

# Developer Centre API specification

Postage Assessment Calculation (PAC) and Postcode Search (PCS)  
Version 1.4

# Contents

1.	Overview	3
1.1	Purpose	3
1.2	Australia Post Websites	3
1.3	Document Tracking	3
2.	Accessing API	4
2.1	API Key	4
2.2	API output examples	4
2.3	XML	4
2.4	JSON	4
2.5	Error message	5
3.	API Methods	6
3.1	Data oriented method	6
3.2	Service oriented method	6
4.	Data Oriented Method API's	7
4.1	Country	7
4.2	Domestic Letter Thickness	7
4.3	Domestic Letter Weight	7
4.4	Domestic Letter Envelope Size	8
4.5	International Letter Weight	8
4.6	International Parcel Weight	8
4.7	Domestic Parcel Weight	9
4.8	Domestic Parcel Box Type	9
4.9	Domestic Parcel Box Size	9
4.10	Domestic Letter Service list	10
4.11	Domestic Parcel Service list	10
4.12	International Letter Service list	11
4.13	International Parcel Service list	11
5.	Service Oriented Method API's	13
5.1	Postcode Search API	13
5.2	Domestic Letter Postage calculation API	13
5.3	Domestic Parcel Postage calculation API	14
5.4	International Letter Postage calculation API	14
5.5	International Parcel Postage calculation API	15
6.	API Examples	16
6.1	Data oriented API examples	16
6.2	Service oriented API examples	16
7.	API Usage Example	18
7.1	Flow of events	18
7.2	Getting User input	18
7.3	Get Service types	18
7.4	Calculate Postage Cost	21

# 1. Overview

---

## 1.1 Purpose

This document is intended to provide customers with the ability to integrate Australia Post services within their website. The specific services covered by this document are:

1. Postage Assessment Calculation (PAC). This service is used to calculate the cost of posting a letter or parcel either within Australia or overseas.
2. Postcode Search (PCS). This service is used to look up Australian postcodes.

## 1.2 Australia Post Websites

Production website URL: <https://auspost.com.au>

Non-production website URL: <https://test.npe.auspost.com.au>

## 1.3 Document Tracking

Version	Date	Description
1.0	August 2011	Initial Draft
1.1	September 2011	Added reference data oriented services
1.2	September 2011	Internal review
1.3	September 2011	Released to production
1.4	January 2013	Changed test endpoints for API's

## 2. Accessing API

---

### 2.1 API Key

A unique 32 character API key is used to authorise inbound requests to the APIs. To obtain an API key please visit <http://auspost.com.au/devcentre> and follow the link to create your Australia Post ID. After registration has been successfully completed you will be emailed an API key along with links to documentation specifying how to call the APIs and the terms and conditions associated with using the APIs.

Australia Post requires customers to use these APIs via HTTP GET over SSL (Secure Sockets Layer) protocol so as to protect the API keys. The required design pattern is for customers to use **AUTH-KEY** and SSL for both the production and non-production websites. The API keys cannot be used in address line calls and **MUST** be sent within the **HTTP HEADERS**.

#### 2.1.1 Non-production API key

During integration phase of your project, you might want to use non-production APIs available at <https://test.npe.auspost.com.au>, rather than live APIs.

The following API key is required to access these APIs:  
28744ed5982391881611cca6cf5c2409

### 2.2 API output examples

For both the PAC and PCS APIs two output formats are provided. These are XML and JSON. The default output option is XML.

