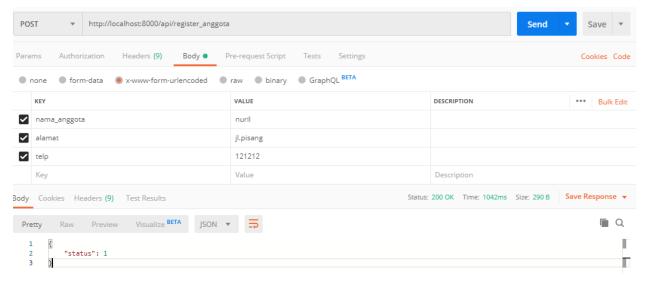
CRUD Anggota & Buku

- 1. Anggota
 - Create

```
public function create(Request $req){
    $validator = Validator::make($req->all(),
            'nama_anggota' => 'required',
            'alamat' => 'required',
            'telp' => 'required'
    );
    if($validator->fails()){
        return Response()->json($validator->errors());
    $simpan = ModelAnggota::create([
        'nama_anggota'=>$req->nama_anggota,
        'alamat'=>$req->alamat,
        'telp'=>$req->telp,
    ]);
    if($simpan){
        return Response()->json(['status'=>1]);
    } else {
        return Response()->json(['status'=>0]);
```



Read

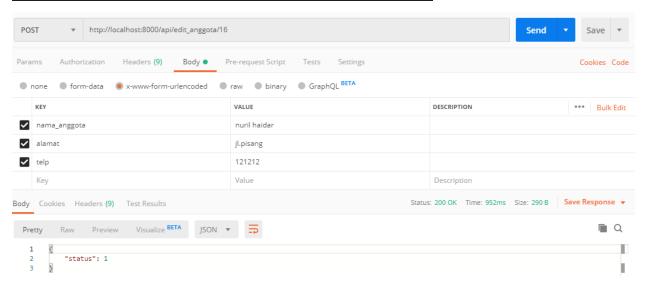
```
public function tampil(){
    $tampil = ModelAnggota::get();

    if($tampil){
        return Response()->json([$tampil]);
    } else {
        return Response()->json([$tampil->errors()]);
    }
}
```

```
▼ http://localhost:8000/api/tampil_anggota
                                                                                                                                                                           Send
                                                                                                                                                                                                Save ▼
                                                                                                                                   Status, 200 OK TITTE, 099THS SIZE, T.OT KD
Body Cookies Headers (9) Test Kesults
               ■ Q
  Pretty
     1
                           "id_anggota": 11,
"nama_anggota": "Ayam",
"alamat": "Malang",
                           "telp": 123456789,
"created_at": null,
"updated_at": null
    11
                           "id_anggota": 12,
"nama_anggota": "Burung",
    12
    13
                           "alamat": "Surabaya",
"telp": 123456789,
   15
                           "created_at": null,
"updated_at": null
   16
    17
   19
20
                           "id_anggota": 13,
"nama_anggota": "Merak",
                           "alamat": "Bandung",
"telp": 123456789,
   22
   23
                           "created_at": null,
"updated_at": null
    26
```

Update

```
public function edit($id,Request $req){
   $validator = Validator::make($req->all(),
            'nama anggota'=>'required',
            'alamat'=>'required',
            'telp'=>'required'
    );
    if($validator->fails()){
        return Response()->json($validator->errors());
   $edit = ModelAnggota::where('id_anggota',$id)->update([
        'nama_anggota'=>$req->nama_anggota,
        'alamat'=>$req->alamat,
        'telp'=>$req->telp
    ]);
    if($edit){
        return Response()->json(['status'=>1]);
    } else {
        return Response()->json(['status'=>0]);
```

