

# DOS

Ola Awwad  
Takreem Abdullah

November 2020

## 1 Output

In my system i used lumen framework using php langouge on ubuntu operating system using three virtual machines. We implement three services as wanted:

- Front-end service
- Order service
- Cataloge service

There are 3 operations and i will explain each one and how to works:

search

after sending the query from front-end controller service to the the cataloge service then the output will shown in the browser page with the book information : name and id of all books.

```
47 // search
48 public function getAllBooks($item){
49
50     // make a query
51     $startQuery2 = microtime(true);
52     if (Cache::has($item))
53     {
54         $valueBook = Cache::get($item);
55         $endcache2 = microtime(true);
56         echo " time of cache: ", ($endcache2 - $startQuery2);
57         return response()->json(json_decode($valueBook));
58     }
59
60     $url = "";
61     $randomNum = rand(1,4);
62     if ($randomNum < 3){
63         $url = "http://192.168.12.141:8080/query/booktopic/" . $item;
64     }
65     else {
66         $url = "http://192.168.12.141:8080/query/booktopic/" . $item;
67     }
68
69     ///////////////////////////////////////////////////
70     // get the topics from catalogue
71     $page = file_get_contents($url);
72     Cache::put($topic , $page ,360);
73     $end = microtime(true);
74     echo "outget from cache:" , ($end - $startQuery2);
75     return response()->json(json_decode($page));
76
77 }
```

Figure 1: search/ front end controller

```

17 // Show the specific book
18 public function showBooks($topic){
19 // get the books information from text file with name books, its in catalogue folder
20 $file = new \Illuminate\Filesystem\Filesystem();
21 $content = $file->get($_DIR . '/../books.txt');
22 $books = explode("\n", $content);
23 if (sizeof($books) < 2){
24     return response()->json([' ' => 'unfortunatly there is no books.... try later']);
25 }
26
27 $book_desc;
28 $json_inf;
29 $flag = false;
30 $count = -1;
31 // to display the books information
32 for ($i=0; $i<sizeof($books)-1; $i++){
33     $book_desc[$i] = explode(",",$books[$i]);
34     if ($book_desc[$i][3] == $topic){
35         $count++;
36         $flag = true;
37         $json_inf[$count]['Title'] = $book_desc[$i][0];
38         $json_inf[$count]['ID'] = $book_desc[$i][4];
39     }
40 }
41 if (!$flag){
42     return response()->json(['Message' => 'Type another book .']);
43 }
44 return response()->json($json_inf);
45 }

```

Figure 2: search/ catalogue controller

lookup

after sending the query from front-end controller service with the id of book to the the catalogue service then the output will shown in the browser page with the book information : name and id of the specific books.

```

13 // @@@@ function _construct(){}
14
15 // @@@@ //1..lookup
16 public function getBook($id){
17     $url = '';
18     $startQuery1 = microtime(true);
19     if (Cache::has($id)){
20         $bookValue = Cache::get($id);
21         $endCache1 = microtime(true);
22         echo "cache: " . ($endCache1 - $startQuery1);
23         return response()->json(json_decode($bookValue));
24     }
25
26     $num = rand(1,4);
27     if ($num < 3){
28         {
29             $url = 'http://192.168.32.141:8080/query/bookid/' . $id;
30         }
31     }
32     {
33         $url = 'http://192.168.32.141:8081/query/bookid/' . $id;
34     }
35
36     // get book from catalogue
37
38     $page = file_get_contents($url);
39     Cache::put($id, $page, 300);
40     $end = microtime(true);
41     echo "done put the book in the cache:" . ($end - $startQuery1);
42     return response()->json(json_decode($page));
43 }

```

Figure 3: lookup/ front end controller

```

catalogService / app / Http / Controllers / catalogController.php
79 //
80
81 public function showBook($id){
82     $file = new Illuminate\Filesystem\Filesystem();
83     $content = $file->get('_018_','.././././books.txt');
84     $books = explode("\n",$content);
85     if (sizeof($books) < 2){
86         return response()->json(['Message' => 'There is no books in store.']);
87     }
88     $book_details;
89     $json_data;
90     $flag = false;
91     for ($i=0 ; $i<sizeof($books)-1 ; $i++){
92         $book_details[$i] = explode(',',$books[$i]);
93         if ($book_details[$i][4] == $id){
94             $flag = true;
95             $json_data['title'] = $book_details[$i][0];
96             $json_data['Count'] = $book_details[$i][1];
97             $json_data['Price'] = $book_details[$i][2];
98             $json_data['Topic'] = $book_details[$i][3];
99             $json_data['ID'] = $book_details[$i][4];
100         }
101     }
102     if (!$flag){
103         return response()->json(['Message' => 'Wrong ID , check it please.']);
104     }
105     return response()->json($json_data);
106 }
107
108
109
110 }