### 2.3 XML

```
<localities>
  <locality>
    <category>Delivery Area</category>
    <id>7348</id>
    <latitude>-36.755513</latitude>
    <location>BENDIGO</location>
    <longitude>144.284263</longitude>
    <postcode>3550</postcode>
    <state>VIC</state>
  </locality>
  <locality>
    <category>Delivery Area</category>
    <id>7349</id>
    <latitude>-36.755513</latitude>
    <location>BENDIGO SOUTH</location>
    <longitude>144.284263</longitude>
    <postcode>3550</postcode>
    <state>VIC</state>
  </locality>
</localities>
```

### 2.4 JSON

```
{
  "localities" :
  {
    "locality" : [
      {
        "category" : "Delivery Area",
```

```

        "id" : 7348,
        "latitude" : -36.755513000000001,
        "location" : "BENDIGO",
        "longitude" : 144.28426300000001,
        "postcode" : 3550,
        "state" : "VIC"
    },
    {
        "category" : "Delivery Area",
        "id" : 7349,
        "latitude" : -36.755513000000001,
        "location" : "BENDIGO SOUTH",
        "longitude" : 144.28426300000001,
        "postcode" : 3550,
        "state" : "VIC"
    }
]
}

```

## 2.5 Error message

The following format will be presented whenever an error condition occurs.

### 2.5.1 XML error example

```

<error>
  <errorMessage>Please enter Suburb, Town, City or Postcode.</errorMessage>
</error>

```

### 2.5.2 JSON error example

```

{
  "error":
  {
    "errorMessage": "Please enter Suburb, Town, City or Postcode."
  }
}

```

## 3. API Methods

---

Two types of API methods are provided. These are data oriented methods and service oriented methods.

### 3.1 Data oriented method

This method is used to return the available parameters for the API such as a list of countries that can be posted to.

Data oriented methods:

1. Country
2. Domestic Letter Thickness
3. Domestic Letter Weight
4. Domestic Letter Envelope Size
5. International Letter Weight
6. International Parcel Weight
7. Domestic Parcel Weight
8. Domestic Parcel Type
9. Domestic Parcel Size
10. Domestic Letter Service list
11. Domestic Parcel Service list
12. International Letter Service list
13. International Parcel Service list

### 3.2 Service oriented method

This method is used to provide the API's functional capabilities such as postcode searching and postage calculations.

Service oriented methods:

1. Postcode Search
2. Domestic Letter Postage calculation
3. Domestic Parcel Postage calculation
4. International Letter Postage calculation
5. International Parcel Postage calculation

## 4. Data Oriented Method API's

---

This section provides the syntax to use the data oriented methods of the PAC and PCS API's. Service list methods are returning all possible postage services applicable to the parameters passed into these methods. These services are used by Service Oriented Method API's later as input values for postage calculation.

### 4.1 Country

Returns a list of countries which are accepted by the PAC API.

Request must be a HTTPS GET.

**URL:**

`https://auspost.com.au/api/postage/country.{format}`

**Formats:**

xml, json (Default to xml if format not provided)

**Requires Authorisation:**

True (API Key)

### 4.2 Domestic Letter Thickness

Returns a list domestic letter thickness values accepted by the PAC API.

Request must be a HTTPS GET.

**URL:**

`https://auspost.com.au/api/postage/letter/domestic/thickness.{format}`

**Formats:**

xml, json (Default to xml if format not provided)

**Requires Authorisation:**

True (API Key)

### 4.3 Domestic Letter Weight

Returns a list of domestic letter weight values accepted by the PAC API.

Request must be a HTTPS GET.

**URL:**

`https://auspost.com.au/api/postage/letter/domestic/weight.{format}`

**Formats:**

xml, json (Default to xml if format not provided)

**Requires Authorisation:**

True (API Key)

## 4.4 Domestic Letter Envelope Size

Returns a list of available domestic envelope type values that are offered by Australia Post. The developer is not forced to send these values to the PAC API, but it may help facilitate with building a PAC user interface.

Request must be a HTTPS GET.

**URL:**

`https://auspost.com.au/api/postage/letter/domestic/size.{format}`

**Formats:**

xml, json (Default to xml if format not provided)

**Requires Authorisation:**

True (API Key)

**Return Value:**

Returns a list of letter sizes, with value set to 'x' separated list of width and height values

```
<size>
  <code>AUS_LETTER_SIZE_DL</code>
  <name>DL (110 x 220 mm)</name>
  <value>110x220</value>
</size>
```

## 4.5 International Letter Weight

Returns a list of international letter weight values that are accepted by the PAC API.

