MINISTRY OF HEALTH
MANATŪ HAUORA

Health Identity
eSAM Address Web Services
Integration Guide

Version 8.0
August 2013

Table Of Contents

1. Introduction 3

Business Context 3

Access, Authorisation and Security Requirements 3

Prerequisites 4

2. Overview 5

SuggestAddress 5

FindAddress 5

Get Address Details 6

3. Understanding the Web Services 7

SuggestAddress 7

FindAddress 11

GetAddressDetails 17

GetExtraDetails 22

GetCAUExtraDetails 25

4. Testing the Web Services 28

Basic Connectivity Testing 28

Standard Test Approach 28

5. Integrating the Web Services 32

.NET Sample 32

Java Sample 35

6. Appendix 37

Web Services Architecture 37

Support 37

Connection Endpoints 37

1. Introduction

The Health Identity Programme (HIP) has delivered Address Validation Web Services to support accurate and standardised Address and Geospatial data within health sector applications. The Web Services are called eSAM and are available for integration into sector patient management applications. This resource pack will enable and support the integration and use of the new services.

Note: eSAM Address Validation Web Services are a replacement for the existing geocoding mechanisms used in the health sector – Geostan and NADIS.

Business Context

Users of sector applications need to enter accurate addresses when adding and updating demographic details. There are a number of reasons for this including locating and sending mail to a patient or provider, business processes that depend on geocoding such as population-based funding, and reporting and monitoring the efficacy and coverage of health initiatives such as screening.

The new eSAM services perform a match against an authoritative dataset of validated (known) addresses as opposed to estimating the location of any submitted address (whether it exists or not) using a geocoding engine.

Although the objective is to validate every address against the reference data, there will be times when a sector user is unable to do this and will need to have a mechanism available in the local application to store an un-validated address with a flag to indicate the need for a business process to subsequently follow-up and resolve. These flagged records also require a reporting mechanism to monitor unresolved exceptions.

For performance reasons, it is strongly recommended that the SuggestAddress Web Service development includes logic that only calls the Web Service on entry of the 3rd character rather than the first or second characters.

Access, Authorisation and Security Requirements

The eSAM Address Web Services must be used over Connected Health and only by authorised health provider organisations and users.

Each Web Service call must include the provider HPI-ORG (organisation) identifier, unique Application identifier and User ID.

The HPI-ORG identifier is issued by Ministry of Health from the Health Provider Index for each health provider. This should be implemented as a configurable setting within the vendor application.

The Application identifier is issued to each sector vendor application and has a unique value assigned by MOH.

The User ID will reflect the specific ID of the user accessing the Address service from the local application i.e. the individual's login ID.

Once integration of the services within a local application is completed, a compliance test will be necessary before live production access is approved. This will be arranged with the Ministry's health identity team at a time that suits both parties and standard test data will be issued for the test.

Prerequisites

Become a Health Network member	http://www.ithealthboard.health.nz/connectedhealth
Complete Address service access request	http://www.health.govt.nz/our-work/health-identity/addressing-and-geocoding/addressing-and-geocoding-information-health-providers

2. Overview

The eSAMMoHAddress Web Service allows an application to perform an address search using a predictive text search (similar to that used by Google) or an address search by supplying a list of address strings. The Web Service includes 5 operations:

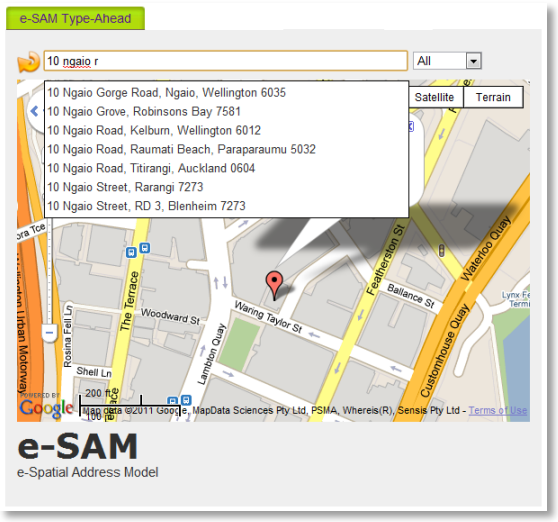
- SuggestAddress
- FindAddress
- GetAddressDetails
- GetExtraDetails
- GetCAUExtraDetails

The eSAMMoHAddress Web Service can be integrated into applications by:

- Type-ahead address input field into a Web page (using HTML and Javascript). This uses a REST protocol where the request is passed to the Web Service via HTTP GET and results returned as JSON.
- Five address input fields passed to FindAddress using SOAP over the HTTP POST protocol. An example of this would be a simple 5 line search form displayed on a Web page.

SuggestAddress

SuggestAddress is an incremental search interface that can be used to present users with a dynamic list of addresses as characters are entered into the search window. Below is an example of what this may look like to a user:



SuggestAddress was designed for New Zealand addresses and references an authoritative dataset supplied by NZ Post and updated monthly. Typically SuggestAddress is used for 'point of entry' systems that provide the user with the ability to decide on the 'correct' address.

FindAddress

FindAddress is a simple search interface that accepts five lines of address data and returns a list of address matches. Each of the address returned includes a MatchScore (confidence score) which indicates how closely the address matches the input supplied. Typically, FindAddress is used for:

- Systems that require an address search by supplying a set of free-text fields

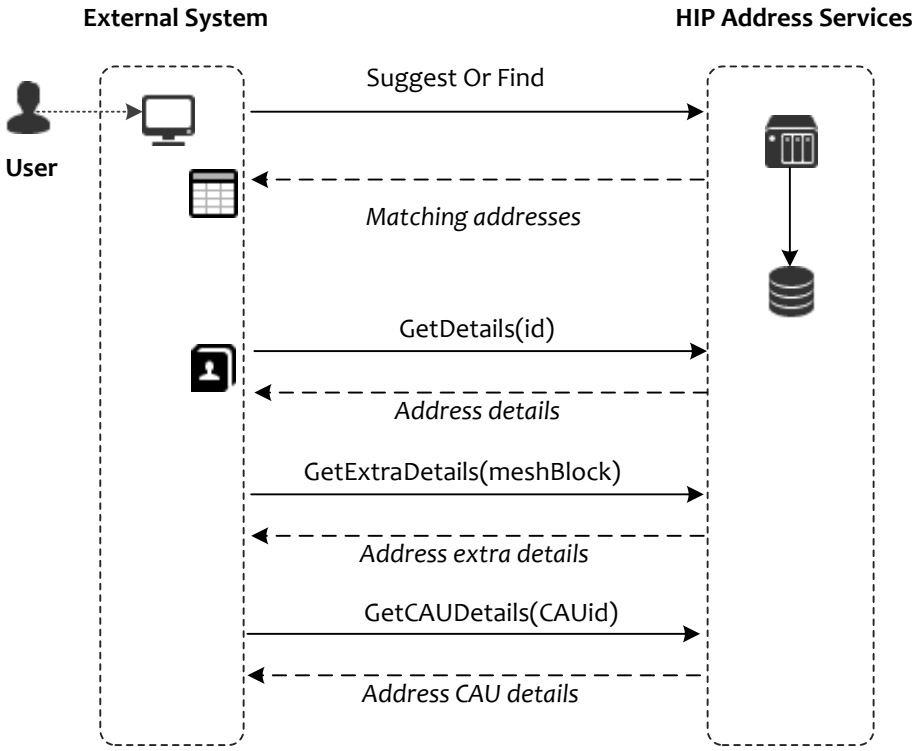
- Machine to machine address validation

FindAddress is also useful for legacy applications which don't support asynchronous services (as exposed through SuggestAddress).

Get Address Details

There are three services which will return Address Details (GetAddressDetails, GetExtraDetails and GetCAUExtraDetails). In an online scenario the user interface will generally display users with a list of addresses (returned either from FindAddress or dynamically using SuggestAddress). When the user selects an address the application will then retrieve details using the Address ID. Please refer to the schema to obtain lower-level data definitions.

The diagram below illustrates this typical interaction:



Note: the data elements that relate to census values (e.g. meshblock) will be implemented with 2006 values initially and updated with each census.

