BACKEND

1. Web => controller => api => commoncontroller – getservicename(added additionaly in line 660)

[ValidateAuthToken]

#region public HttpResponseMessage GetServiceNameList()

[HttpGet]

[Route("Api/Common/GetServiceName")]

public HttpResponseMessage GetServiceName()

{

HttpResponseMessage response;

var serviceCollectionResult = new CollectionResult<Service>();

try

{

var serviceTypeResult =

\_serviceBusiness.Where(x => x.ParentId != -1 && x.ParentId < 56 && x.IsActive);

if (!serviceTypeResult.Status)

throw new Exception(serviceTypeResult.Message);

foreach (var ser in serviceTypeResult.Data)

{

ser.ServiceName = ser.Name;

ser.ServiceTypeId = ser.Id;

ser.ServiceId = ser.Id;

serviceCollectionResult.Data.Add(ser);

}

response = Request.CreateResponse(HttpStatusCode.OK, new Result<object>

{

Status = true,

Message = string.Format("Service List"),

Data = from sc in serviceCollectionResult.Data.OrderBy(x => x.ServiceName)

select new

{

sc.ServiceId,

sc.ServiceTypeId,

sc.ServiceName

}

});

}

catch (Exception ex)

{

response = Request.CreateResponse(HttpStatusCode.OK, new Result<int>

{

Status = false,

Message = ex.Message,

Data = default(int)

});

}

return response;

}

#endregion

FRONTEND

Hirer => employer => Createjob

Hirer => jobmenu => grouplist => Creategroup

(hint: employer => about => services --- uses filter)