Request must be a HTTPS GET.

**URL:**

`https://auspost.com.au/api/postage/letter/international/weight.{format}`

**Formats:**

xml, json (Default to xml if format not provided)

**Requires Authorisation:**

True (API Key)

## 4.6 International Parcel Weight

Returns a list of international parcel weight values that are accepted by the PAC API.

Request must be a HTTPS GET.

**URL:**

`https://auspost.com.au/api/postage/parcel/international/weight.{format}`

**Formats:**

xml, json (Default to xml if format not provided)

**Requires Authorisation:**

True (API Key)

## 4.7 Domestic Parcel Weight

Returns a list of available domestic parcel weight values that are accepted by the PAC API.

Request must be a HTTPS GET.

**URL:**

`https://auspost.com.au/api/postage/parcel/domestic/weight.{format}`

**Formats:**

xml, json (Default to xml if format not provided)

**Requires Authorisation:**

True (API Key)

## 4.8 Domestic Parcel Box Type

Returns a list of available parcel type values (Boxed items or Satchels). The developer is not forced to send these values to the PAC API, but it may help facilitate with building a PAC user interface.

Request must be a HTTPS GET.

**URL:**

`https://auspost.com.au/api/postage/parcel/domestic/type.{format}`

**Formats:**

xml, json (Default to xml if format not provided)

**Requires Authorisation:**

True (API Key)

## 4.9 Domestic Parcel Box Size

Returns a list of available parcel box size values that are offered by Australia Post. The developer is not forced to send these values to the PAC API, but it may help facilitate with building a PAC user interface.

Request must be a HTTPS GET.

**URL:**

`https://auspost.com.au/api/postage/parcel/domestic/size.{format}`

**Formats:**

xml, json (Default to xml if format not provided)

**Requires Authorisation:**

True (API Key)

**Return Value:**

Returns list of parcel sizes, with value set to 'x' separated list of width, height and depth values

```
<size>
  <code>AUS_PARCEL_TYPE_BOXED_OTH</code>
  <name>My own box</name>
</size>
<size>
  <code>AUS_PARCEL_TYPE_BOXED_BC</code>
  <name>BC (22.0 x 16.0 x 7.7)</name>
  <value>22.0x16.0x7.7</value>
</size>
```

## 4.10 Domestic Letter Service list

Request must be a HTTPS GET.

**URL:**

<https://auspost.com.au/api/postage/letter/domestic/service.{format}>

**Formats:**

xml, json (Default to xml if format not provided)

**Requires Authorisation:**

True (API Key)

**Parameters:**

1. [length]: **Required**. The length of the letter in mm. (max 260mm) Input value will be round up to the next integer if decimal value is specified.
2. [width]: **Required**. The width of the letter in mm. (max 360mm) Input value will be round up to the next integer if decimal value is specified.
3. [thickness]: **Required**. The thickness of the letter in mm. (max 20mm) Input value will be round up to the next integer if decimal value is specified.
4. [weight]: **Required**. The weight of the letter in grams (max weight is 500g). Input value will be round up to the next integer if decimal value is specified.

## 4.11 Domestic Parcel Service list

Request must be a HTTPS GET.

**URL:**

<https://auspost.com.au/api/postage/parcel/domestic/service.{format}>

**Formats:**

xml, json (Default to xml if format not provided)

**Requires Authorisation:**

True (API Key)

**Parameters:**

1. [from\_postcode]: **Required**. The from postcode
2. [to\_postcode]: **Required**. The to postcode
3. [length]: **Required**. The length of the parcel in cm. Input value will be round up to the next mm if decimal value is specified.
4. [width]: **Required**. The width of the parcel in cm. Input value will be round up to the next mm if decimal value is specified.
5. [height]: **Required**. The height of the parcel in cm. Input value will be round up to the next mm if decimal value is specified.
6. [weight]: **Required**. The weight of the parcel in kg.