3. Understanding the Web Services

This section explains each Address Web Service in more detail. Each Web Service request/response has a header and body. REST services will typically be accessed as an HTTP GET as opposed to POST with SOAP.

SuggestAddress

The SuggestAddress request is called every time a character is typed into an application that has implemented the address predictive search. It returns a list of addresses from which the user can choose one. Usually, this service would be used in an interactive mode rather than as part of system integration.

Request

Parameters for SuggestAddress:

Parameter	Type	Mandatory	Comments
SearchFor	string	Yes	The partial address string used for searching.
SearchType	string	No	e-SAM is designed to take addressing information from a variety of sources. Currently the sources for e-SAM are NZ Post (NPAD) and LINZ. This parameter determines which source information is required. Valid values for SearchType are: Physical – LINZ (Physical address data) Postal – NZ Post (Postal address data) All – Both (default) L – LINZ (Physical address data) P – NZ Post (Postal address data) B – Both (default)
MaxResults	integer	No	Maximum number of results to return. The default is 20, the maximum is 100.
UniqueOrganisationCode	string	Yes	MoH Mandated
UniqueUserId	string	Yes	MoH Mandated
UniqueApplicationId	string	Yes	MoH Mandated

Table 1, SuggestAddress arguments

Response

HEADER			
Field	**Type**	**Width**	**Comments**
ResultCode	String	5	Code relates to ResultCodes returned for each query • A0000 – Successful • E1002 – Search Type must be blank, L, P, B, Physical, Postal or All.
BODY			
Field	**Type**	**Width**	**Comments**
UniqueId	integer		Unique Address ID stored in ESAM databases
FullAddress	string	397	Address formatted into a single line.
SourceDesc	string	50	Description of source, one of: • Postal • Physical • Physical – Sub Address • Postal – Not Delivered • Physical – Not Delivered • Box\Bag • Box\Bag – Not Delivered • Postal/Physical

Table 2, SuggestAddress call results

Examples:

SOAP	
Request	<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:esam="http://esam.co.nz/eSAMMoHAddress_v01/"> <soapenv:Header/> <soapenv:Body> <esam:SuggestAddressRequest> <SearchFor>133 Molesworth</SearchFor> <SearchType>All</SearchType> <MaxResults>10</MaxResults> <UniqueOrganisationCode>myOrg</UniqueOrganisationCode> <UniqueUserId>myUsername</UniqueUserId> <UniqueApplicationId>myApp</UniqueApplicationId> </esam:SuggestAddressRequest> </soapenv:Body> </soapenv:Envelope>
Response	<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"> <soapenv:Header> <ns:BAMEvent activityID="b870b61e-30eb-4299-83bf-7d925579906f"

	xmlns:ns="http://wso2.org/ns/2010/10/bam"/> </soapenv:Header> <soapenv:Body> <ns1:SuggestAddressResponse xmlns:ns1="http://esam.co.nz/eSAMMoHAddress_v01/"> <Header> <ResultCode>A0000</ResultCode> </Header> <AddressSuggestion> <UniqueId>98404</UniqueId> <FullAddress>133 Molesworth Street, New Plymouth 4312</FullAddress> <SourceDesc>Postal\Physical</SourceDesc> </AddressSuggestion> <AddressSuggestion> <UniqueId>477616</UniqueId> <FullAddress>133 Molesworth Street, Taita, Lower Hutt 5011</FullAddress> <SourceDesc>Postal\Physical</SourceDesc> </AddressSuggestion> <AddressSuggestion> <UniqueId>1112139</UniqueId> <FullAddress>133 Molesworth Street, Thorndon, Wellington 6011</FullAddress> <SourceDesc>Postal\Physical - Not Delivered</SourceDesc> </AddressSuggestion> <AddressSuggestion> <UniqueId>117370</UniqueId> <FullAddress>133H Molesworth Street, New Plymouth 4312</FullAddress> <SourceDesc>Postal\Physical - Not Delivered</SourceDesc> </AddressSuggestion> </ns1:SuggestAddressResponse> </soapenv:Body> </soapenv:Envelope>
REST	
Request	http://rest-test.moh.health.nz/services/eSAMMoHAddressREST_v01/SuggestAddress?SearchType=All&UniqueApplicationId=1&UniqueOrganisationCode=1&MaxResults=10&UniqueUserId=myUserName&SearchFor=133%20Molesworth%20Street

Response	{
	   "ResultCode": "A0000",
	   "AddressSuggestion": [
	            {

```
            "UniqueId": "98404",
            "SourceDesc": "Postal\\Physical",
            "FullAddress": "133 Molesworth Street, New Plymouth 4312"
        },
            {
            "UniqueId": "477616",
            "SourceDesc": "Postal\\Physical",
            "FullAddress": "133 Molesworth Street, Taita, Lower Hutt 5011"
        },
            {
            "UniqueId": "1112139",
            "SourceDesc": "Postal\\Physical - Not Delivered",
            "FullAddress": "133 Molesworth Street, Thorndon, Wellington 6011"
        },
            {
            "UniqueId": "117370",
            "SourceDesc": "Postal\\Physical - Not Delivered",
            "FullAddress": "133H Molesworth Street, New Plymouth 4312"
        }
    ]
}
```

FindAddress

The FindAddress expects a set of address strings, parses them into the core address components then searches for candidate addresses based upon the address lines supplied. Although the FindAddress call will allow five address lines it will also work if the address is supplied entirely in AddressLine1.

If the address is passed entirely in AddressLine1 the components should be separated by commas. For example: “10 Ngaio Gorge Road, Ngaio, Wellington” rather than “10 Ngaio Gorge Road Ngaio Wellington”. This is because the parser (separates the address lines into their core components) works more efficiently if the lines are separated by commas). There are no specific rules about the contents of the five lines – they simply reflect the conventional lines of information used when creating a postal address.

Upon parsing the address lines the service then returns a list of candidate addresses to the calling application\user to decide what address is the most appropriate.

The candidate addresses are returned with a `MatchScore` and a list of flags indicating the type of match found.

Request

Parameters for FindAddress:

Parameter	Type	Mandatory	Comments
`AddressText1`	string	At least one of these fields is required.	First address line
`AddressTest2`	string		Second address line
`AddressText3`	string		Third address line
`AddressText4`	string		Fourth address line
`AddressText5`	string		Fifth address line
`SearchType`	string	No	e-SAM is designed to take addressing information from a variety of sources. Currently the sources for e-SAM are NZ Post (NPAD) and LINZ. This parameter determines which source information is required. Valid values for SearchType are: • Physical – LINZ (Physical address data) • Postal – NZ Post (Postal address data) • All – Both (default) • L – LINZ (Physical address data) • P – NZ Post (Postal address data) • B – Both (default)
`MaxResults`	integer	No	Maximum number of results to return. The default is 20, the maximum is 100.
`UniqueOrganisationCode`	string	Yes	MoH Manadated
`UniqueUserId`	string	Yes	MoH Manadated
`UniqueApplicationId`	string	Yes	MoH Manadated

Table 3: FindAddress arguments

Response

HEADER			
Field	**Type**	**Width**	**Comments**
ResultCode	String	5	Code relates to ResultCodes returned for each query • A0000 – Successful • E1001 – More than 5 lines of address submitted • E1002 – Search Type must be blank, L, P, B, Physical, Postal or All. • E1003 – No matches were found for the address details submitted • I1004 – Matches found exceed the Maximum Results parameter.
BODY			
Field	**Type**	**Width**	**Comments**
UniqueId	integer		Unique Address ID stored in ESAM databases
FullAddress	string	397	Address formatted into a single line
SourceId	integer		Search source identifier for an address: • 0 – box/bag • 1 – postal/physical • 2 – postal only • 3 – physical only • 4 – physical sub-address only • 10 – box/bag not delivered • 11 – postal/physical not delivered • 12 – postal not delivered
SourceDesc	string	50	Description of address source, one of: • Postal • Physical • Physical – Sub Address • Postal – Not Delivered • Physical – Not Delivered • Box/Bag • Box/Bag – Not Delivered • Postal/Physical
MatchScore	integer		Score to indicate the confidence in the candidate address based upon the input. Maximum value is 100 for a perfect score.
QualityIndicator	integer		Currently the same as MatchCode but will change. Currently the maximum value is 100.
MatchedUnit	string	1	If component supplied matches Y if not an N
MatchedFloor	string	1	If component supplied matches Y if not an N
MatchedNumber	string	1	If component supplied matches Y if not an N. NOTE: This field is used for a PO Box\Private Bag\Urban\Rural and CMB search.
MatchedStreetAlpha	string	1	If component supplied matches Y if not an N

