**Australia Post API****Functional Specification**

Abhi A  
Last Updated: March 1, 2021

**Overview**

**AusPostAPI** is a REST API that tells people about the suburb they belong to depending on their postcode and vice versa. It also allows the admin to add new suburb and postcode combinations over the web.

**This spec is not, by any stretch of the imagination, complete.** All of the wording will need to be revised several times before it is finalized.

This spec does not discuss the algorithms used by the REST API, which will be discussed elsewhere. It merely outlines how a REST API will be used by end clients (for query) and admins (to add), postcode-suburb information.

**Non Goals**

This version will *not* support the following features:

* Connection to an external database to persist the information. Spring derby embedded database will be used to do the same.
* Mobile client.

**REST Specification**

**Technical Note**

Requirements

* An API that allows mobile clients to retrieve the suburb information by postcode. [Implemented]
* An API that allows mobile clients to retrieve a postcode given a suburb name [Implemented]
* A secured API to add new suburb and postcode combinations [Implemented and Secured]
* Some form of persistence [Implemented]

**Technical Note**

* Spring MVC design pattern
* Spring Data JPA for persistence
* Spring Security (Basic Authentication)
* Apache Derby embedded database
* REST API
* Resources: Postcode, Suburb
* URLs:

GET /postcode/ {suburb} 🡪 gets the postcode for a suburb

Example: GET /postcode/ ABEY WA

Output:

[

{

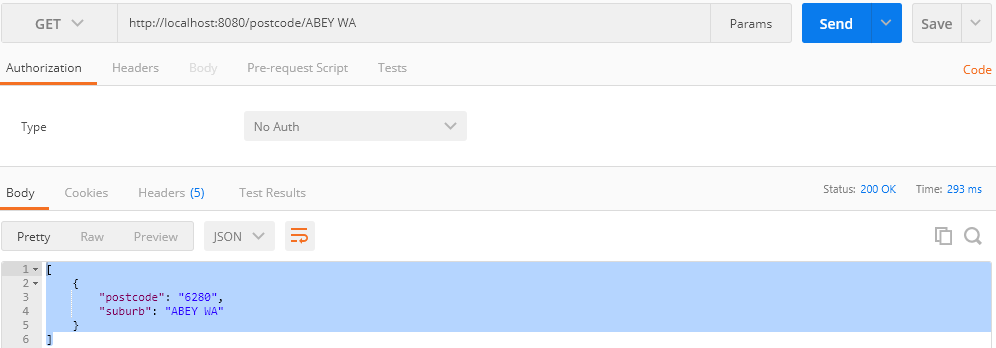
"postcode": "6280",

"suburb": "ABEY WA"

}

]

Postman snapshot below:



GET /suburb/ {postcode} 🡪 gets the suburb for a postcode

Example: GET /suburb/2850

Output:

[

{

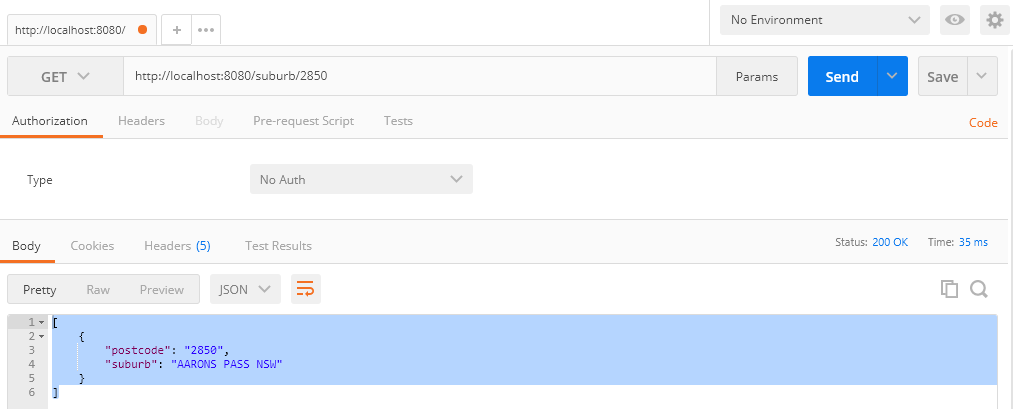
"postcode": "2850",

"suburb": "AARONS PASS NSW"

}

]

Postman snapshot below:



POST /admin/postsub 🡪 Creates new postcode-suburb combo

Example: POST /admin/postsub

POST body:

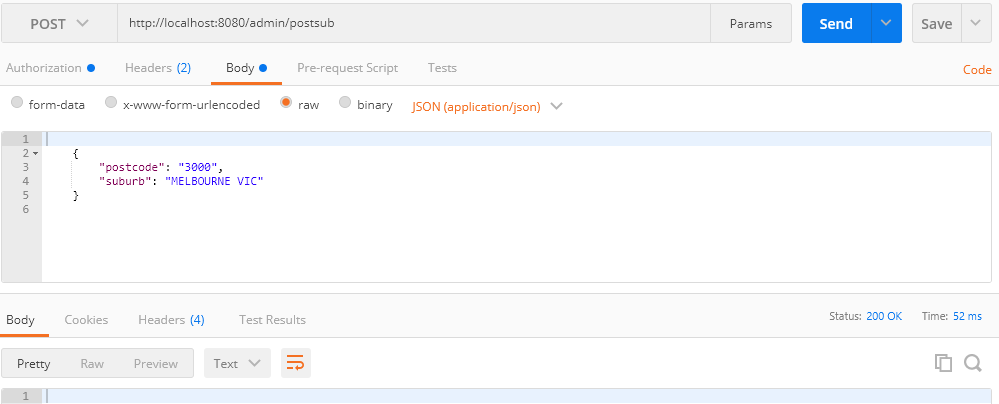
{

"postcode": "3000",

"suburb": "MELBOURNE VIC"

}

Postman snapshot below:



Note: Spring Security Basic Authentication is on the POST URL. So, user=admin and password=admin are needed in the REST API clients like POSTMAN to add postcode-suburb information into the database.

**Controller**

* PostcodeSuburbController.java handles requests that:
* Gets the postcode for a suburb
* Gets the suburb for a postcode
* Creates new postcode-suburb combo

**Business Services**

* PostcodeService (PostcodeService.java returns postcode for suburb information)
* SuburbService (SuburbService.java returns suburb for postcode information)
* AdminService (AdminService.java creates new suburb - postcode). It is a secured service layer.

**Data Service**

* PostcodeSuburbRepository.java

**Entity**

* PostcodeSuburb.java represents the entity class that maps to postcode\_suburb table of the Apache Derby embedded database.

Table: postcode\_suburb

Columns:

* + VARCHAR postcode
  + VARCHAR suburb

**Security**

* SecurityConfiguration.java

**How to deploy and run the AusPostAPI?**

* Executable jar (postcode-suburb-api.jar)

Any machine or AWS instance having java 1.8+ can be used to start the standalone application using:

java –jar postcode-suburb-api.jar

* Deployable war (postcode-suburb-api.war)

A war file is also provided if an application server is available in the AWS instance

**References**

* standard\_postcode\_file\_pc001\_27022021.pdf
* AusPost PE - Java.pptx