## 4.12 International Letter Service list

Request must be a HTTPS GET.

**URL:**

`https://auspost.com.au/api/postage/letter/international/service.{format}`

**Formats:**

xml, json (Default to xml if format not provided)

**Requires Authorisation:**

True (API Key)

**Parameters:**

1. [country\_code]: **Required**. The country code (code is provided by Country data oriented service)
2. [weight]: Optional. The letter weight in grams (max weight is 500g, if not specified, assume it is a postcard). Input value will be round up to the next integer if decimal value is specified.

## 4.13 International Parcel Service list

Request must be a HTTPS GET.

**URL:**

`https://auspost.com.au/api/postage/parcel/international/service.{format}`

**Formats:**

xml, json (Default to xml if format not provided)

**Requires Authorisation:**

True (API Key)

**Parameters:**

1. [country\_code]: **Required**. The country code (code is provided by Country data oriented service)
2. [weight]: **Required**. The weight of the parcel in kg

## 5. Service Oriented Method API's

---

This section provides the syntax to use the service oriented methods of the PAC and PCS API's.

### 5.1 Postcode Search API

Search the collection of Postcode records for potential matches. The search criterion is location name or postcode with an optional state filter. Request must be a HTTPS GET.

**URL:**

`https://auspost.com.au/api/postcode/search.{format}`

**Formats:**

xml, json (Default to xml if format not provided)

**Requires Authorisation:**

True (API Key)

**Parameters:**

1. [q]: **Required**. The search criterion used to drive the search query. This parameter will hold either the location name or postcode
2. [state]: Optional. Used to filter possible search results. Must be formatted using an Auspost accepted abbreviation (i.e. VIC for Victoria, QLD for Queensland)
3. [excludePostBoxFlag]: Optional. Set value to true, if post boxes should be excluded from search results. That is useful for PAC.

### 5.2 Domestic Letter Postage calculation API

Request must be a HTTPS GET.

**URL:**

`https://auspost.com.au/api/postage/letter/domestic/calculate.{format}`

**Formats:**

xml, json (Default to xml if format not provided)

**Requires Authorisation:**

True (API Key)

**Parameters:**

1. [service\_code]: **Required**. The service code of the letter (Code retrieved from Domestic Letter Service and Option data oriented method)
2. [weight]: **Required**. The weight of the letter in grams (max weight is 500g). Input value will be round up to the next integer if decimal value is specified.
3. [option\_code]: Optional. The option code of the letter. (Code retrieved from Domestic Letter Service and Option data oriented method)

4. [suboption\_code]: Optional. The sub-option code of the letter, multi-valued. (Code retrieved from Domestic Letter Service and Option data oriented method)
5. [extra\_cover]: Optional. The monetary value of the extra cover, represented in dollar value only. Integer only, no dollar sign, decimal or comma accepted. This parameter is only applicable when the suboption\_code is AUS\_SERVICE\_OPTION\_EXTRA\_COVER

### 5.3 Domestic Parcel Postage calculation API

Request must be a HTTPS GET.