MatchedRoadName	string	1	If component supplied matches Y, if not an N or an A if matched to an alias
MatchedRoadTypeName	string	1	If component supplied matches Y, if not an N or an A if matched to an alias
MatchedRoadSuffixName	string	1	If component supplied matches Y, if not an N or an A if matched to an alias
MatchedSuburb	string	1	If component supplied matches Y, if not an N or an A if matched to an alias
MatchedRuralDelivery	string	1	If component supplied matches Y if not an N
MatchedLobby	string	1	If component supplied matches Y, if not an N or an A if matched to an alias
MatchedCity	string	1	If component supplied matches Y, if not an N or an A if matched to an alias
MatchedBoxBagType	string	1	If component supplied matches Y if not an N
Deliverable	string	1	Y or N whether the address is delivered to by NZ Post
Physical	string	1	Y or N if the address is physical

Table 4: FindAddress results

Examples:

SOAP	
Request	<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:esam="http://esam.co.nz/eSAMMoHAddress_v01/"> <soapenv:Header/> <soapenv:Body> <esam:FindAddressRequest> <AddressText1>133 Molesworth Street</AddressText1> <AddressText2>Wellignton</AddressText2> <SearchType>All</SearchType> <MaxResults>5</MaxResults> <UniqueOrganisationCode>MyOrdgID</UniqueOrganisationCode> <UniqueUserId>MyUniqueID</UniqueUserId> <UniqueApplicationId>MyAppID</UniqueApplicationId> </esam:FindAddressRequest> </soapenv:Body> </soapenv:Envelope>
Response	<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"> <soapenv:Header> <ns:BAMEvent activityID="e1e85dfd-dfe1-44e3-bcb5-9ec3d3606706" xmlns:ns="http://wso2.org/ns/2010/10/bam"/> </soapenv:Header>

	<soapenv:Body> <ns1:FindAddressResponse xmlns:ns1="http://esam.co.nz/eSAMMoHAddress_v01/"> <Header> <ResultCode>A0000</ResultCode> </Header> <FoundAddress> <UniqueId>98404</UniqueId> <FullAddress>133 Molesworth Street, New Plymouth 4312</FullAddress> <SourceId>1</SourceId> <SourceDesc>Postal\Physical</SourceDesc> <MatchScore>65</MatchScore> <QualityIndicator>65</QualityIndicator> <MatchedUnit>Y</MatchedUnit> <MatchedFloor>Y</MatchedFloor> <MatchedNumber>Y</MatchedNumber> <MatchedStreetAlpha>Y</MatchedStreetAlpha> <MatchedRoadName>Y</MatchedRoadName> <MatchedRoadTypeName>Y</MatchedRoadTypeName> <MatchedRoadSuffixName>Y</MatchedRoadSuffixName> <MatchedSuburb>N</MatchedSuburb> <MatchedRuralDelivery>N</MatchedRuralDelivery> <MatchedLobby>N</MatchedLobby> <MatchedCity>N</MatchedCity> <MatchedBoxBagType>N</MatchedBoxBagType> <Deliverable>Y</Deliverable> <Physical>Y</Physical> </FoundAddress> { NOTE – Additional addresses removed from this XML fragment } </ns1:FindAddressResponse> </soapenv:Body> </soapenv:Envelope>
REST	
Request	http://rest.moh.health.nz/services/eSAMMoHAddressREST_v01/FindAddress?SearchType=b&AddressText1=83%20Crawford%20Street%20Dunedin%20Central&MaxResults=20&UniqueApplicationId=a&UniqueOrganisationCode=b&UniqueUserId=c
Response	{ "ResultCode": "A0000", "FoundAddress": [{ "UniqueId": "821073", "FullAddress": "83 Crawford Street, Glengarry, Invercargill 9810",

```
            "SourceId": "1",
            "SourceDesc": "Postal\\Physical",
            "MatchScore": "60",
            "QualityIndicator": "60",
            "MatchedUnit": "Y",
            "MatchedFloor": "Y",
            "MatchedNumber": "Y",
            "MatchedStreetAlpha": "Y",
            "MatchedRoadName": "Y",
            "MatchedRoadTypeName": "Y",
            "MatchedRoadSuffixName": "N",
            "MatchedSuburb": "N",
            "MatchedRuralDelivery": "N",
            "MatchedLobby": "N",
            "MatchedCity": "N",
            "MatchedBoxBagType": "N",
            "Deliverable": "Y",
            "Physical": "Y"
        }
    ]
}
{ NOTE – Additional addresses removed from this fragment }
```


The `MatchScore` score is comprised of a summary of the values stored in the 'Matched' columns. Each of these component columns has an individual score based upon what was used in the matching process. These scores are tabulated below:

Component	**Urban Addresses**			
	Full Match		**Alias Match**	
	Flag	**Score**	**Flag**	**Score**
Street Address				
Unit Number (including null)	Y	5	N	n/a
Floor Number (including null)	Y	5	N	n/a
House Number	Y	15	N	n/a
House Alpha (including null)	Y	10	N	n/a
Road Name	Y	10	A	5
Road Type (including null)	Y	15	A	7
Road Suffix (including null)	Y	5	A	2
RD Number	N	n/a	N	n/a

Suburb (including null)	Y	5	A	2
City	Y	30	A	15
Input City matches Suburb or Suburb Alias	Y	30	A	15
Box/Bag Address				
Box/Bag Number	Y	20	N	n/a
Box/Bag Delivery type	Y	10	N	n/a
RD Number	N	n/a	N	n/a
Lobby	Y	30	A	15
City	Y	40	A	20

Table 5, FindAddress Component Match columns with their score that contributes to the overall MatchScore

Component	**Rural Addresses**			
	Full Match		**Alias Match**	
	Flag	**Score**	**Flag**	**Score**
Street Address				
Unit Number (including null)	Y	5	N	n/a
Floor Number (including null)	Y	5	N	n/a
House Number	Y	15	N	n/a
House Alpha (including null)	Y	10	N	n/a
Road Name	Y	10	A	5
Road Type (including null)	Y	15	A	7
Road Suffix (including null)	Y	5	A	2
RD Number	Y	5	N	n/a
Suburb (including null)	N	n/a	N	n/a
City	Y	30	A	15
Input City matches Suburb or Suburb Alias	Y	30	A	15
Box/Bag Address				
Box/Bag Number	Y	20	N	n/a
Box/Bag Delivery type	Y	10	N	n/a
RD Number	Y	30	N	n/a
Lobby	N	n/a	N	n/a
City	Y	40	N	20

Table 6, FindAddress Component Match columns with their score that contributes to the overall MatchScore

GetAddressDetails

The `GetAddressDetails` request returns information for a single address which has been chosen from either the SuggestAddress or FindAddress results.

Request

Parameters for `GetAddressDetails`:

Parameter	Type	Mandatory	Comments
`UniqueId`	integer	Yes	Unique Address ID stored in ESAM databases
`UniqueOrganisationCode`	string	Yes	MoH Mandated
`UniqueUserId`	string	Yes	MoH Mandated
`UniqueApplicationId`	string	Yes	MoH Mandated

Table 7, GetAddressDetails arguments

Response

HEADER			
Field	**Type**	**Width**	**Comments**
`ResultCode`	String	5	Code relates to ResultCodes returned for each query • A0000 – Successful • E1004 – The submitted reference cannot be found.

