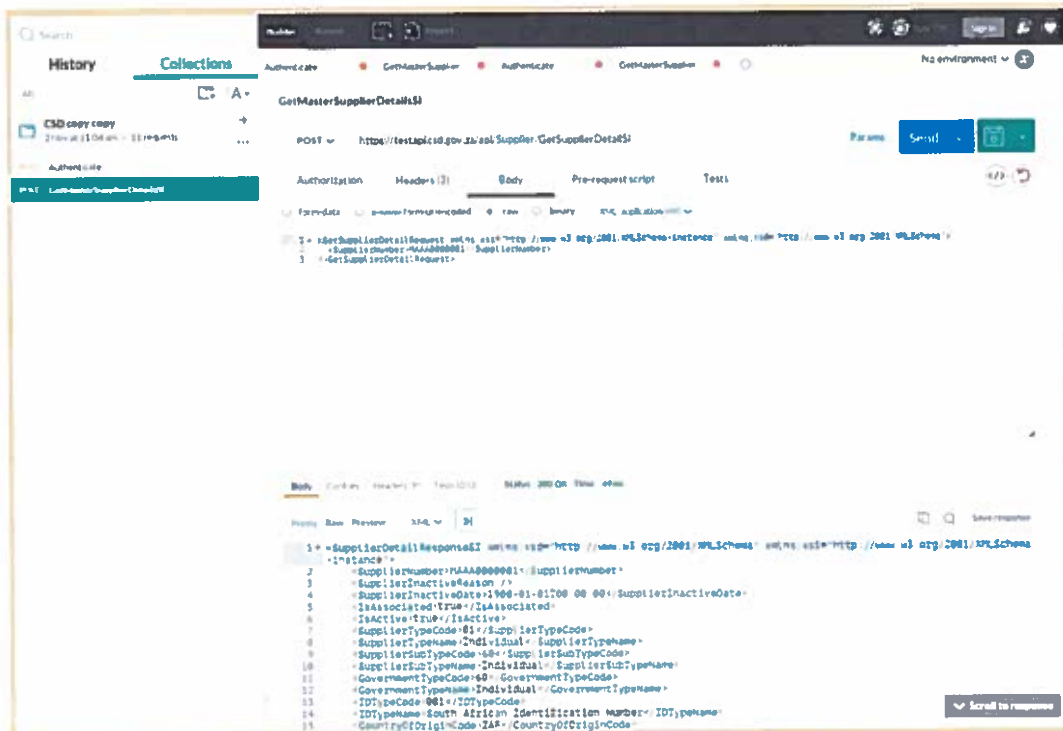


Figure 3 – Supplier identification

5.6. INVOKE SERVICES

Once connectivity is established the following services can be invoked:

- Real time services covered in reference 5 to obtain supplier specific information;
- Bulk xml services to obtain the bulk files as covered in reference 4; and
- Bulk file download utility can be used to download the bulk files as covered in reference 8.

5.7. TEST DATA

An OoS is responsible to perform integration testing, create their own integration test scenarios and maintain their own supplier test data.

In order to facilitate integration testing, the UAT environment is made available to OoS to create and maintain suppliers and also invoke API services to obtain the data. (<https://uatapi.csd.gov.za/>).

An OoS uses the standard supplier functionality to create a user account and create and maintain supplier data as if being a supplier. Each OoS (acting as if being a supplier) can maintain various supplier types and scenarios to serve as test data.

Using the front end (as if being a supplier), an OoS would only have access to the suppliers they maintain which is exactly the way suppliers maintain their own data on the production environment.



In order to test both the Real Time supplier services and the Bulk Data services, an OoS would invoke the API services and methods as described in references 4 and 5, to obtain supplier information and thereby perform test integration and execute various test scenarios.

As designed, all the suppliers maintained by OoS users on the UAT environment (as if being a supplier), is thus accessible by all OoS systems invoking the API services. This thus simulates the same behaviour provided by the production environment.

