**Understanding Route Prefix in Web APIs**

Attribute routing gives you more control over the URIs in your web API. For example, you can easily define a resource hierarchy using a single API controller.

The RoutePrefixAttribute is an extensibility point that allows you more control of how you define your routes/Urls.

**Defining a Common Prefix**

If you want to specify a common prefix for an entire controller, then instead of specifying [RoutePrefix]attribute on the top of each action, you should specify it in the controller level.

[RoutePrefix("Movie")]

public class HomeController : Controller

{

//Route: Movie/Index

public ActionResult Index()

{

ViewBag.Message = "You are in Home Index";

return View();

}

//Route: Movie/About

public ActionResult About()

{

ViewBag.Message = "You successfully reached USERS/About route";

return View();

}

}

**Overriding a Common Prefix**

There could be situations where you might not want your actions to be response under the same route prefix. In such cases, you will have to override the common route prefix. Doing so is easy.

Use a tilde (~) on the method attribute to override the route prefix.

[RoutePrefix("Movie")]

public class HomeController : Controller

{

//Route: Movie/Index

[Route]

public ActionResult Index()

{

ViewBag.Message = "You are in Home Index";

return View();

}

//Route: NewRoute/About

[Route("~/NewRoute/About")]

public ActionResult About()

{

ViewBag.Message = "You successfully reached NEWRoute/About route";

return View();

}

}

Attached is a sample application that demonstrates the usage of Route Prefix attribute.

Web API,Route,Custom Route,ASP.Net