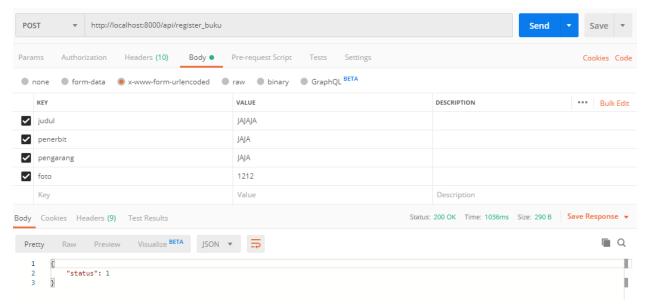


• Delete

2. Buku

Create

```
public function create(Request $req){
   $validator = Validator::make($req->all(),
            'judul' => 'required',
            'penerbit' => 'required',
            'pengarang' => 'required',
'foto' => 'required'
    if($validator->fails()){
        return Response()->json($validator->errors());
    }
   $simpan = ModelBuku::create([
        'judul'=>$req->judul,
        'penerbit'=>$req->penerbit,
        'pengarang'=>$req->pengarang,
        'foto'=>$req->foto,
    ]);
    if($simpan){
        return Response()->json(['status'=>1]);
        return Response()->json(['status'=>0]);
    }
```



Read

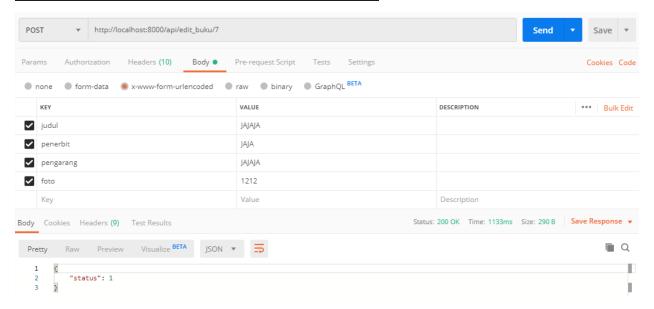
```
public function tampil(){
    $tampil = ModelBuku::get();

    if($tampil){
        return Response()->json([$tampil]);
    } else {
        return Response()->json([$tampil->errors()]);
    }
}
```

```
GET
                    ▼ http://localhost:8000/api/tampil_buku
                                                                                                                                                                                                            Send
                                                                                                                                                                                                                                     Save ▼
                         Preview Visualize BETA JSON ▼
                                                                                                                                                                                                                                          ■ Q
Pretty
                              "id_buku": 1,
                              "judul": "Laskar Pelangi",
"penerbit": "Erlangga",
"pengarang": "Andrea Hirata",
                              "foto": "10290192",
                             "created_at": null,
"updated_at": null
  10
 11
 12
                             "id_buku": 2,
"judul": "Sang Pemimpi",
"penerbit": "Erlangga",
"pengarang": "Andrea Hirata",
"foto": "10290192",
"created_at": null,
 13
 14
 16
17
 18
                              "updated_at": null
 20
 21
                             "id_buku": 3,
"judul": "Dilan 1990",
"penerbit": "Angkasa",
"pengarang": "Pidi Baiq",
 23
 24
 25
                              "foto": "10290192",
                             "created_at": null,
```

Update

```
public function edit($id,Request $req){
   $validator = Validator::make($req->all(),
            'judul'=>'required',
            'penerbit'=>'required',
            'pengarang'=>'required',
            'foto'=>'required',
   if($validator->fails()){
       return Response()->json($validator->errors());
   $edit = ModelBuku::where('id_buku',$id)->update([
        'judul'=>$req->judul,
        'penerbit'=>$req->penerbit,
        'pengarang'=>$req->pengarang,
        'foto'=>$req->foto
   ]);
   if($edit){
       return Response()->json(['status'=>1]);
   } else {
       return Response()->json(['status'=>0]);
```



• Delete

```
public function hapus($id){
    $hapus = ModelBuku::where('id_buku',$id)->delete();

$a = "Data Berhasil Dihapus";
$b = "Data Gagal Dihapus";
if($hapus){
    return Response()->json([$a]);
} else {
    return Response()->json([$b]);
}
```