BODY			
Parameter	**Type**	**Width**	**Comments**
`UniqueId`	integer		Unique Address ID stored in ESAM databases
`AddressLine1`	string	100	First Address Line in NZ Post compliant format
`AddressLine2`	string	100	Second Address Line in NZ Post compliant format
`AddressLine3`	string	100	Third Address Line in NZ Post compliant format
`AddressLine4`	string	100	Fourth Address Line in NZ Post compliant format
`AddressLine5`	string	100	Fifth Address Line in NZ Post compliant format
`Postcode`	string	4	Postcode for address
`NZTMXCoord`	double	(12,3)	X Coordinate in New Zealand Transverse Mercator (9999999.999)
`NZTMYCoord`	double	(12,3)	Y Coordinate in New Zealand Transverse Mercator (9999999.999)
`NZGD2KXCoord`	double	(12,6)	Longitude in New Zealand Geodetic Datum 2000 (999.999999)
`NZGD2KYCoord`	double	(12,6)	Latitude in New Zealand Geodetic Datum 2000 (99.999999)

NZMGXCoord	double	(12,3)	X Coordinate in New Zealand Map Grid (9999999.999)
NZMGYCoord	double	(12,3)	Y Coordinate in New Zealand Map Grid (9999999.999)
Meshblock	string	7	Meshblock correspond to Stats Census Date. E.g. currently 2006 Census meshblocks
StatsCensusDate	date		Date of the Census
DPID	integer		NZ Post Unique Identifier from NPAD
SadId	integer		NZ Post base address unique identifier
SubId	integer		e-Spatial unique identifier for an interpolated address
SourceDesc	string	50	Source Description of an Address
Deliverable	string	1	Does NZ Post deliver mail to the address
Physical	string	1	Is the address a physical address
UnitType	string	20	Type of Unit e.g. Unit, Flat, Apartment
UnitValue	string	20	Value of Unit e.g. 1 in Unit 1
Floor	string	20	Floor Value
StreetNumber	integer		Street Number (excluding StreetAlpha)
StreetAlpha	String	20	Street Alpha (excluding StreetNumber)
RoadName	string	60	Road Name Component e.g Ngaio Gorge in Ngaio Gore Road West
RoadTypeName	string	60	Road Type Component e.g Road in Ngaio Gorge Road West
RoadSuffixName	string	60	Road Suffix Component e.g West in Ngaio Gorge Road West
Suburb	string	60	Suburb
RuralDelivery	string	10	RD Number in for rural postal address
Lobby	string	60	PO Box/Private Bag/CMB or Counter Delivery lobby
CityTown	string	60	Urban City/Town value for an urban address
MailTown	string	60	Rural City/Town value for a rural address
BoxBagNumber	string	20	PO Box, Private Bag or CMB number
BoxBagType	string	20	PO Box, Private Bag, Counter Delivery or CMB

Table 8, GetAddressDetails call results

Examples:

SOAP	
Request	<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:esam="http://esam.co.nz/eSAMMoHAddress_v01/"> <soapenv:Header/> <soapenv:Body> <esam:GetAddressDetailsRequest> <UniqueId>1112139</UniqueId>

	<UniqueOrganisationCode>MyOrdgID</UniqueOrganisationCode> <UniqueUserId>MyUniqueID</UniqueUserId> <UniqueApplicationId>MyAppID</UniqueApplicationId> </esam:GetAddressDetailsRequest> </soapenv:Body> </soapenv:Envelope>
Response	<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"> <soapenv:Header> <ns:BAMEvent activityID="7676e0c9-08f4-4d4a-862e-afc44a8330a7" xmlns:ns="http://wso2.org/ns/2010/10/bam"/> </soapenv:Header> <soapenv:Body> <ns1:GetAddressDetailsResponse xmlns:ns1="http://esam.co.nz/eSAMMoHAddress_v01/"> <Header> <ResultCode>A0000</ResultCode> </Header> <AddressDetails> <UniqueId>1112139</UniqueId> <AddressLine1>133 Molesworth Street</AddressLine1> <AddressLine2>Thorndon</AddressLine2> <AddressLine3>Wellington 6011</AddressLine3> <Postcode>6011</Postcode> <NZTMXCoord>1748894.374</NZTMXCoord> <NZTMYCoord>5429415.743</NZTMYCoord> <NZGD2kXCoord>174.777729</NZGD2kXCoord> <NZGD2kYCoord>-41.272972</NZGD2kYCoord> <NZMGXCoord>2658915.977</NZMGXCoord> <NZMGYCoord>5991127.701</NZMGYCoord> <Meshblock>2119300</Meshblock> <StatsCensusDate>2006-01-01+13:00</StatsCensusDate> <DPID>1112139</DPID> <SadId>606137</SadId> <SourceDesc>Postal\Physical - Not Delivered</SourceDesc> <Deliverable>N</Deliverable> <Physical>Y</Physical> <StreetNumber>133</StreetNumber> <RoadName>Molesworth</RoadName> <RoadTypeName>Street</RoadTypeName> <Suburb>Thorndon</Suburb> <CityTown>Wellington</CityTown>

	</AddressDetails> </ns1:GetAddressDetailsResponse> </soapenv:Body> </soapenv:Envelope>
REST	
Request	http://rest-test.moh.health.nz/services/eSAMMoHAddressREST_v01/GetAddressDetails?UniqueId=1112139&UniqueApplicationId=MyApp&UniqueOrganisationCode=MyOrg&UniqueUserId=MyUser
Response	{ "ResultCode": "A0000", "AddressDetails": { "UniqueId": "1112139", "AddressLine1": "133 Molesworth Street", "AddressLine2": "Thorndon", "AddressLine3": "Wellington 6011", "AddressLine4": "null", "AddressLine5": "null", "Postcode": "6011", "NZTMCoord": { "type": "point", "crs": { "type": "EPSG", "properties": {"code": "2193"} }, "coordinates": ["1748894.374", "5429415.743"] }, "NZGD2kCoord": { "type": "point", "crs": { "type": "EPSG", "properties": {"code": "4326"} }, "coordinates": ["174.777729", "-41.272972"] }, "NZMGCoord": {

```
            "type": "point",
            "crs":          {
               "type": "EPSG",
               "properties": {"code": "27200"}
            },
            "coordinates":          [
               "2658915.977",
               "5991127.701"
            ]
         },
         "Meshblock": "2119300",
         "StatsCensusDate": "2006-01-01+13:00",
         "DPID": "1112139",
         "SadId": "606137",
         "SubId": "null",
         "SourceDesc": "Postal\\Physical - Not Delivered",
         "Deliverable": "N",
         "Physical": "Y",
         "UnitType": "null",
         "UnitValue": "null",
         "Floor": "null",
         "StreetNumber": "133",
         "StreetAlpha": "null",
         "RoadName": "Molesworth",
         "RoadTypeName": "Street",
         "RoadSuffixName": "null",
         "Suburb": "Thorndon",
         "RuralDelivery": "null",
         "Lobby": "null",
         "CityTown": "Wellington",
         "MailTown": "null",
         "BoxBagNumber": "null",
         "BoxBagType": "null"
      }
   }
```

GetExtraDetails

The `GetExtraDetails` request returns information for a single meshblock which has been chosen from the `GetAddressDetails` results.

Request

Parameters for `GetExtraDetails`:

Parameter	Type	Required	Comments
`Meshblock`	String	Yes	Meshblock obtained from `GetAddressDetails`.
`UniqueOrganisationCode`	String	Yes	MoH Mandated
`UniqueUserId`	String	Yes	MoH Mandated
`UniqueApplicationId`	String	Yes	MoH Mandated

Table 9, GetExtraDetails arguments

Response

HEADER			
Field	**Type**	**Width**	**Comments**
`ResultCode`	String	5	Code relates to ResultCodes returned for each query • A0000 – Successful • E1004 – The submitted reference cannot be found.
BODY			
Field	**Type**	**Width**	**Comments**
`Meshblock`	string	7	Meshblock identifier from the Census that corresponds to the date returned in the StatsCensusDate column
`CAUId`	string	7	Area unit identifier from the Census that corresponds to the date returned in the StatsCensusDate column. A group of meshblocks roll up to form an area unit.
`CAUName`	string	80	Area unit name from the Census that corresponds to the date returned in the StatsCensusDate column.
`DomicileCode`	string	4	Health Domicile from the Census that corresponds to the date returned in the StatsCensusDate column.
`DomicileDescription`	string	80	Health Domicile name from the Census that corresponds to the date returned in the StatsCensusDate column.
`DHBCode`	string	8	Specific MoH District Health Board code from the Census that corresponds to the date returned in the StatsCensusDate column.
`DHBName`	string	80	District Health Board name from the Census that corresponds to the date returned in the StatsCensusDate column.
`DHBBdyAssignmentDate`	date*		Date that the District Health Board code/name was assigned