#### URL:

`https://auspost.com.au/api/postage/parcel/domestic/calculate.{format}`

#### Formats:

xml, json (Default to xml if format not provided)

#### Requires Authorisation:

True (API Key)

#### Parameters:

1. [from\_postcode]: **Required**. From postcode
2. [to\_postcode]: **Required**. To postcode
3. [length]: **Required**. The length of the parcel in cm. Input value will be round up to the next mm if decimal value is specified.
4. [width]: **Required**. The width of the parcel in cm. Input value will be round up to the next mm if decimal value is specified.
5. [height]: **Required**. The height of the parcel in cm. Input value will be round up to the next mm if decimal value is specified.
6. [weight]: **Required**. The weight of the parcel in kg
7. [service\_code]: **Required**. The service code of the parcel (Code retrieved from Domestic Parcel Service and Option data oriented method)
8. [option\_code]: Optional. The option code of the parcel. (Code retrieved from Domestic Parcel Service and Option data oriented method)
9. [suboption\_code]: Optional. The sub-option code of the parcel, multi-valued. (Code retrieved from Domestic Parcel Service and Option data oriented method)
10. [extra\_cover]: Optional. The monetary value of the extra cover, represented in dollar value only. Integer only, no dollar sign, decimal or comma accepted. This parameter is only applicable when the suboption\_code is AUS\_SERVICE\_OPTION\_EXTRA\_COVER

### 5.4 International Letter Postage calculation API

Request must be a HTTPS GET.

#### URL:

`https://auspost.com.au/api/postage/letter/international/calculate.{format}`

#### Formats:

xml, json (Default to xml if format not provided)

#### Requires Authorisation:

True (API Key)

**Parameters:**

1. [country\_code]: **Required**. Country code (Code retrieved from Country data oriented method)
2. [weight]: **Optional**. The letter weight in grams (max weight is 500g , if not specified, assume it is a postcard). Input value will be round up to the next integer if decimal value is specified.
3. [service\_code]: **Required**. The service code of the letter (Code retrieved from International Letter Service and Option data oriented method)
4. [option\_code]: **Optional**. The option code of the letter, multi-valued. (Code retrieved from International Letter Service and Option data oriented method)
5. [suboption\_code]: **Optional**. The sub-option code of the letter. (Code retrieved from International Letter Service and Option data oriented method)
6. [extra\_cover]: **Optional**. The monetary value of the extra cover, represented in dollar value only. Integer only, no dollar sign, decimal or comma accepted. This parameter is only applicable when the option\_code is INTL\_SERVICE\_OPTION\_EXTRA\_COVER.

## 5.5 International Parcel Postage calculation API

Request must be a HTTPS GET.

**URL:**

`https://auspost.com.au/api/postage/parcel/international/calculate.{format}`

**Formats:**

xml, json (Default to xml if format not provided)

**Requires Authorisation:**

True (API Key)

**Parameters:**

1. [country\_code]: **Required**. Country code (Code retrieved from Country data oriented method)
2. [weight]: **Required**. The weight of the parcel in kg (max weight is 20kg)
3. [service\_code]: **Required**. The service code of the letter (Code retrieved from International Parcel Service and Option data oriented method)
4. [option\_code]: **Optional**. The option code of the parcel, multi-valued. (Code retrieved from International Parcel Service and Option data oriented method)
5. [extra\_cover]: **Optional**. The monetary value of the extra cover, represented in dollar value only. Integer only, no dollar sign, decimal or comma accepted. This parameter is only applicable when the option\_code is INTL\_SERVICE\_OPTION\_EXTRA\_COVER.

## 6. API Examples

---

This section provides examples of each data oriented and service oriented API.

### 6.1 Data oriented API examples

These examples return results in JSON format, however note that XML format is also permitted.

Request type: **HTTPS GET**  
Header: **AUTH-KEY:** {Use your personal 32 character API key}

**Country:**

<https://auspost.com.au/api/postage/country.json>

**Domestic letter service list:**

<https://auspost.com.au/api/postage/letter/domestic/service.json?length=114&width=162&thickness=4&weight=20>

**Domestic parcel service list:**

[https://auspost.com.au/api/postage/parcel/domestic/service.json?from\\_postcode=3000&to\\_postcode=4000&length=10&height=10&width=10&weight=0.5](https://auspost.com.au/api/postage/parcel/domestic/service.json?from_postcode=3000&to_postcode=4000&length=10&height=10&width=10&weight=0.5)

**International letter service list:**

[https://auspost.com.au/api/postage/letter/international/service.json?country\\_code=AF&weight=20](https://auspost.com.au/api/postage/letter/international/service.json?country_code=AF&weight=20)

**International parcel service list:**

[https://auspost.com.au/api/postage/parcel/international/service.json?country\\_code=AF&weight=10](https://auspost.com.au/api/postage/parcel/international/service.json?country_code=AF&weight=10)

### 6.2 Service oriented API examples

These examples return results in XML format, however note that JSON format is also permitted.