```

Figure 4: lookup/ cataloge controller

buy

after sending the query from front-end controller service with the id of book to the the order service then check if the book is available in cataloge service via sending 2 queries. the first is check if there is copies from this book, if not the second query will paused. otherwise the second query will send to stressing the success of process.

```

orderService / app / Http / Controllers / frontEndController.php
84 public function buyBook($id){
85     $startQuery = microtime(true);
86     if (cache::has('Buy' . $id)){
87         // check if the item in cache
88         $bookIdvalue = cache::get('Buy' . $id);
89         $val = json_decode($bookIdvalue)->Message;
90         if ($val != "Your Buy operation done successfully.")
91         {
92             $endcache3 = microtime(true);
93             echo "cache:" . ($endcache2 - $startQuery);
94             return response()->json(json_decode($bookIdvalue));
95         }
96         cache::forget('Buy' . $id);
97     }
98     $url = '';
99     $num = rand(1,4);
100     if ($num < 3)
101     {
102         $url = 'http://192.168.12.142:8000/buy/' . $id;
103     }
104     else
105     {
106         $url = 'http://192.168.12.142:8001/buy/' . $id;
107     }
108     $page = file_get_contents($url);
109     Cache::put('Buy' . $id , $page ,60); // put it in cache
110     $end = microtime(true); // end time of query without cache
111     echo "put the book in the cache : " . ($end - $startQuery);
112     return response()->json(json_decode($page));
113 }

```

Figure 5: buy/ front end controller

```

47 public function find($id){
48     $file = new \Illuminate\Filesystem\Filesystem();
49     $content = $file->get(_DIR_.'/../books.txt');
50     $books = explode("\n",$content);
51     if (sizeof($books) < 2){
52         return response()->json(['Message' => 'No']);
53     }
54     $book_details;
55     $flag = false;
56     for ($i=0 ; $i<sizeof($books)-1 ; $i++){
57         $book_details[$i] = explode(",",$books[$i]);
58         if ($book_details[$i][4] == $id){
59             if ($book_details[$i][1] < 1){
60                 return response()->json(['Message' => 'Zero']);
61             }
62             $book_details[$i][1]--;
63             $book_details[$i][1] = $book_details[$i][1].".";
64             $flag=true;
65         }
66     }
67     if(!($flag)){
68         return response()->json(['Message' => 'wrong']);
69     }
70     for ($i=0 ; $i<sizeof($books)-1 ; $i++){
71         $books[$i] = implode(",",$book_details[$i]);
72     }
73     $content = implode("\n",$books);
74     $file->put(_DIR_.'/../books.txt' , $content,false);
75     return response()->json(['Message' => 'Done']);
76 }
77

```

Figure 6: buy/ cataloge controller

```

10 * Create a new controller instance.
11 * @return void
12 */
13 public function __construct(){}
14
15 public function buy($id){
16     $url = 'http://192.168.17.141:8080/query/check/'. $id;
17     $page = file_get_contents($url);
18     $resp = response()->json(json_decode($page));
19     $flag = json_decode($page)->Message;
20     if ($flag == 'Zero'){
21         return response()->json(['Message' => 'There is no copies of this book.']);
22     }
23     if ($flag == 'No'){
24         return response()->json(['Message' => 'There is no books in the store.']);
25     }
26     if ($flag == 'wrong'){
27         return response()->json(['Message' => 'Wrong ID , please check it.']);
28     }
29     if ($flag == 'Done'){
30         $url2 = 'http://192.168.17.141:8080/query/check/'. $id;
31         $page1 = file_get_contents($url2);
32         $url2 = 'http://192.168.17.141:8080/update/buy/'. $id;
33         $page2 = file_get_contents($url2);
34         return response()->json(json_decode($page2));
35     }
36 }
37

```

Figure 7: buy/ order controller

but unfortunately i didn't take the responses because our virtual boxes were crashed and we reinstall them but many issues happened and this make us very upset for this. Also we tried in 2 another computers but the same problems happened, we don't know why unfortunately