TLAId	string	3	Territorial Local Authority identifier from the Census that corresponds to the date returned in the StatsCensusDate column.
TLAName	string	80	Territorial Local Authority name from the Census that corresponds to the date returned in the StatsCensusDate column.
DeprivationDecile	integer		Numeric value (1 – 10) of Socioeconomic deprivation derived by Otago University.
DeprivationQuintile	Integer		Numeric value (1 – 5) of Socioeconomic deprivation derived by Otago University.
StatsCensusDate	date		Date of the Census

Table 10, GetExtraDetails call results

Examples:

SOAP	
Request	<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:esam="http://esam.co.nz/eSAMMoHAddress_v01/"> <soapenv:Header/> <soapenv:Body> <esam:GetExtraDetailsRequest> <Meshblock>2119300</Meshblock> <UniqueOrganisationCode>MyOrdgID</UniqueOrganisationCode> <UniqueUserId>MyUniqueID</UniqueUserId> <UniqueApplicationId>MyAppID</UniqueApplicationId> </esam:GetExtraDetailsRequest> </soapenv:Body> </soapenv:Envelope>
Response	<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"> <soapenv:Header> <ns:BAMEvent activityID="730f5a3b-c7d2-42b5-ae02-bda94d2ceb5b" xmlns:ns="http://wso2.org/ns/2010/10/bam"/> </soapenv:Header> <soapenv:Body> <ns1:GetExtraDetailsResponse xmlns:ns1="http://esam.co.nz/eSAMMoHAddress_v01/"> <Header> <ResultCode>A0000</ResultCode> </Header> <ExtraDetails> <Meshblock>2119300</Meshblock> <CAUId>572900</CAUId> <CAUName>Thorndon-Tinakori Road</CAUName>

	<DomicileCode>2077</DomicileCode> <DomicileDescription>Thorndon - Tinakori Road</DomicileDescription> <DHBCode>G00036-D</DHBCode> <DHBName>Capital and Coast District Health Board</DHBName> <DHBBdyAssignmentDate>2006-01-01+13:00</DHBBdyAssignmentDate> <TLAId>047</TLAId> <TLAName>Wellington City</TLAName> <DeprivationDecile>7</DeprivationDecile> <DeprivationQuintile>4</DeprivationQuintile> <StatsCensusDate>2006-01-01+13:00</StatsCensusDate> </ExtraDetails> </ns1:GetExtraDetailsResponse> </soapenv:Body> </soapenv:Envelope>
REST	
Request	http://rest.moh.health.nz/services/eSAMMoHAddressREST_v01/GetExtraDetails?Meshblock=2119300&UniqueApplicationId=MyApp&UniqueOrganisationCode=MyOrg&UniqueUserId=UC%20AD03
Response	{ "ResultCode": "A0000", "ExtraDetails": { "Meshblock": "2119300", "CAUId": "572900", "CAUName": "Thorndon-Tinakori Road", "DomicileCode": "2077", "DomicileDescription": "Thorndon - Tinakori Road", "DHBCode": "G00036-D", "DHBName": "Capital and Coast District Health Board", "DHBBdyAssignmentDate": "2006-01-01+13:00", "TLAId": "047", "TLAName": "Wellington City", "DeprivationDecile": "7", "DeprivationQuintile": "4", "StatsCensusDate": "2006-01-01+13:00" } }

GetCAUExtraDetails

The `GetCAUExtraDetails` request returns information for a single census area unit which has been chosen from the `GetExtraDetails` results.

Request

Parameters for `GetCAUExtraDetails`:

Parameter	Type	Required	Comments
`CAUId`	string	Yes	Census area unit identifier obtained from `GetExtraDetails`.
`UnqiueOrganisationCode`	string	Yes	MoH Mandated
`UnqiueUserId`	string	Yes	MoH Mandated
`UnqiueApplicationId`	string	Yes	MoH Mandated

Table 11, GetCAUExtraDetails arguments

Response

`ResultCode` field for `GetCAUExtraDetails`:

There will be a single header row for each set of result(s)

HEADER			
Field	**Type**	**Width**	**Comments**
ResultCode	**String**	**5**	Code relates to ResultCodes returned for each query • A0000 – Successful • E1004 – The submitted reference cannot be found.
BODY			
Field	**Type**	**Width**	**Comments**
`CAUId`	string	7	Census area unit identifier from the Census that corresponds to the date returned in the StatsCensusDate column. A group of meshblocks roll up to form an area unit.
`CAUName`	string	80	Area unit name from the Census that corresponds to the date returned in the StatsCensusDate column.
`DomicileCode`	string	4	Health Domicile from the Census that corresponds to the date returned in the StatsCensusDate column.
`DomicileDescription`	string	80	Health Domicile name from the Census that corresponds to the date returned in the StatsCensusDate column.
`DHBCode`	string	8	Specific MoH District Health Board code from

			the Census that corresponds to the date returned in the StatsCensusDate column.
DHBName	string	80	District Health Board name from the Census that corresponds to the date returned in the StatsCensusDate column.
DHBBdyAssignmentDate	date*		Date that the District Health Board code/name was assigned
AverageDeprivationDecile	integer		Numeric value (1 – 10) of Socioeconomic deprivation derived by Otago University.
AverageDeprivationQuintile	Integer		Numeric value (1 – 5) of Socioeconomic deprivation derived by Otago University.
StatsCensusDate	date		Date of the Census

Table 12, GetCAUExtraDetails call results

Examples:

SOAP	
Request	<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:esam="http://esam.co.nz/eSAMMoHAddress_v01/"> <soapenv:Header/> <soapenv:Body> <esam:GetCAUExtraDetailsRequest> <CAUId>572900</CAUId> <UniqueOrganisationCode>MyOrdgID</UniqueOrganisationCode> <UniqueUserId>MyUniqueID</UniqueUserId> <UniqueApplicationId>MyAppID</UniqueApplicationId> </esam:GetCAUExtraDetailsRequest> </soapenv:Body> </soapenv:Envelope>
Response	<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"> <soapenv:Header> <ns:BAMEvent activityID="3de311df-74fd-4cf6-82e8-08e7957e2721" xmlns:ns="http://wso2.org/ns/2010/10/bam"/> </soapenv:Header> <soapenv:Body> <ns1:GetCAUExtraDetailsResponse xmlns:ns1="http://esam.co.nz/eSAMMoHAddress_v01/"> <Header> <ResultCode>A0000</ResultCode> </Header> <CAUExtraDetails> <CAUId>572900</CAUId>

	<CAUName>Thorndon-Tinakori Road</CAUName> <DomicileCode>2077</DomicileCode> <DomicileDescription>Thorndon - Tinakori Road</DomicileDescription> <DHBCode>G00036-D</DHBCode> <DHBName>Capital and Coast District Health Board</DHBName> <DHBBdyAssignmentDate>2006-01-01+13:00</DHBBdyAssignmentDate> <AverageDeprivationDecile>4</AverageDeprivationDecile> <AverageDeprivationQuintile>2</AverageDeprivationQuintile> <StatsCensusDate>2006-01-01+13:00</StatsCensusDate> </CAUExtraDetails> </ns1:GetCAUExtraDetailsResponse> </soapenv:Body> </soapenv:Envelope>
REST	
Request	http://rest.moh.health.nz/services/eSAMMoHAddressREST_v01/GetCAUExtraDetails?CAUId=572900&UniqueApplicationId=MyApp&UniqueOrganisationCode=MyOrg&UniqueUserId=UCAD03

Response	{
	"ResultCode": "A0000",
	"CAUExtraDetails": {
	"CAUId": "572900",
	"CAUName": "Thorndon-Tinakori Road",
	"DomicileCode": "2077",
	"DomicileDescription": "Thorndon - Tinakori Road",
	"DHBCode": "G00036-D",
	"DHBName": "Capital and Coast District Health Board",
	"DHBBdyAssignmentDate": "2006-01-01+13:00",
	"AverageDeprivationDecile": "4",
	"AverageDeprivationQuintile": "2",
	"StatsCensusDate": "2006-01-01+13:00"
	}
	}

4. Testing the Web Services

Basic Connectivity Testing

Open a browser and enter the following address to access the WSDL:

`http://hid-inttest.moh.health.nz:8080/eSAMMoHAddressWS/services/eSAMMoHAddress_v01?wsdl`[1]

This will display the WSDL for the end-point. The ability to access the WSDL will verify you can connect to the service end-point and should be able to call each operation on the Address Web Services.

