



### Aufgabe 3: Service-Endpunkte | Einzelarbeit | 35'

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* 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", () =>
{
    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();
});

```

Implementieren Sie die Endpunkte mit Beispielcode (ohne DB-Anbindung) so, dass die übergebenen Parameter entgegengenommen werden und ein korrektes Ergebnis (gemäss Beschreibung) zurück geliefert wird.

**Hinweis** Geben Sie ein C#-Objekt mit `Results.Ok(myObject)` zurück, wird es im Response im JSON-Format serialisiert und mit HTTP-Statuscode 200 OK zurückgegeben. Geben Sie `Results.NotFound()` zurück, wird ein HTTP-Statuscode 404 Not found zurückgegeben.

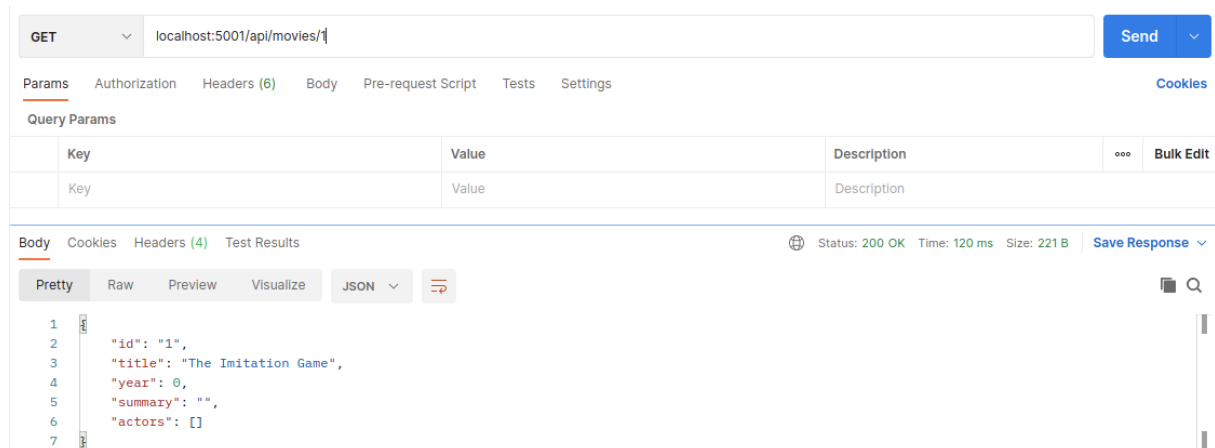
Beispiel für *Get Movie by id*:

```

app.MapGet("api/movies/{id}", (string id) =>
{
    if(id == "1")
    {
        var myMovie = new Movie()
        {
            Id = "1",
            Title = "Asterix und Obelix",
        };
        return Results.Ok(myMovie);
    }
    else
    {
        return Results.NotFound();
    }
});

```

Starten Sie die Anwendung und testen Sie die verschiedenen Endpunkte mit Postman. Vergewissern Sie sich, dass das API auf einen Request von Postman korrekt antwortet.



**Abbildung 1:** Postman GET-Request für Movie mit Id 1

**Hinweis** Achten Sie darauf, dass die HTTP-Methode korrekt gesetzt ist. Wenn Sie im Request JSON-Daten übergeben, wählen sie in Tab Body die Option raw mit Typ JSON.

**Tipp:** Wenn Sie einen Postman-Account haben, können Sie die verschiedenen Requests in Ihrem Postman-Workspace als Collection abspeichern.