* **Part 1. API endpoint**

*Get loalhost/v1/costs/:name/in-state?:roomboard*

There are many ways to implement the API endpoints, In/out of state and room&board information can be implemented as separate endpoints, they also can be implemented as optional parameters.

In my demo, I have /in-state and /out-of-state as separate endpoints and have roomboard as query parameter.

* **Part 2. Some thoughts on this application design**

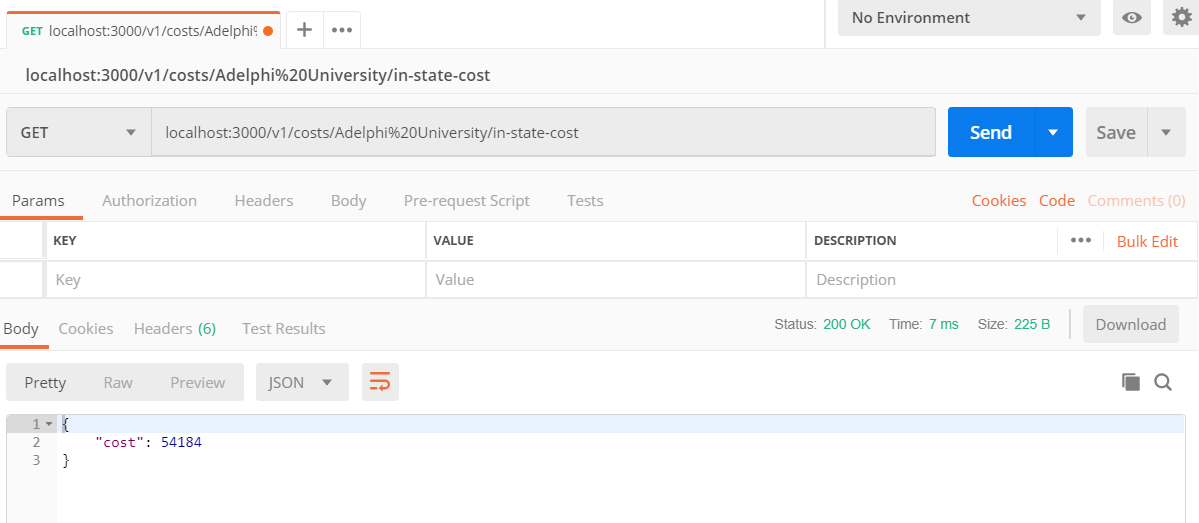
| **Data access layer** | **Option1(selected)**  Import data into database | **Option2**  Hold data in memory |
| --- | --- | --- |
| pros | Better support for other data operations for future extension. | Only requirement is to get cost, data structures like hashtable can provide fast look up. |
| cons | Some development overhead for this requirement. | Takes more time for initialization, and more memory usage. |

| **NoSQL / SQL** | **Option1(selected)**  **NoSQL** | **Option2**  **SQL** |
| --- | --- | --- |
| pros | Good for read heavy application.  Dynamic schema. document-oriented. | ACID operation |
| cons | Eventual consistency can cause problem in some situations. | Detailed database model needed before creation. Hard to scale. |

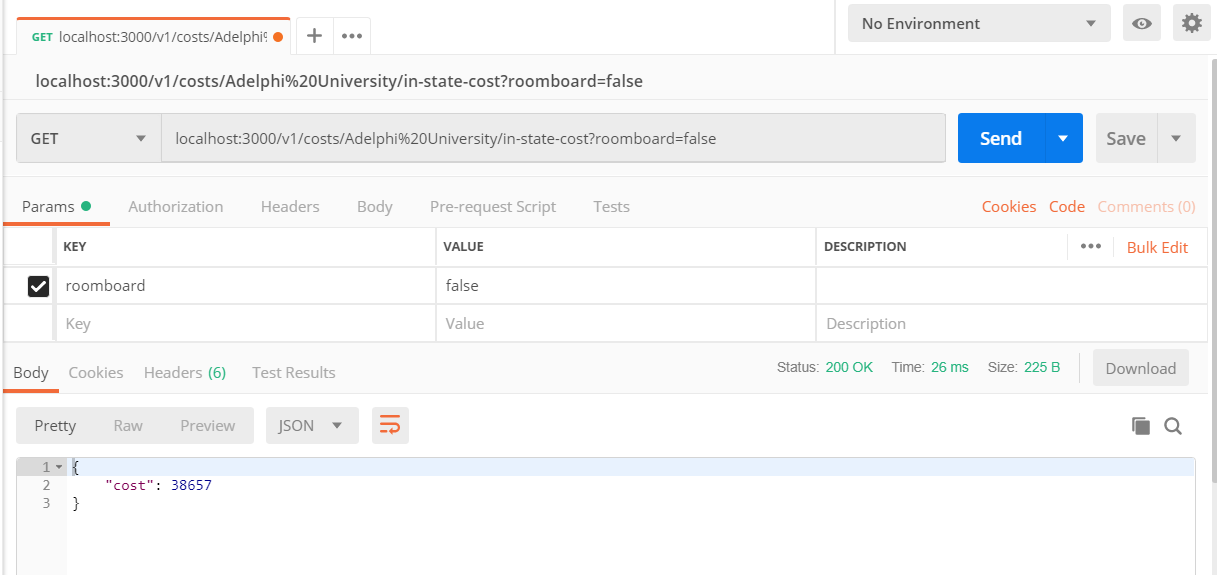
* **Part 3. Test Results**

There’s no client-side requirement specified, I used Postman as the tool to demonstrate consuming API.