The Bulk Data services are also available to obtain the bulk files. Batch jobs are executed at scheduled times which creates the full, monthly and daily files. Initially, as the API is made available on the UAT environment, the full file(s) is created which includes all the suppliers on the UAT environment at that point in time.

For testing purposes, the UAT environment executes the daily batch file creation job every half hour and includes all the changes made on that day. The production environment executes this job only once a day and include supplier changes of the previous day.

In this way, an OoS are able to create a supplier using the standard supplier functionality, maintain the data as required and within a half hour, obtain and process the daily file.

The only way the monthly file creation differs from the daily, is that it contains all the new and updated suppliers of the previous month. The processing logic is exactly the same and therefore the monthly file creation on UAT will be executed on a monthly basis to mirror the production environment.

**6. OPEN ITEMS**

No open items are noted.

7. OUTSTANDING ISSUES

No outstanding issues noted.

8. DECISION LOG

No decision are noted.

9. REVISION HISTORY

Version	Revision Description	Date	By
0.1	Created	28/01/2016	CSD Team
1.0	Published	05/02/2016	CSD Team
1.1	Updated UAT links	15/02/2016	CSD Team

10. REFERENCES

No	Name	Version	Date
1.	CSD Master Data OCPO web site > Buyer's Area > Supplier Management http://ocpo.treasury.gov.za/Buyers_Area/Pages/Supplier-Management.aspx	N/A	N/A
2.	Commodity List OCPO web site > Supplier's Area > Central Supplier Database http://ocpo.treasury.gov.za/Suppliers_Area/Pages/Central-Supplier-Database.aspx	N/A	N/A
3.	CSD Job Aid: On-boarding http://ocpo.treasury.gov.za/Buyers_Area/Pages/Supplier-Management.aspx	N/A	N/A
4.	CSD Job Aid: Bulk Data http://ocpo.treasury.gov.za/Buyers_Area/Pages/Supplier-Management.aspx	N/A	N/A
5.	CSD Real Time Services Job Aid http://ocpo.treasury.gov.za/Buyers_Area/Pages/Supplier-Management.aspx	N/A	N/A
6.	Hypertext Transfer Protocol -- HTTP/1.1 World Wide Web Consortium (W3C) http://www.w3.org Internet Engineering Task Force (IETF) https://tools.ietf.org/html/rfc7230 http://tools.ietf.org/html/7237	N/A	N/A
7.	HTTP response codes: http://www.w3.org/Protocols/rfc2616/rfc2616-sec6.html#sec6.1.1	N/A	N/A
8.	CSD Job Aid: Bulk File Download Utility	N/A	N/A



No	Name	Version	Date
	http://ocpo.treasury.gov.za/Buyers_Area/Pages/Supplier-Management.aspx		

11. ABBREVIATIONS

Abbreviation	Description
API	Application Programming Interface
CSD	Central Supplier Database
OoS	Organ of State
SOE	State Owned Entity



12. APPENDIX A: AUTHENTICATION AND AUTHORISATION

12.1. OVERVIEW

The authentication service verifies the credentials of the consumer and establishes a session stored on the CSD. Once authenticated, this service returns a token (GUID), which needs to be passed in with all subsequent service calls as the Authorisation field in the HTTP Request Header. Each individual service in turn uses the token to authorise the service request.

12.2. AUTHENTICATION REQUEST

The authentication request format is as follows (on UAT):

Field	Format	Value
General		
Operation		POST
URI		https://uatapi.csd.gov.za/api/Authenticate
Header		
Content-Type		application/xml
Accept		application/xml
Body		
AuthenticationRequest	•	• Container node/object name
AcceptTermsandConditions	• Boolean (Required)	• True/False
Email	• String (Required)	• Email address of the consumer as per enrolled credentials
Password	• String (Required)	• Password of the consumer as per enrolled credentials

12.2.1. SAMPLE XML

```
<AuthenticationRequest xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <AcceptTermsandConditions>true</AcceptTermsandConditions>
  <Email>ConsumerEmailaddress@domain.gov.za</Email>
  <Password>ConsumerPassword</Password>
</AuthenticationRequest>
```