Request type: **HTTPS GET**  
Header: **AUTH-KEY:** {Use your personal 32 character API key}

**Postcode search:**

<https://auspost.com.au/api/postcode/search?q=melbourne>

**Domestic letter postage calculation** (Please note how multi-valued suboption\_code is passed to the API call):

[https://auspost.com.au/api/postage/letter/domestic/calculate?weight=500&service\\_code=AUS\\_LETTER\\_REGULAR\\_LARGE&option\\_code=AUS\\_SERVICE\\_OPTION\\_REGISTERED\\_POST&suboption\\_code=AUS\\_SERVICE\\_OPTION\\_DELIVERY\\_CONFIRMATION&suboption\\_code=AUS\\_SERVICE\\_OPTION\\_EXTRA\\_COVER&extra\\_cover=3000](https://auspost.com.au/api/postage/letter/domestic/calculate?weight=500&service_code=AUS_LETTER_REGULAR_LARGE&option_code=AUS_SERVICE_OPTION_REGISTERED_POST&suboption_code=AUS_SERVICE_OPTION_DELIVERY_CONFIRMATION&suboption_code=AUS_SERVICE_OPTION_EXTRA_COVER&extra_cover=3000)

**Domestic parcel postage calculation:**

[https://auspost.com.au/api/postage/parcel/domestic/calculate?from\\_postcode=3000&to\\_postcode=4000&length=10&width=10&height=10&weight=10&service\\_code=AUS\\_PARCEL\\_REGULAR&option\\_code=AUS\\_SERVICE\\_OPTION\\_REGISTERED\\_POST&suboption\\_code=AUS\\_SERVICE\\_OPTION\\_EXTRA\\_COVER&extra\\_cover=3000](https://auspost.com.au/api/postage/parcel/domestic/calculate?from_postcode=3000&to_postcode=4000&length=10&width=10&height=10&weight=10&service_code=AUS_PARCEL_REGULAR&option_code=AUS_SERVICE_OPTION_REGISTERED_POST&suboption_code=AUS_SERVICE_OPTION_EXTRA_COVER&extra_cover=3000)

**International letter postage calculation:**

[https://auspost.com.au/api/postage/letter/international/calculate?country\\_code=US&weight=10&service\\_code=INTL\\_SERVICE\\_ECI\\_D&option\\_code=INTL\\_SERVICE\\_OPTION\\_EXTRA\\_COVER&extra\\_cover=2000](https://auspost.com.au/api/postage/letter/international/calculate?country_code=US&weight=10&service_code=INTL_SERVICE_ECI_D&option_code=INTL_SERVICE_OPTION_EXTRA_COVER&extra_cover=2000)

**International parcel postage calculation** (Please note how multi-valued option\_code is passed to API call):

[https://auspost.com.au/api/postage/parcel/international/calculate?country\\_code=TG&service\\_code=INTL\\_SERVICE\\_AIR\\_MAIL&option\\_code=INTL\\_SERVICE\\_OPTION\\_EXTRA\\_COVER&option\\_code=INTL\\_SERVICE\\_OPTION\\_CONFIRM\\_DELIVERY&weight=10&extra\\_cover=2000](https://auspost.com.au/api/postage/parcel/international/calculate?country_code=TG&service_code=INTL_SERVICE_AIR_MAIL&option_code=INTL_SERVICE_OPTION_EXTRA_COVER&option_code=INTL_SERVICE_OPTION_CONFIRM_DELIVERY&weight=10&extra_cover=2000)

## 7. API Usage Example

---

This section provides the developer with a usage example of the PAC API and explains how to read and process the returned data (xml or json). We will go into some detail on how the Domestic Letter API methods have to be used in order to calculate domestic letter postage costs.

