

# MINIMAL APIS

## CRUD



```
dotnet new web -o MyMinimalApp
```

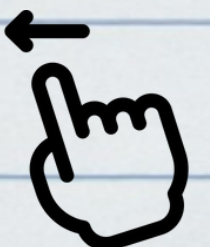


```
var builder = WebApplication.CreateBuilder(args);  
var app = builder.Build();  
  
app.MapGet("/", () => "Hello World!");  
  
app.Run();
```

@romain-od



www.code-review.tech



# CREATE



```
app.MapPost("/items", async (Item item, MyContextDb db) =>
{
    db.Items.Add(item);
    await db.SaveChangesAsync();
    return Results.Created($" /items/{item.Id}", item);
});
```

# READ



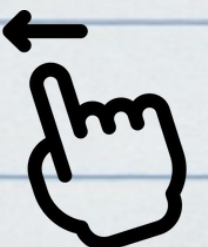
```
app.MapGet("/items", async (MyContextDb db) =>
    await db.Items.ToListAsync());

app.MapGet("/items/{id}", async (int id, MyContextDb db) =>
    await db.Items.FindAsync(id)
    is Item item
        ? Results.Ok(item)
        : Results.NotFound());
```

@romain-od



[www.code-review.tech](http://www.code-review.tech)





# UPDATE



```
app.MapPut("/items/{id}", async (int id, Item inputItem, MyContextDb db) =>
{
    var item = await db.Items.FindAsync(id);
    if (item is null) return Results.NotFound();
    item.Name = inputItem.Name;
    await db.SaveChangesAsync();
    return Results.NoContent();
});
```

# DELETE



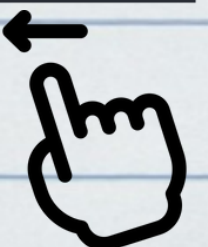
```
app.MapDelete("/items/{id}", async (int id, MyContextDb db) =>
{
    if (await db.Items.FindAsync(id) is Item item)
    {
        db.Items.Remove(item);
        await db.SaveChangesAsync();
        return Results.Ok(item);
    }

    return Results.NotFound();
});
```

@romain-od



www.code-review.tech





**THANKS FOR READY**  
**SHARE YOUR THOUGHTS**

**FOLLOW ME FOR MORE SHARING**



@romain-od



[www.code-review.tech](http://www.code-review.tech)