12.3. AUTHENTICATION RESPONSE

The authentication response contains a number of fields, including:

Field	Format	Value/Rules
Response codes		
		• This is a list of standard HTTP codes as described in section 13 Appendix B: Service Errors.
Header		
		• Standard HTTP response header as defined by reference 6.
Body		
AuthenticationResponse		• Container node/object name.
IsAccountActive	• Boolean	• Indicates if the account is active. Enables CSD support to de-activate an account
IsAccountVerified	• Boolean	• Indicates if the account has been verified by completing the registration process
IsPasswordExpired	• Boolean	• Indicates if the password has expired
IsSuspended	• Boolean	• Indicates if the account is suspended (after three failed login attempts)



Field	Format	Value/Rules
LockoutEnabled	• Boolean	• Indicates if the account is locked (after three failed suspensions)
SuspendedUntil	• DateTime	• Indicates until when the account is suspended
Token	• GUID	• Token value
TokenCreatedDateTime	• DateTime	• When the token was created
TokenExpireDateTime	• DateTime	• When the token expires
Body response when error		
ErrorResponse		• Container node/object name in case of error
ErrorCode	• String (optional)	• Code of the error
IsError	• Boolean	• Indicates if there is a business error
ErrorResponse	• String	• Error response description

12.3.1. SAMPLE XML

```
<AuthenticationResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <IsAccountActive>true</IsAccountActive>
  <IsAccountVerified>true</IsAccountVerified>
  <IsPasswordExpired>>false</IsPasswordExpired>
  <IsSuspended>>false</IsSuspended>
  <LockoutEnabled>>false</LockoutEnabled>
  <SuspendedUntil>1900-01-01T01:00:00</SuspendedUntil>
  <Token>1f3c5e07-dd90-4679-ab9d-9354767c6d6e</Token>
  <TokenCreatedDateTime>2015-11-03T11:32:10.368661+02:00</TokenCreatedDateTime>
  <TokenExpireDateTime>2015-11-03T12:02:10.368661+02:00</TokenExpireDateTime>
</AuthenticationResponse>
```

12.4. BUSINESS RULES

The following business rules are noted:

- The token is valid for a configurable period and the expiry date time indicated by field TokenExpireDateTime;
- Whilst the token is in use by subsequent service calls, it is automatically refreshed and the TokenExpireDateTime updated. When there is no activity for an extended period (CSD configurable parameter) and the TokenExpireDateTime reached, the token expires and authentication required;
- If the account is suspended, IsSuspended is set as true and remains suspended until the SuspendedUntil date time;
- If the account is locked, LockoutEnabled is set as true and the account has to be unlocked via the CSD support team;
- The following business error responses are noted:
 - If the AcceptTermsandConditions is set to false
Message = Terms and conditions not accepted.
 - If the email address does not exist:
Message = User emailaddress@domain.gov.za not found
 - If the account is locked:
Message = User emailaddress@domain.gov.za account locked out
 - If the account is not active:
Message = User emailaddress@domain.gov.za account deactivated
 - If the password expired:
Message = User emailaddress@domain.gov.za password expired
 - If the password is incorrect: and
Message = Login failed attempt 1 (or 2). Please verify your Email and Password. Note: Your account will be suspended on the 3rd attempt



- o After three failed login attempts.
Message = User account suspended. Please retry after 2015-11-03
01:03:58 PM

12.4.1. SAMPLE XML

```
<ErrorResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <ErrorCode i:nil="true" />
  <IsError>true</IsError>
  <Message>Login failed attempt 1. Please verify your Email and Password. Note: Your account will be
suspended on the 3rd attempt.</Message>
</ErrorResponse>
```



13. APPENDIX B: SERVICE ERRORS

13.1. OVERVIEW

This section describes the way the CSD raises and returns business related errors to consumers. As noted in the above diagram, the HTTP response contains a response codes section made available by the protocol. The CSD uses some of these codes to indicate the status of requests and is detailed in the following sections.