### 7.1 Flow of events

In order to calculate postage costs the API has to be supplied with item weight, selected service type and the service options the user wishes to have (for example delivery confirmation, or providing insurance cover on item, etc...)

The calculation is done in three steps:

1. Get letter size input from the user
2. Generate service type and service options available for that letter size
3. Get service type and service options input from the user and calculate the postage cost

### 7.2 Getting User input

As per the Domestic Letter Service list API method the application has to query user input with the following parameters:

- Length
- Width
- Thickness
- Weight

Imagine standard letter sizes have been selected by the user, ie: length = 110, width = 220, thickness = 4 and weight = 15

### 7.3 Get Service types

Based on the user input, the following Data Oriented Method API call has to be placed:

<https://auspost.com.au/api/postage/letter/domestic/service.xml?length=110&width=220&thickness=4&weight=15>

The API will do validation of the input parameters and provided no problems are found will return the XML. If validation fails at any stage the API will return a standard XML error as defined in the 2.5 Error message section of this document with a clear error message text that details the validation failure.

#### 7.3.1 Service type XML result

```
<services>
  <service>
    <code>AUS_LETTER_REGULAR_SMALL</code>
    <name>Small Letter</name>
    <price>0.60</price>
    <max_extra_cover>5000</max_extra_cover>
    <options>
      <option>
        <code>AUS_SERVICE_OPTION_STANDARD</code>
        <name>Standard Service</name>
      </option>
      <option>
        <code>AUS_SERVICE_OPTION_REGISTERED_POST</code>
        <name>Registered Post</name>
        <suboptions>
          <option>
            <code>AUS_SERVICE_OPTION_DELIVERY_CONFIRMATION</code>
```

```

        <name>Delivery Confirmation</name>
    </option>
    <option>
        <code>AUS_SERVICE_OPTION_PERSON_TO_PERSON</code>
        <name>Person to Person</name>
    </option>
    <option>
        <code>AUS_SERVICE_OPTION_EXTRA_COVER</code>
        <name>Extra Cover</name>
    </option>
    </suboptions>
</option>
<option>
    <code>AUS_SERVICE_OPTION_COD_POSTAGE_FEES</code>
    <name>C.O.D - Postage & Fees</name>
</option>
<option>
    <code>AUS_SERVICE_OPTION_COD_MONEY_COLLECTION</code>
    <name>C.O.D - Money Collection, Postage & Fees</name>
</option>
</options>
</service>
<service>
    <code>AUS_LETTER_PRIORITY</code>
    <name>Priority Letter</name>
    <price>2.60</price>
</service>
<service>
    <code>AUS_LETTER_EXPRESS_SMALL</code>
    <name>Express Post</name>
    <price>5.05</price>
</service>
</services>

```

Given the nature of the application (iPhone, Android or JS Client), this XML result can be processed and rendered like the screenshot below:



Figure 1 Sample domestic letter service list

Please note, whenever the 'Extra Cover' option is returned by Service List API, your application has to provide the user with a text field input box to accept extra cover value parameter.