Standard Test Approach

The standard approach to testing and understanding the Address services is to use a pure XML-based tool such as Soap UI. Soap UI [2]is available in a number of versions, free and paid. The following explains how to use Soap UI to execute a Get Patient request (the Integration environment is pre-populated with this fake data).

- Create new project
- Create a sample request
- Populate header with your specific application, organisation and user ID.
- Execute request

Create SOAP Project

Start Soap UI and add a new project. Paste into the "Initial WSDL" box the URL for the Address Services on the integration box (you can choose to create tests and mocks here, or just add later):

`http://hid-inttest.moh.health.nz:8080/eSAMMoHAddressWS/services/eSAMMoHAddress_v01?wsdl`

This will generate template requests for all of the operations:

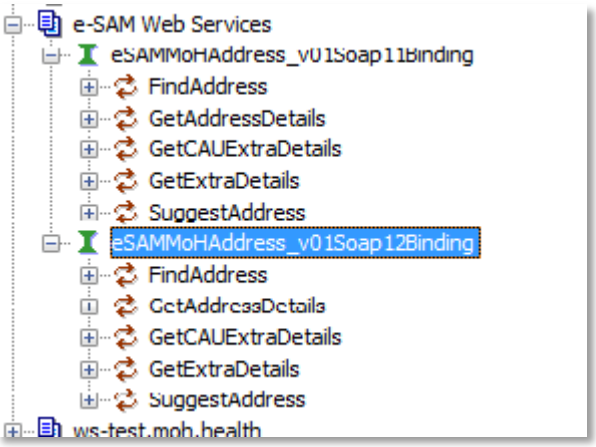


Figure 1 Soap UI Project View

SOAP UI creates a series of example requests for each operation. The next step is to create a valid request.

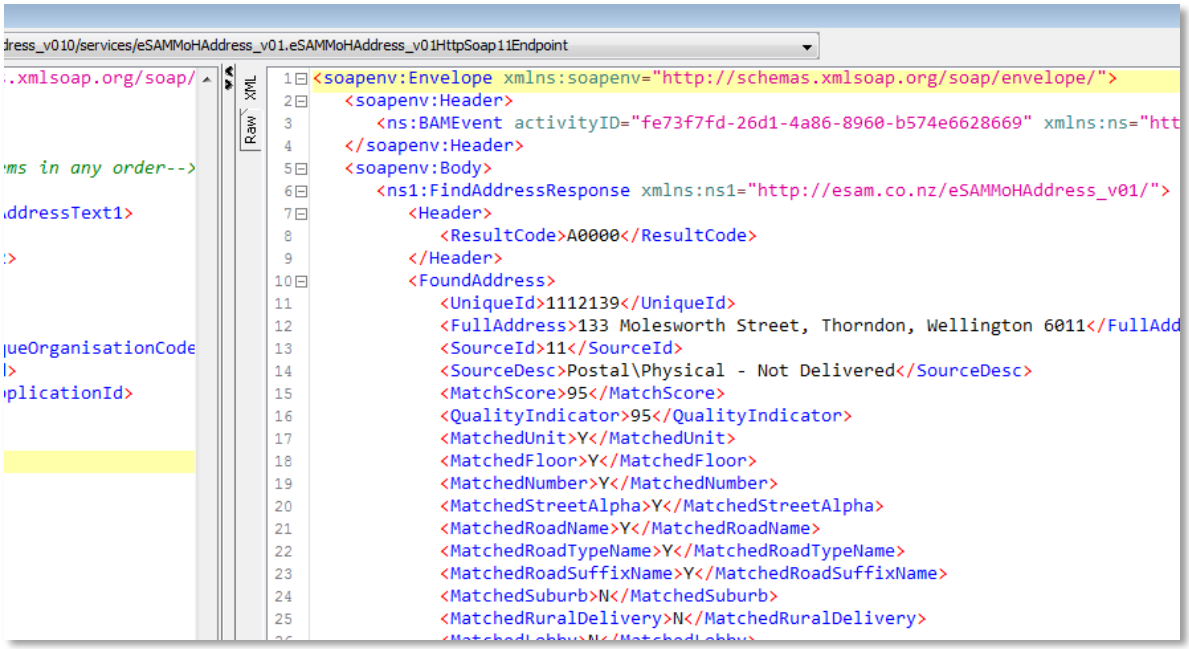
[1] This is the Integration end-point and typically only works over the Internet (depending on your network configuration).
[2] http://www.soapui.org/

Create Sample Request

The example below is the FindAddress request (looking for “133 Molesworth Street”). You will need to edit the end-point to point to the external Web Service address. (By default SOAP UI uses the address it finds within the WSDL; this is the internal IP of the server and will not work). In the example below the end-point is set to the UAT web address:

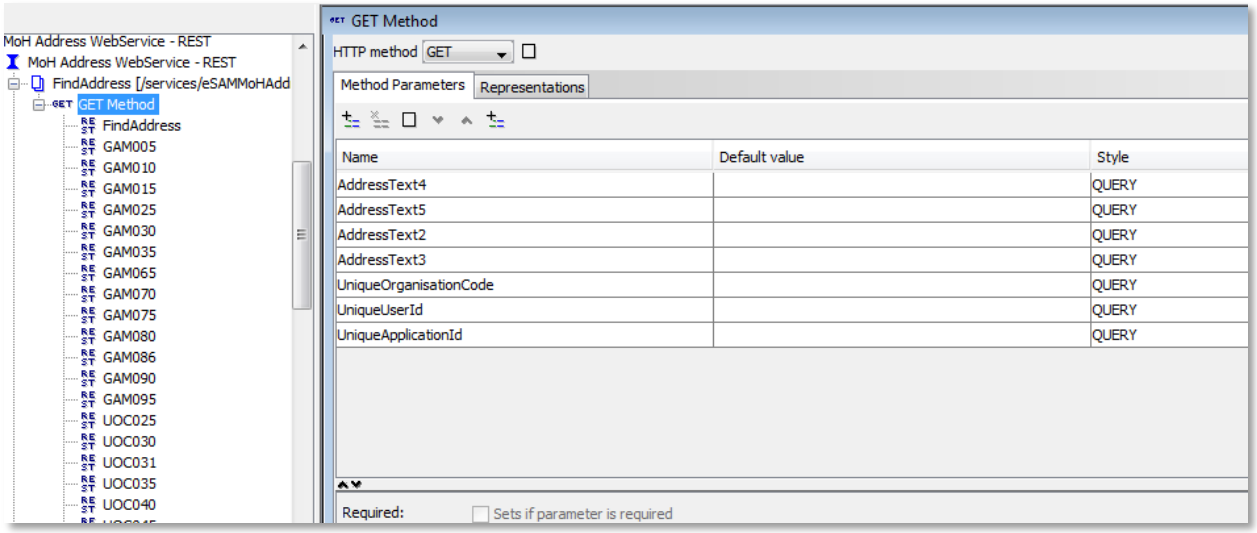
Execute request

Executing the request will return a series of address records. Use the HTTP and SOAPUI logs at the bottom of the request window to evaluate any issues.



