|  |  |
| --- | --- |
| **CONTROLLER** | **AppDbContext** |
| [HttpGet]  public ActionResult Get()  {  try  {  return Ok(context.LOB\_All());  //return Ok("aaaa");  }  catch (Exception ex)  {  return BadRequest(ex.Message);  }  } | public List<LOBs> LOB\_All()  {  return new List<LOBs> {  new LOBs {ID = 1, LOB="uno", Reference="aaa" },  new LOBs {ID = 2, LOB="dos", Reference="ver" }  };  } |
| private readonly AppDbContext context;  //Constructor, mismo nombre que la clase  public LOBController(AppDbContext context)  {  this.context = context;  }  [HttpGet]  public ActionResult Get()  {  try  {  return Ok(context.LOB\_All.ToList());  }  catch (Exception ex)  {  return BadRequest(ex.Message);  }  } | public AppDbContext(DbContextOptions<AppDbContext> options) : base(options)  {  }  public DbSet<LOBs> LOB\_All { get; set; } |
| var listado = context.LOB\_All2.ToList();  if (listado.Count == 0)  return StatusCode(404, "Empty");  return Ok(listado); |  |
|  |  |

// Get: api/<controller>

[HttpGet]

public ActionResult Get()

{

try

{

List<ML\_LOBs> lista = new List<ML\_LOBs>();

var conn = "data source=(localdb)\\MeLi;Initial Catalog=MeLi;Integrated Security=True;";

using (SqlConnection cn = new SqlConnection(conn))

{

SqlCommand cmd = cn.CreateCommand();

cmd.CommandText = "Select \* from ML\_LOBs";

cn.Open();

SqlDataReader rdr = cmd.ExecuteReader();

ML\_LOBs lob;

while (rdr.Read())

{

lob = new ML\_LOBs

{

ID = (int)rdr["ID"],

LOB = (string)rdr["LOB"]

};

lista.Add(lob);

}

}

return Ok(lista.ToList());

}

catch (Exception ex)

{

return BadRequest(ex.Message);

}

}