

In Provider Project (First Project)

1. **php artisan serve** to run the project in 8000 port

2. Create a controller

```
php artisan make:controller APIController --resource
```

3. In the APIController, complete your function

Example for getting all jobs:

a. SQL command to get all job information

b. Return response in JSON format

```
public function getAllJobs()
{
    $jobAds = Advertisement::join('jobs', 'jobs.id', '=',
    'advertisements.job_id')
        ->join('companies', 'companies.id', '=',
    'advertisements.company_id')
        ->join('job_specializations', 'job_specializations.id', '=',
    'jobs.specialization_id')
        ->where(['advertisements.status', '=', 'Active'])
        ->get();

    return $this->returnJSONresponse($jobAds);
}
```

Method to return JSON response (if you want to directly do it in your function also np)

```
public function returnJSONresponse($params)
{
    if($params == "[]"){
        return response()->json([
            'success' => true,
            'message' => 'Record(s) Retrieval Fail',
            'data' => "Jobs not found"
        ]);
    }

    return response()->json([
        'success' => true,
        'message' => 'Record(s) Retrieval Success',
        'data' => $params
    ]);
}
```

3. Register your api in [api.php](#)

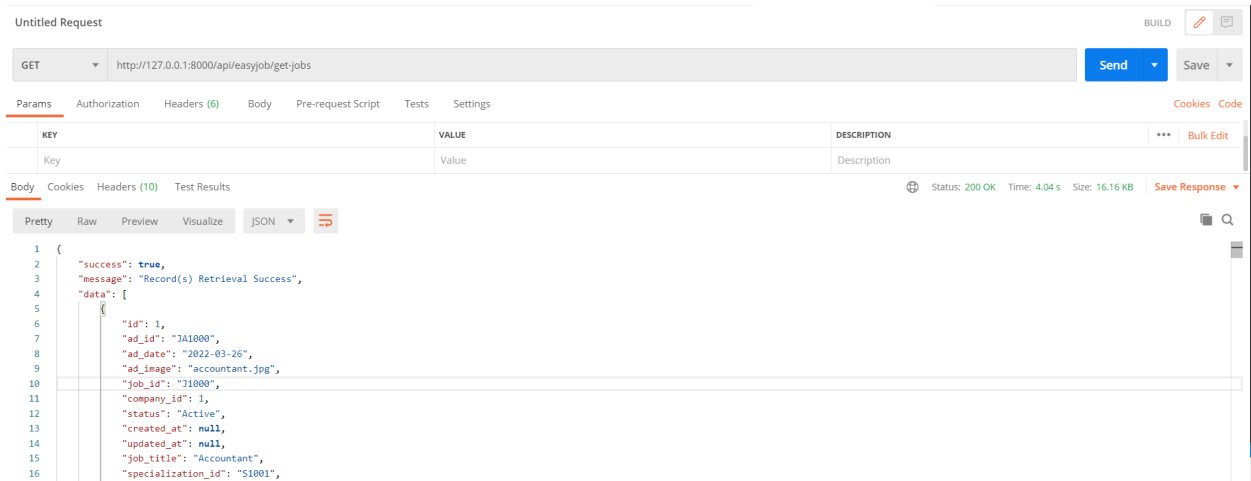
```
Route::get("easyjob/get-jobs",
```

```
"App\Http\Controllers\API\JobsAPIController@getAllJobs");
```

Notes: When we want to consume API, you will need to put **/api** in front your path in order to make it as a complete url, for example,

<http://127.0.0.1:8000/api/easyjob/get-jobs>

4. You can test your API in [PostMan](#) (can skip this part if you don't have)
 - a. Choose GET (since you are getting the jobs now)
 - b. Paste the url (<http://127.0.0.1:8000/api/easyjob/get-jobs>)
 - c. Click Send
 - d. Under Body section, view the response (Can choose the Pretty model to have a clearer view)



OR

In first project, you can test you API by using **dd** helper function in controller by entering the link such as <http://127.0.0.1:8000/api/easyjob/get-jobs>

5. **Do not stop your project 1** [You will need it to test your api in 2nd project]

In Consumer Project (Second Project)

1. Create a new project as 2nd project

```
composer create-project laravel/laravel secondApp
cd secondApp
```
2. **php artisan serve --port=8001** to allow second project to run in different port
3. Create a controller to consume your API

```
php artisan make:controller APIController --resource
```
4. In the controller, import this `use Illuminate\Support\Facades\Http;`

```
5 use Illuminate\Http\Request;
6 use Illuminate\Support\Facades\Http;
```
5. In my case, I want to display all the jobs through `index()` controller method

Direct API call in method

```
public function index()
{
    $response = Http::get("http://127.0.0.1:8000/api/easyjob/get-jobs");
    $response = json_decode($response);
    $jobs = $response->data;

    dd($jobs)    // comment this code if the returned data is okay

    return view('/JobSeeker/Jobs/joblisting', compact('jobs'));
}
```

Notes: You can use **dd** helper function to view the return data before return back to the blade

More HTTP Client info can be found on <https://laravel.com/docs/9.x/http-client>

6. Define route in web.php

```
Route::get('view-easyjob-jobs',
'App\Http\Controllers\JobAPIController@index');
```

7. Design your blade view

```
@foreach ($jobs as $job)
    <h5 class="card-title mb-3">{{ $job->job_title }}</h5>
    <div class="card-job-info ml-4">{{ $job->specialization_name }}</div>
@endforeach
```

8. Run it.

Extra Notes:

If the next API method you want to do view job details based on the job id (parameter included),

In Provider Project (First Project)

In API controller, add on this function

```
public function getJobDetails(Request $request){
    $id = $request->query('id');

    $jobAd = Advertisement::join('jobs','jobs.id','=','advertisements.job_id')
        ->join('companies', 'companies.id', '=', 'advertisements.company_id')
        ->join('job_specializations', 'job_specializations.id', '=',
'jobs.specialization_id')
        ->where([[ 'advertisements.ad_id', '=', $id],
[ 'advertisements.status', '=', 'Active' ]])
        ->first();

    return $this->returnJSONresponse($jobAd);
}
```

```
}
```

In api.php, add on this route

```
Route::get("easyjob/get-job-details",  
"App\Http\Controllers\API\JobsApiController@getJobDetails");
```

In Consumer Project (Second Project)

1. In my case, I want to display job details using show(\$id) function

```
public function show(Request $request)  
{  
    $id = $request->query('id');  
    $response =  
Http::get("http://127.0.0.1:8000/api/easyjob/get-job-details", [  
        'id' => $id,  
    ]);  
    $response = json_decode($response);  
    $job_detail = $response->data;  
  
    return view('/JobSeeker/Jobs/jobDetails', compact('job_detail'));  
}
```

2. You can directly use **\$job_detail** that you returned to get the data in blade view by using for each loop