

7.4-3: Minimal API mit MongoDB - Teil 3

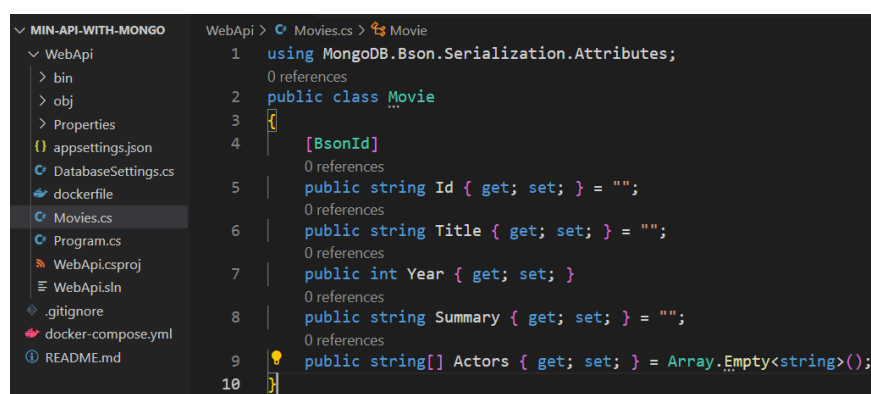
1. Ziele

- Sie unterscheiden GET, POST, PUT und DELETE-Requests
- Sie erstellen CRUD-Endpunkte für ein Web-API
- Sie testen ein Web-API mit einem REST-Client

2. Klassen für Filme erstellen

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

```
using MongoDB.Bson.Serialization.Attributes;
public class Movie
{
    [BsonId]
    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>();
}
```



Die Kennzeichnung der Property Id als [BsonId] markiert Id als Repräsentation des MongoDB _id Attributes. Sie können somit die Primärschlüssel später selber angeben und einfacher nach diesen suchen z.B. Id = "1"

3. Endpunkte für REST vorbereiten

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

```
// 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();
});
// 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) =>
{
    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();  
});
```

4. Beispiel-Code für GET und POST ohne Datenbankbindung erstellen

Implementieren Sie den Endpunkt Get Movie by id mit dem abgebildeten Beispielcode (ohne DB-Anbindung) so, dass die übergebenen Parameter entgegengenommen werden und ein korrektes Ergebnis zurück geliefert wird.

```
// 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();  
    }  
});
```

Wenn alles korrekt ist wird der Endpunkt für Get Movie by Id aufgerufen und der Film mit der Id 1 zurückgegeben. Testen Sie auch mit anderen Id's.

→ Man muss zuerst die Applikation mit `dotnet run` starten und dann auf auf `"Send Request"` drücken, damit der Request ausgeführt wird.

- Wenn id existiert

```
Send Request
GET http://localhost:5001/api/movies/1

1 HTTP/1.1 200 OK
2 Connection: close
3 Content-Type: application/json; charset=utf-8
4 Date: Thu, 22 May 2025 13:54:09 GMT
5 Server: Kestrel
6 Transfer-Encoding: chunked
7
8 {
9   "id": "1",
10  "title": "Ein Quantum Trost",
11  "year": 0,
12  "summary": "",
13  "actors": []
14 }
```

- Wenn id nicht existiert

```
Send Request
1 GET http://localhost:5001/api/movies/3

1 HTTP/1.1 404 Not Found
2 Content-Length: 0
3 Connection: close
4 Date: Thu, 22 May 2025 13:50:20 GMT
5 Server: Kestrel
```

- eigene id hinzugefügt zum testen

```
WebApi > C# Program.cs
50 app.MapGet("/api/movies/{id}", (string id) =>
51     if (id == "1")
52     {
53         var myMovie = new Movie()
54         {
55             Id = "2",
56             Title = "Meow",
57         };
58         return Results.Ok(myMovie);
59     }
60     else
61     {
62         return Results.NotFound();
63     }
64 });
```

```
WebApi > testing.http > GET /api/movies/2
Send Request
1 GET http://localhost:5001/api/movies/2

1 HTTP/1.1 200 OK
2 Connection: close
3 Content-Type: application/json; charset=utf-8
4 Date: Thu, 22 May 2025 13:55:34 GMT
5 Server: Kestrel
6 Transfer-Encoding: chunked
7
8 {
9   "id": "2",
10  "title": "Meow",
11  "year": 0,
12  "summary": "",
13  "actors": []
14 }
```