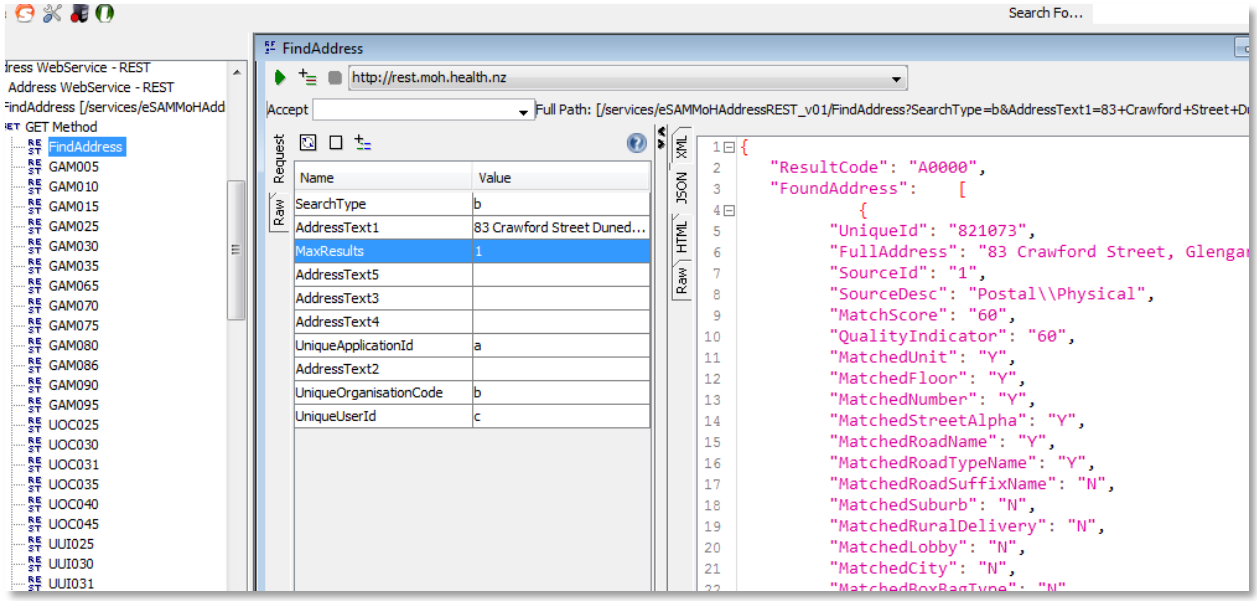
Testing REST Services

SOAP UI may also be used to test REST services. Do this by selecting "Add REST Service" in the New Project dialog box. You can set application, organisation and user ID in the template and tailor other parameters as required:



Use the HTTP log window when testing these services to gain a better understanding of how data is being transmitted. The firewall will evaluate the existence and type of data on the GET but not the data itself. For example, application ID is tested for existence but not (at present) validated against any server table.

Below is an example of creating and executing a REST FindAdress:

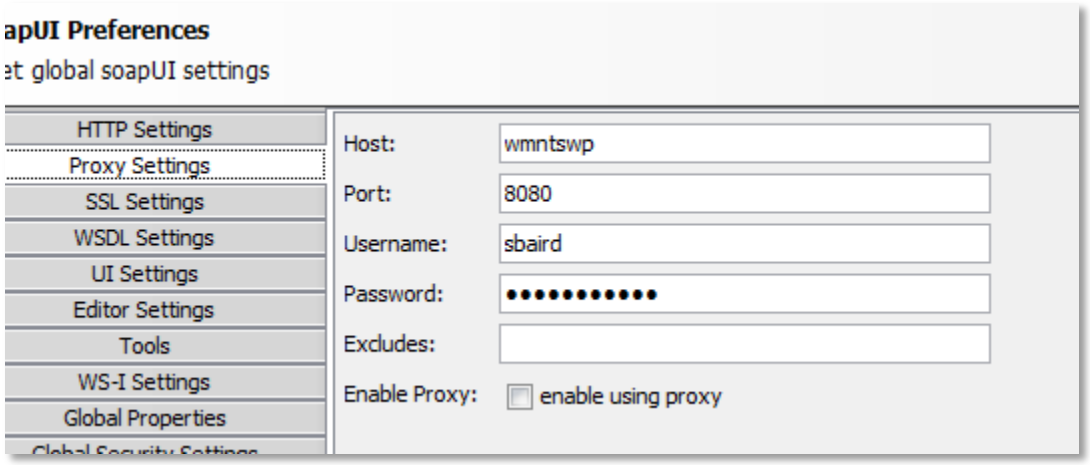


As with the SOAP requests you will need to set the end-point to correct one (that is, not the default IP-based from the WSDL).

Summary

Use SOAP UI to explore and test each service operation. Doing this will separate concerns when debugging code issues.

On a final note you may need to set your username and password if your network uses a proxy server. These settings are found under File->Preferences->Proxy Settings:



5. Integrating the Web Services

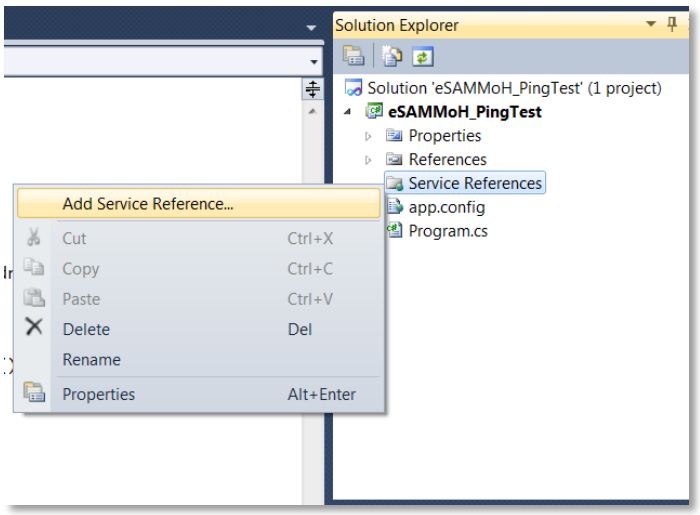
Integrating eSAM MoH Web Services into your application is relatively simple, but is dependent upon your language and development environment of choice. Integrated Development Environments (IDE) will typically provide plugins or have features that allow for the integration of SOAP and REST Web Service calls. This section will illustrate how to implement the eSAM MoH Web Services using Java and Microsoft .NET.

All of the Web Services available in this release are listed in WSDL section which contains hyperlinks to more details information about each Web Service call.

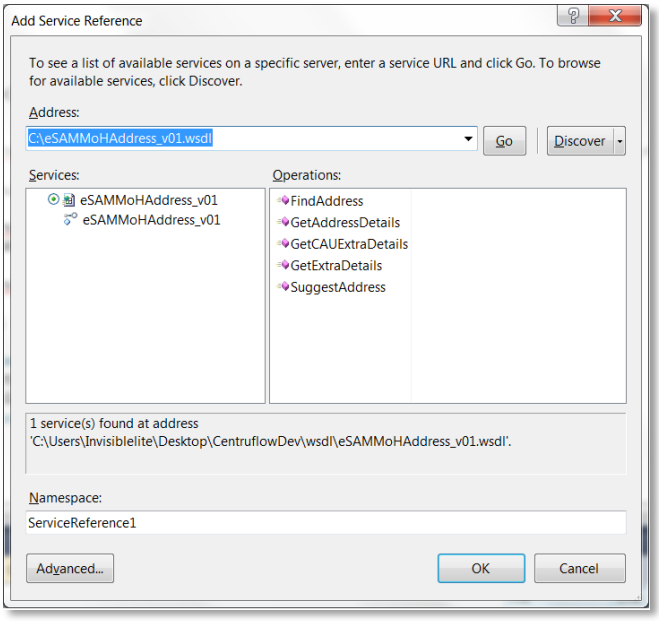
.NET Sample

Microsoft .NET provides a simple and flexible method of implementing Web Service calls through SOAP or REST methods. To implement Web Service calls, follow the following example:

In your solution explorer, add a Service Reference by right-clicking Service References and selecting it from the menu.



In the Add Service Reference window, enter the either location of the eSAMMoHAddress.WSDL file or browse to the HTTP end-point. If successful, this will load the WSDL file and show the interfaces and Web Service calls. Select the eSAMMoHAddress service from the menu on the left of the screen, and give it an appropriate name in the Namespace field.



From here, you can reference the namespace of the service you have created. The service will be a class within the namespace, eSAMMoHAddress_client, or similar.

- To create an instance of a Web service, as an object that you can call, simply instantiate the Web Service object in the namespace.

```
eSAMMoHAddress.eSAMMoHAddress_v01Client connection =
new eSAMMoHAddress.eSAMMoHAddress_v01Client();
```

- Establish the network connection using the Open() method.

```
connection.Open();
```

- Finally, to make service calls on the Web Service, you can make individual service calls using the methods that were generated when .NET processed the WSDL file. Each service call is a self-contained send and receive method. For example, we can call the suggest address service to find addresses that partially match our query string.

```
eSAMMoHAddress.AddressSuggestionType[] output;
ResultHeaderType response = connection.SuggestAddress(searchString,
"",
20,
"myorganisation",
"myuserid",
"myapplication",
out output);
```

In this example, significant parameter definitions are as follows:

- “133 Molesworth Street” is the search string.
- “myorganisation”, “myuserid” and “myapplication” are necessary fields that provide the Web Service server information regarding Web Service usage. You should fill them in according to usage guidelines.

