[Skip to content](https://chatgpt.com/#main)

**You said:**

in .NET Core WebAPI how do i return a list of all data from the ResourceBooks plus the related (Foreign Key) data from the Resource table using EF Core and LINQ method syntax?

**ChatGPT said:**

To return a list of all data from the ResourceBooks table **including** the related Resource data using **Entity Framework Core** and **LINQ method syntax**, you can use the .Include() method.

Assuming you have a one-to-many or many-to-one relationship between ResourceBooks and Resource (i.e., ResourceBook has a foreign key pointing to Resource), here's how you can do it:

**Example**

**Models:**

public class ResourceBook

{

public int Id { get; set; }

public string Title { get; set; }

public int ResourceId { get; set; } // Foreign Key

public Resource Resource { get; set; } // Navigation Property

}

public class Resource

{

public int Id { get; set; }

public string Name { get; set; }

public ICollection<ResourceBook> ResourceBooks { get; set; }

}

**Query in Controller or Service:**

using Microsoft.EntityFrameworkCore;

public class ResourceBooksController : ControllerBase

{

private readonly YourDbContext \_context;

public ResourceBooksController(YourDbContext context)

{

\_context = context;

}

[HttpGet]

public async Task<ActionResult<IEnumerable<ResourceBook>>> GetResourceBooks()

{

var resourceBooks = await \_context.ResourceBooks

.Include(rb => rb.Resource)

.ToListAsync();

return Ok(resourceBooks);

}

}

**Explanation:**

* .Include(rb => rb.Resource) tells EF Core to **eager load** the related Resource data for each ResourceBook.
* .ToListAsync() executes the query and materializes the data as a list.
* Ok() wraps the result in a 200 OK response.

**Optional: Project into DTOs (Recommended for APIs)**

Instead of returning EF entities directly, project into DTOs:

var resourceBooks = await \_context.ResourceBooks

.Include(rb => rb.Resource)

.Select(rb => new ResourceBookDto

{

Id = rb.Id,

Title = rb.Title,

Resource = new ResourceDto

{

Id = rb.Resource.Id,

Name = rb.Resource.Name

}

})

.ToListAsync();