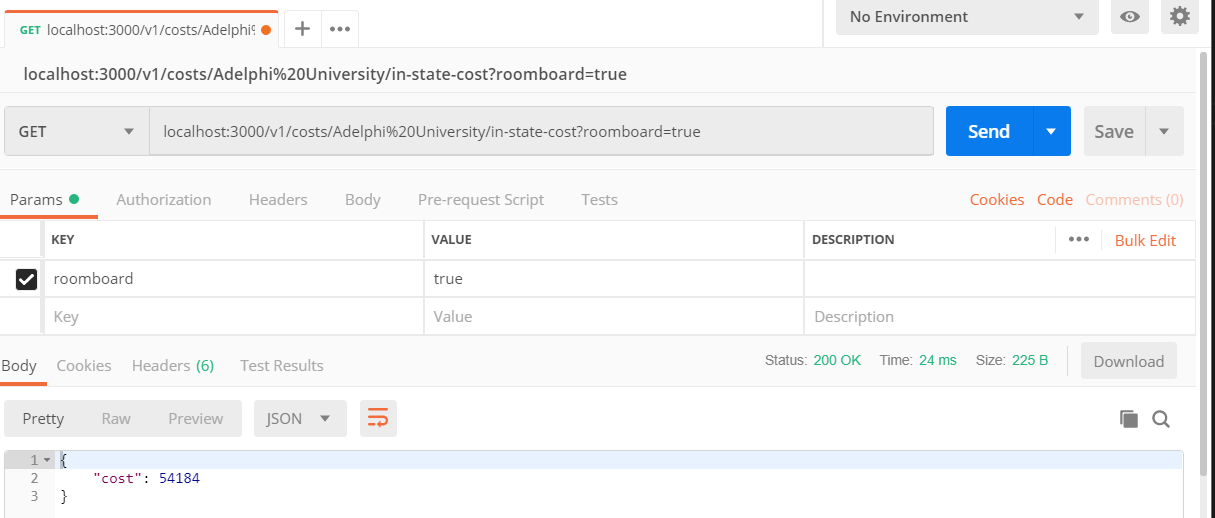
**Test case1:** correct college name, no room&board specified



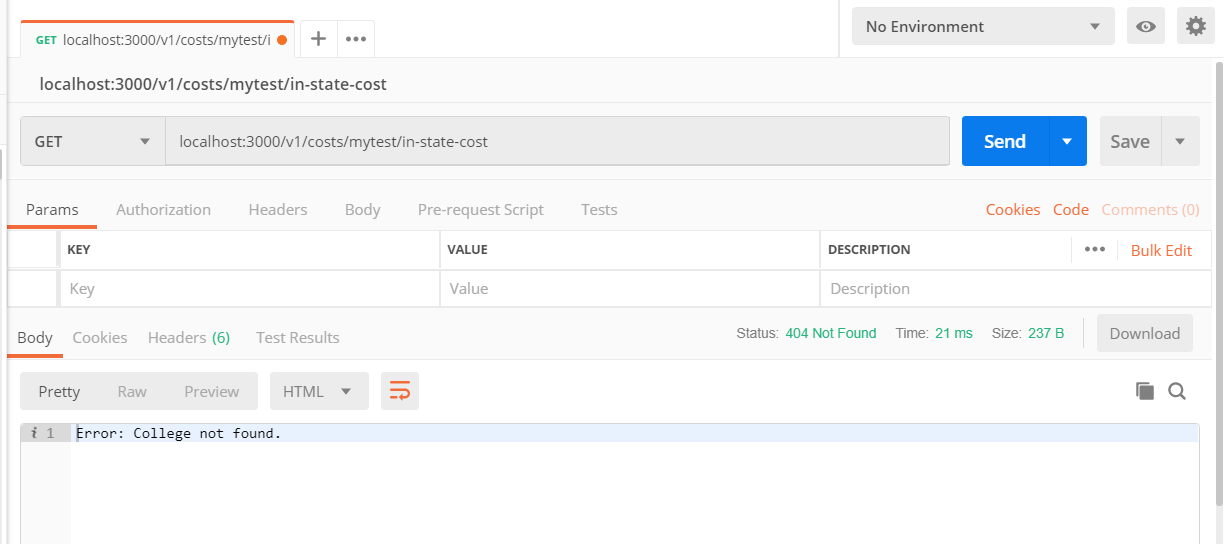
**Test** **case2**: correct college name, room&board specified as false



**Test** **cass3**: correct college name, room&board specified as true



**Test case4:** college name not found in the database



**Test case5:** college name missing in the request

