

7.4-3 Endpunkte mit CRUD-Operationen

Installation Postman

Installieren Sie als Client für den Aufruf des WebApis die API-Plattform Postman:

[snap install postman](#)

Service-Endpunkte

Um die verschiedenen Eigenschaften eines Films abzubilden, erstellen Sie unter min-api-with-mongo/WebApi ein neues File **Movie.cs** mit folgendem Codegerüst:

```
public class Movie
{
    public string Id { get; set; } = "";
    public string Title { get; set; } = "";
    public int Year { get; set; }
    public string Summary { get; set; } = "";
    public string[] Actors { get; set; } = Array.Empty<string>();
}
```

Erweitern Sie min-api-with-mongo/WebApi/Program.cs:

```
using MongoDB.Driver;

var builder = WebApplication.CreateBuilder(args);

var movieDatabaseCollection = builder.Configuration.GetSection("DatabaseSettings");
builder.Services.Configure<DatabaseSettings>(movieDatabaseCollection);

var app = builder.Build();

app.MapGet("/", () => "Minimal API Version 1.0");

app.MapGet("/check", (Microsoft.Extensions.Options.IOptions<DatabaseSettings> options) =>
{
    try
    {
        var mongoClient = new MongoClient(options.Value.ConnectionString);
        var dbs = mongoClient.ListDatabaseNames().ToList();

        return $"Zugriff auf MongoDB ok. Vorhandene DBs: " + String.Join(", ", dbs);
    }
    catch
```

```

    {
        return "Leider hat Zugriff auf MongoDB nicht geklappt.";
    }
});

// Get all Movies
// Gibt alle vorhandenen Movie-Objekte mit Statuscode 200 OK zurück.
app.MapGet("/api/movies", () =>
{
    throw new NotImplementedException();
});

// Get Movie by id
// Gibt das gewünschte Movie-Objekt mit Statuscode 200 OK zurück.
// Bei ungültiger id wird Statuscode 404 not found zurückgegeben.
app.MapGet("/api/movies/{id}", (string id) =>
{
    if (id == "1")
    {
        var myMovie = new Movie()
        {
            Id = "1",
            Title = "Ein Quantum Trost",
        };

        return Results.Ok(myMovie);
    }
    else
    {
        return Results.NotFound();
    }
});

// Insert Movie
// Wenn das übergebene Objekt eingefügt werden konnte,
// wird es mit Statuscode 200 zurückgegeben.
// Bei Fehler wird Statuscode 409 Conflict zurückgegeben.
app.MapPost("/api/movies", (Movie movie) =>
{
    throw new NotImplementedException();
});

// Update Movie
// Gibt das aktualisierte Movie-Objekt zurück.
// Bei ungültiger id wird Statuscode 404 not found zurückgegeben.
app.MapPut("/api/movies/{id}", (string id, Movie movie) =>
{
    throw new NotImplementedException();
});

// Delete Movie

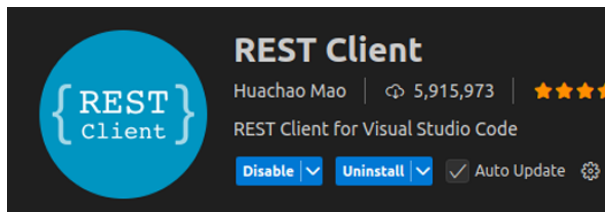
```

```
// Gibt bei erfolgreicher Löschung Statuscode 200 OK zurück.
// Bei ungültiger id wird Statuscode 404 not found zurückgegeben.
app.MapDelete("/api/movies/{id}", (string id) =>
{
    throw new NotImplementedException();
});

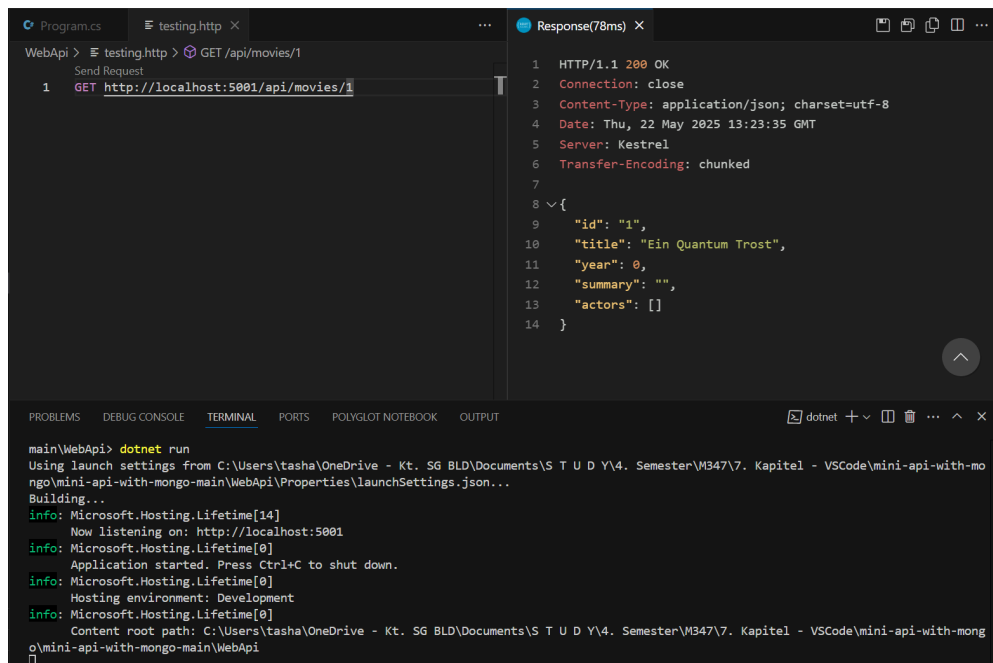
app.Run();
```

Testen mit REST-Client

Installieren Sie in Visual Code die Extension REST-Client:



Erstellen Sie im Projektverzeichnis eine Datei **testing.http** mit dem abgebildeten Inhalt und klicken Sie auf **Send Request**:



```
Program.cs  testing.http X  ...  Response(83ms) X
WebApi >  testing.http > GET /api/movies/6
1  ###
  Send Request
2  GET http://localhost:5001/api/movies/6
1  HTTP/1.1 404 Not Found
2  Content-Length: 0
3  Connection: close
4  Date: Thu, 22 May 2025 13:32:37 GMT
5  Server: Kestrel
6
7
```