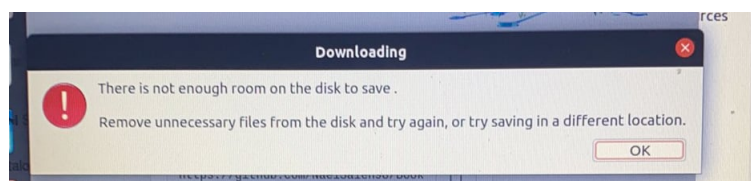


Figure 8: problem1

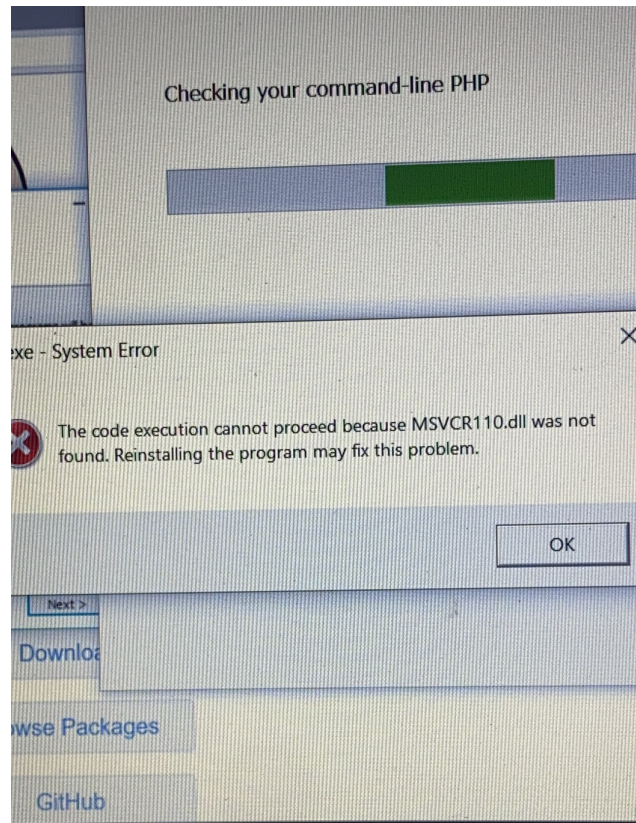


Figure 9: problem 2

## 2 How to make it works

first: make the machines works with each other: create a network:  
go to preferences then network, add anew network with networkcidr:192.168.1/24  
with name: lab1.  
connect three virtuals to this network:  
go to network in each virtual  
change the port for each:  
go to settings in network then to ip v4 then change the port for each, for exam-  
ple : 192.168.12.141:8001  
192.168.19.141:8000

ubuntu3 [Running] - Oracle VM VirtualBox  
File Machine View Input Devices Help

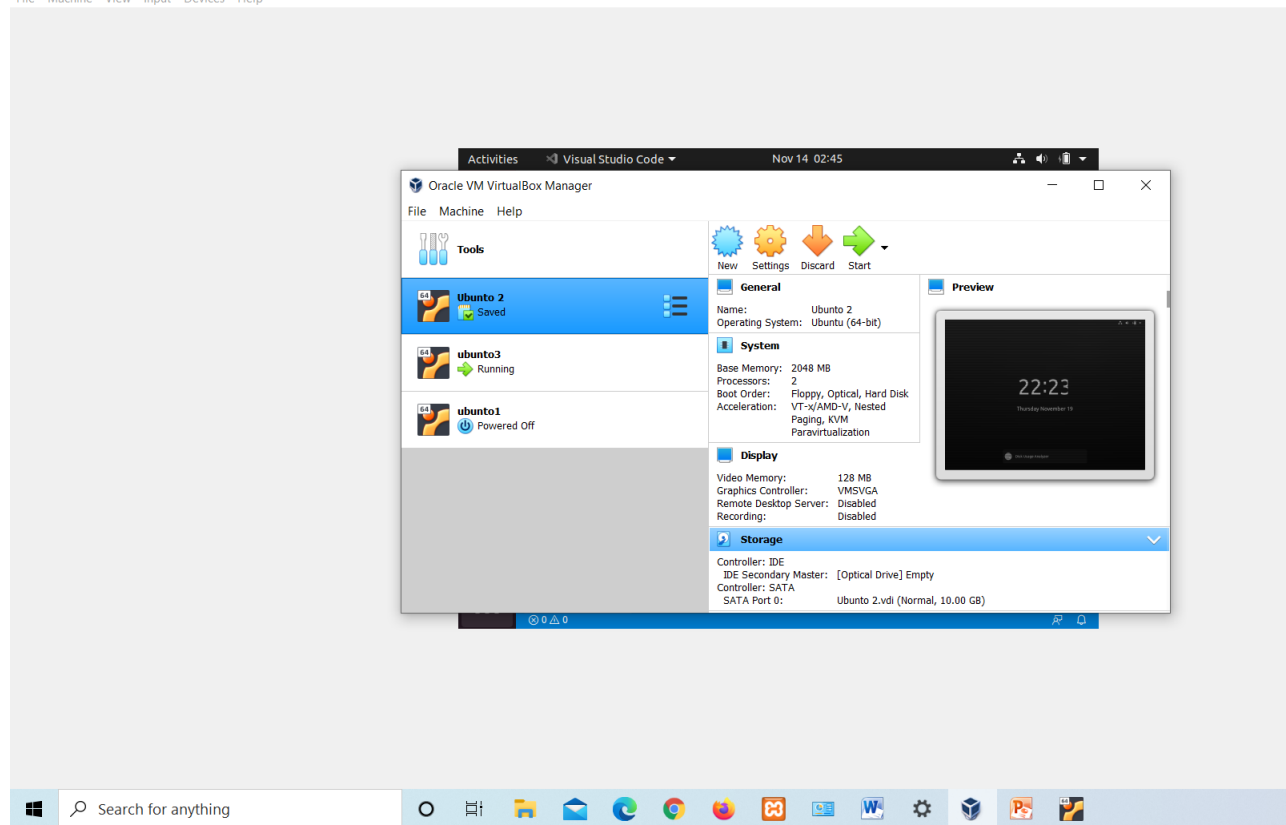


Figure 10: virtual 1