13.2. HTTP ERRORS

As described in section 14.2 HTTP response format above, the HTTP <Response code> section contains the status of the HTTP request. This response is a standard 3-digit HTTP response codes (e.g. "404" - Not Found, or 500 - Internal Server Error, etc.) as defined by open standards response codes (refer 7) of the World Wide Web Consortium (W3C) open standards for Hypertext Transfer Protocol -- HTTP/1.1 and the Internet Engineering Task Force (IETF) (refer 6).

The CSD uses these response codes as follows:

- HTTP response code 200 = "OK" is returned when there is no errors;
- HTTP response code 400 = "BadRequest" is returned when there is an error in which case the ErrorResponse below needs to be interrogated for further details; and
- HTTP response code 401 = "Unauthorised" is returned when:
 - The user could not be authenticated;
This usually occurs when the user credentials are incorrect in which case the ErrorResponse below needs to be interrogated for further details.
 - No token or an incorrect token is passed in;
 - The user does not have the required permission to perform the action.
This usually occurs when the role to which the user is assigned does not have the required permission to perform the action.

NOTE: HTTP response 401 can be returned by each service if the token is not passed in, incorrect or expired and therefore is not repeated in the following sections.

13.3. CSD BUSINESS ERRORS

Business errors or exceptions generated by the CSD application are returned in the header as HTTP response code 400 or 401 along with the ErrorResponse object in the body having the following structure:

- ErrorResponse – the error object
 - ErrorCode – String with the error code (optional);
 - IsError – Boolean field that indicates if there was an error; and
 - Message – String with a CSD specific error message.

CSD errors is a function of the business rules implemented by the underlying business components and is documented per service in the sections below.



14. APPENDIX A: REST AND HTTP

14.1. HTTP REQUEST FORMAT

An HTTP request has the following format:

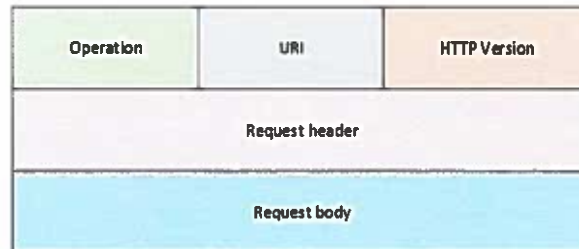


Figure 4 – HTTP request format

- <Operation> is one of the HTTP methods like GET, PUT, POST, DELETE, OPTIONS, etc;
- <URI> is the URI of the resource on which the operation is going to be performed;
- <HTTP Version> is the version of HTTP, generally "HTTP v1.1";
- <Request Header> contains the metadata as a collection of key-value pairs of headers and their values. These settings contain information about the message and its sender like client type, the formats client supports, format type of the message body, cache settings for the response, and a lot more information; and
- <Request Body> is the message content. In a RESTful service, it contains the representations of resources.

14.2. HTTP RESPONSE FORMAT

An HTTP response has the following format:

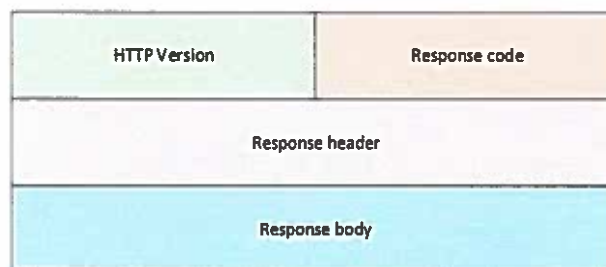


Figure 5 – HTTP response format

- <HTTP Version> is the version of HTTP, generally "HTTP v1.1";
- <Response code>, which contains the status of the request. This response code is generally the 3-digit HTTP status code (refer 6);
- <Response Header> contains the metadata as a collection of key-value pairs of headers and their values. These settings contain information about the message and its sender like client type, the formats client supports, format type of the message body, cache settings for the response, and a lot more information; and
- <Response Body> is the message content. In a RESTful service, it contains the representations of resources.