




Module Review and Takeaways

To create a compelling web application that is easy to navigate, you must consider carefully the architecture of the information you present. You should try to ensure that the hierarchy of objects you present in URLs and navigation controls matches user expectations. You should also ensure that users can understand this hierarchy of objects without specialist technical knowledge. By using routes and site map providers, you can present logical information architecture to application visitors.

 **Best Practice:** The default route is logical. However, it requires knowledge of controllers and actions for users to understand URLs. You can consider creating custom routes that can be understood with information that users already have.

 **Best Practice:** You can use breadcrumb trails and tree view navigation controls to present the current location of a user, in the context of the logical hierarchy of the web application.

 **Best Practice:** You can use unit tests to check routes in the routing table, similar to the manner with which you use tests to check controllers, actions, and model classes. It is easy for a small mistake to completely change the way URLs are handled in your web application. This is because new routes, if poorly coded, can override other routes. Unit tests highlight such bugs as soon as they arise.

Common Issues and Troubleshooting Tips

Common Issue	Troubleshooting Tip
A route never takes effect.	

Review Question(s)

Verify the correctness of the statement by placing a mark in the column to the right.

Statement	Answer
You have implemented the MVC Site Map Provider in your web application and used it to build menus and breadcrumb trails with which users can navigate the logical hierarchy. MVC automatically takes routes from the MVC Site Map Provider so the same hierarchy is used in URLs.	

Question: You want to ensure that when the user specifies a relative URL in the form, “customer/3546”, the request is forwarded to the **DisplayById()** action in the **CustomerController**. You also want to ensure that when the user specifies a relative URL in the form, “customer/fullname”, the request is forwarded to the **DisplayByName()** action in the **CustomerController**. What routes should you add?