

WebAPI and Fiddler Basics

Estimated time for completion: 30 minutes

Overview:

In this lab you will use WebAPI to expose simple movie data as JSON and XML over HTTP. You will then learn to use Fiddler to make HTTP requests to access the movie data.

Goals:

* Use a WebAPI controller to provide movie listings
* Use Fiddler to view and make HTTP requests

Lab Notes:

The lab uses a movie review website as its base. Registered users can view movies and reviews. Reviewers can write reviews. Administrators can do most anything. The accounts (username and passwords are the same value) for the application are:

* **Registered Users:** user1, user2 and user3
* **Reviewers:** reviewer1, reviewer2 and reviewer3
* **Administrator:** admin

WebAPI Controller basics

In this part of the lab, you will implement the GET methods for a WebAPI controller to return movie data.

Criteria:

* Modify the WebAPI controller to provide movie listings and movie details (via GET)

Steps:

1. This lab is meant to be a very basic introduction to using WebAPI controllers to generate data. The intent is to become familiar with the movie review code base and use your first WebAPI controller.

The labs have both a client and a server project. Start with the server:

* 1. Open the server from ~/WebAPIIntro/before/service.
  2. Familiarize yourself with the WebAPI controller code in the ~/api directory.
  3. Feel free to run the web application and browse around.

1. Now implement the two “GET” methods: One for returning all the movie listings and another that returns a single movie based upon the “id” URL parameter.
   1. The MovieController is in ~/api/MoviesController.cs.
   2. The MovieService member variable provides the APIs needed. It returns Movie objects – just use these as the content to return from the service.
   3. To test, you can either access the service from the browser or from the pre-built client. The client project is located in ~/WebAPIIntro/before/client.

**Helpful links:**

* + [Your First ASP.NET WebAPI](http://www.asp.net/web-api/overview/getting-started-with-aspnet-web-api/tutorial-your-first-web-api)

Using Fiddler

In this part of the lab, you will install and use Fiddler to view and make HTTP requests.

Criteria:

* Use Fiddler to view HTTP requests
* Use Fiddler to make HTTP requests

Steps:

1. Fiddler is an excellent tool for debugging and monitoring HTTP traffic. In this part, you will install Fiddler (if it’s not already) and then use it to view and make HTTP requests to the movie review service.
   1. If it’s not already installed, download Fiddler from [here](https://www.fiddler2.com/Fiddler2/version.asp) and run the installer.

**Helpful Links:**

* If you have issues with Fiddler, [here](http://www.fiddler2.com/Fiddler/help/) is the help and documentation page.

1. Next use Fiddler to monitor browser traffic.
   1. Launch fiddler and then use your browser to access the movie listings.
   2. Inspect the details of the various requests in Fiddler.
2. Next use Fiddler to monitor HTTP traffic from the client application.
   1. Launch Fiddler and then use the pre-built client application to access the movie listings.

You probably won’t see any traffic in Fiddler due to the client bypassing the proxy when the URL is http://localhost, so change the client code to connect to http://localhost.fiddler instead and then you should be able to monitor the traffic. This change needs to be done in MoviesService.cs.

1. Finally, use Fiddler to make HTTP requests.
   1. Use the “Composer” feature of Fiddler to make HTTP requests to the movie service.
   2. In the composer set the Accept HTTP request header to use both application/xml and see the difference in the response.

The header should look like:

Accept: application/xml

Solutions:

The final solution for this lab is available in the ~/after directory.