### 7.3.2 Service type JSON result

If you need to obtain JSON data from the service just modify the URL above to get service.json, rather than service.xml. The JSON result set is outlined below:

```

{ "services" : {
  "service" : [
    {
      "code" : "AUS_LETTER_REGULAR_SMALL",
      "max_extra_cover" : 5000,
      "name" : "Small Letter",
      "price" : "0.60"
      "options" : {
        "option" : [
          {
            "code" : "AUS_SERVICE_OPTION_STANDARD",
            "name" : "Standard Service"
          },
          {
            "code" : "AUS_SERVICE_OPTION_REGISTERED_POST",
            "name" : "Registered Post",
            "suboptions" : {
              "option" : [
                {
                  "code" : "AUS_SERVICE_OPTION_DELIVERY_CONFIRMATION",
                  "name" : "Delivery Confirmation"
                },
                {
                  "code" : "AUS_SERVICE_OPTION_PERSON_TO_PERSON",
                  "name" : "Person to Person"
                },
                {
                  "code" : "AUS_SERVICE_OPTION_EXTRA_COVER",
                  "name" : "Extra Cover"
                }
              ]
            }
          }
        ]
      }
    },
    {
      "code" : "AUS_SERVICE_OPTION_COD_POSTAGE_FEES",
      "name" : "C.O.D - Postage & Fees"
    },
    {
      "code" : "AUS_SERVICE_OPTION_COD_MONEY_COLLECTION",
      "name" : "C.O.D - Money Collection, Postage & Fees"
    }
  ],
  {
    "code" : "AUS_LETTER_PRIORITY",
    "name" : "Priority Letter",
    "price" : "2.60"
  },
  {
    "code" : "AUS_LETTER_EXPRESS_SMALL",
    "name" : "Express Post",
    "price" : 5.05
  }
]
}

```

## 7.4 Calculate Postage Cost

The example below outlines the following scenario:

- a. Delivering of Regular Small letter
- b. via Registered Post service type
- c. with Insurance for \$1000.00 and
- d. with Delivery confirmation.

Provided the user has selected the desired service type and optionally selected service options the following Service Oriented Method API call has to be placed:

```
https://auspost.com.au/api/postage/letter/domestic/calculate.xml?suboption_code=AUS_SERVICE_OPTION_DELIVERY_CONFIRMATION&suboption_code=AUS_SERVICE_OPTION_EXTRA_COVER&option_code=AUS_SERVICE_OPTION_REGISTERED_POST&weight=15&service_code=AUS_LETTER_REGULAR_SMALL&extra_cover=1000
```

Please notice how multi-valued suboption code parameters are passed to the API to ensure optional Delivery Confirmation and Extra cover service sub options are displayed.

The API will do validation of input parameters and provided that no problems are found will return a result set of XML. If validation fails at any stage the API will return a standard XML error as defined in section 2.5 Error message section of this document with clear error message text that details the validation failure.

### 7.4.1 Postage Calculation XML result

```
<postage_result>
  <service>Small Letter</service>
  <delivery_time>
    Depending on the distance your article has to travel,
    it may take between 1 to 4 business days to reach
    its destination*
  </delivery_time>
  <total_cost>18.70</total_cost>
  <costs>
    <cost>
      <cost>0.60</cost>
      <item>Small Letter</item>
    </cost>
    <cost>
      <cost>3.05</cost>
      <item>Registered Post</item>
    </cost>
    <cost>
      <cost>2.00</cost>
      <item>Delivery Confirmation</item>
    </cost>
    <cost>
      <cost>13.05</cost>
      <item>Extra Cover</item>
    </cost>
  </costs>
</postage_result>
```

### 7.4.2 Postage Calculation JSON result

```
{
  "postage_result" :
  {
```

```

    "delivery_time" : "Depending on the distance your article has to travel, it may
take between 1 to 4 business days to reach its destination*",
    "service" : "Small Letter",
    "total_cost" : "18.70",
    "costs" :
    {
        "cost" :
        [
            {
                "cost" : "0.60",
                "item" : "Small Letter"
            },
            {
                "cost" : 3.0499999999999998,
                "item" : "Registered Post"
            },
            {
                "cost" : "2.00",
                "item" : "Delivery Confirmation"
            },
            {
                "cost" : 13.050000000000001,
                "item" : "Extra Cover"
            }
        ]
    }
}

```