- output is the array, passed by reference to the method for population with resulting data.

Note that the SOAP methods use pass-by-reference (`AddressSuggestionType[]` output) to return the resulting query data, and that the query response code is contained in the `ResultHeaderType`. Response codes can be used to filter out successful or failed queries in a more efficient manner than analysing the returned data.

Finally, close the connection:

```
connection.Close();
```

The following code illustrates a simple .NET application that, using the above methods, finds an address and returns a matching address to the user:

```
using Client.eSAMMoHAddress; //imported through service references

using System;

namespace Client

{
    class Example
    {
        public Example(string searchString)
        {
            try
            {
                eSAMMoHAddress.eSAMMoHAddress_v01Client connection = new
                            eSAMMoHAddress.eSAMMoHAddress_v01Client();
                connection.Open();
                eSAMMoHAddress.AddressSuggestionType[] output;
                ResultHeaderType response = connection.SuggestAddress(searchString,

                   "",

                   20,

                   "myorganisation",

                   "myuserid",

                   "myapplication",

                   out output);

                connection.Close();
                foreach (AddressSuggestionType addr in output)
                {
                    Console.WriteLine(addr.FullAddress);
                }
            }
            catch (Exception e)
            {
                Console.WriteLine(e.Message);
            }
        }
        static void Main(string[] args)
        {
            if (args.Length != 1)
            {
```

Java Sample

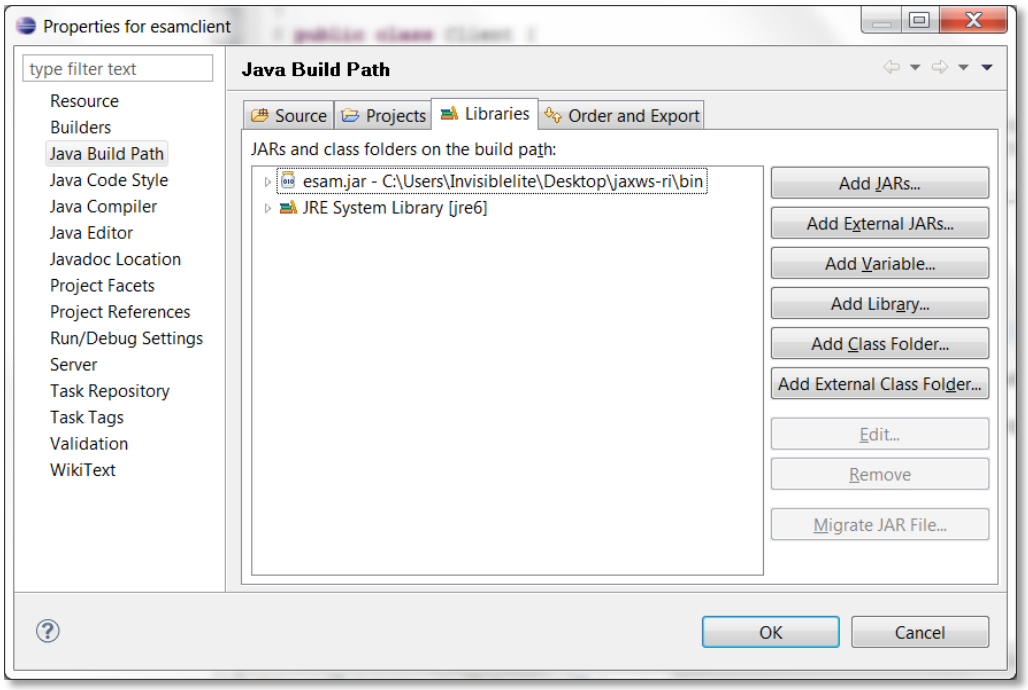
Implementation of SOAP based Web Service calls in Java is relatively similar to Microsoft .NET, but ultimately the actual code you use will depend on the method you use to import the WSDL definitions into your project code space. In this example, we will use the Eclipse IDE and a 3rd party tool, JAX-WS (available for download at http://jax-ws.java.NET/2.2.5/). Depending upon your license requirements and development tools, alternative methods of importing the WSDL definitions into your code may be preferable, including lower-level code implementations.

- Install JAX-WS to your computer by extracting the contents of the download.
- Copy your eSAMMoHAddress.WSDL file to the /bin directory of the JAX-WS installation.
- Using your computer's command line tool, navigate to the location where you have installed jax-ws, and run the necessary command to extract the contents of your WSDL file to a .jar archive.

```
wsimport.bat eSAMMoHAddress_v01.WSDL -clientjar esam.jar
```

You may need to manually populate the .jar's Manifest file to allow Eclipse to import it.

- Create a new Java Project in eclipse, and add the newly created jar archive to the build path. Your project's build path should appear similar to the image below:



- This will now allow you to reference the Web Service calls through your code. To establish a SOAP connection with the Web Service, create a new Service object and obtain the SOAPService object from that.

```
ESAMMoHAddressV01_Service service = new ESAMMoHAddressV01_Service();
ESAMMoHAddressV01 soapService = service.getSOAPOverHTTP();
```

In JAX-WS implementations, service calls are expressed as objects. You can create new request objects, which return response objects when passed through the SOAPService

```
SuggestAddressRequest request = new SuggestAddressRequest();
            request.setMaxResults(new Long(5)); //number of results
            request.setSearchFor("133 Molesworth Street");
            request.setUniqueApplicationId("myapplication");
            request.setUniqueOrganisationCode("myorganisation");
      request.setUniqueUserId("myuserid");
```

When your request is ready, send the request through the SOAPService object and receive the response in the same method.

```
SuggestAddressResponse response = soapService.suggestAddress(request);
```

The response object contains a ResultHeaderType and the resulting data, in this case a List of results.

The following code illustrates a simple Java application that, using the above steps, returns addresses that are similar to the search string.

```
import java.util.List;
import nz.co.esam.esammohaddress_v01.*;

public class Client {

        public Client() {

                ESAMMoHAddressV01_Service service = new
                                ESAMMoHAddressV01_Service();
                ESAMMoHAddressV01 soapService = service.getSOAPOverHTTP();
                SuggestAddressRequest request = new SuggestAddressRequest();
                request.setMaxResults(new Long(5)); //number of results
                request.setSearchFor("133 Molesworth Street");
                request.setUniqueApplicationId("myapplication");
                request.setUniqueOrganisationCode("myorganisation");
                request.setUniqueUserId("myuserid");

                SuggestAddressResponse response =
                                soapService.suggestAddress(request);
```

6. Appendix

Web Services Architecture

In order to provide the widest possible support for existing and emerging technologies the Health Identity services are being made available using Web Service based protocols over standard HTTP networks. A number of security layers will be deployed depending on the data being retrieved, these will range from IP based checks, secure socket layer encryption using certificates and user credential checking against a central role based directory.

Across the Connected Health network ports 80 (http) and 443 (https) are required to be open in both directions. The Web Services provided will be SOAP (Simple Object Access Protocol) based, some services will also be exposed as REST (Representational state transfer) to allow faster access to lookup services. For further details about SOAP and REST see the links below:

http://en.wikipedia.org/wiki/SOAP

http://en.wikipedia.org/wiki/Representational_state_transfer

Support

Ministry of Health National Contact Centre	Email: onlinehelpdesk@moh.govt.nz Telephone: 0800 505 125
More information about Health Identity	http://www.health.govt.nz/our-work/health-identity

Connection Endpoints

Environment	Endpoint
Integration	http://hid-inttest.moh.health.nz:8080/eSAMMoHAddressWS/services/eSAMMoHAddress_v01
UAT (Compliance)	http://ws-test.moh.health.nz/services/eSAMMoHAddress_v01
UAT (REST services)	http://rest-test.moh.health.nz/services/eSAMMoHAddressREST_v01
Production	http://ws.moh.health.nz/services/eSAMMoHAddress_v01
Production(REST services)	http://rest.moh.health.nz/services/eSAMMoHAddressREST_v01