

- **Vendor: Microsoft**
- **Exam Code: 70-486**
- **Exam Name: Developing ASP.NET MVC (5.1) Web Applications**
- **New Questions (June/2018)**

**Visit PassLeader and Download Full Version 70-486 Exam Dumps**

**NEW QUESTION 234**

You are maintaining an ASP.NET Core MVC web application that runs on Azure. Remote debugging is enabled for this role, but the input endpoints for remote debugging have been removed for security reasons. You do not have permission to view the Azure Portal for this deployment. You can log on by using Remote Desktop Protocol (RDP). You must attach the input endpoints to enable remote debugging. You need to add the input endpoints to enable remote debugging. Which file should you modify?

- A. C:\Config\<GUID>.ccf
- B. E:\entrypoint.txt
- C. E:\<GUID>.csman
- D. C:\Config\<DeploymentD>.WebRole.1.xml

**Answer: C**

**NEW QUESTION 235**

You are developing an ASP.NET application that runs on Windows Server 2012. An exception is preventing a page from rendering. You need to view the trace information for the page. Which two actions should you perform? (Each correct answer presents part of the solution. Choose two.)

- A. Add the following markup segment to the web.config file: <trace mostRecent="true"/>.
- B. Load the trace.axd page from the root of the website.
- C. Add the traceEnable element to the Internet Information Service (IIS) Metabase section for the application.
- D. Add the following markup segment to the web.config file: <trace enabled="true"/>.

**Answer: BD**

**Explanation:**

For option D:

You can control whether tracing is enabled or disabled for individual pages. If tracing is enabled, when the page is requested, ASP.NET appends to the page a series of tables containing execution details about the page request. Tracing is disabled by default. To enable tracing for a page:

1. Include an @ Page directive at the top of your .aspx file.
2. Add a Trace attribute and set its value to true, as shown in the following example:

```
<%@ Page Trace="true" %>
```

You can also configure tracing in the Web.config file by setting the enabled, localOnly, and pageOutput attributes of the trace Element (ASP.NET Settings Schema).

For option B:

To view trace details for a specific request, navigate to Trace.axd in the root of your application. For example, if the URL for your application is <http://localhost/SampleApplication>, navigate to <http://localhost/SampleApplication/trace.axd> to view the trace information for that application. Select the View Details link for the request that you want to investigate.

<https://msdn.microsoft.com/en-us/library/94c55d08.aspx>

<https://msdn.microsoft.com/en-us/library/wwh16c6c.aspx>

#### **NEW QUESTION 236**

**Note:** This question is part of a series of questions that present the same scenario. Each question in the series contains a unique solution that might meet the stated goals. Some questions sets might have more than one correct solution, while others might not have a correct solution. After you answer a question in this section, you will NOT be able to return to it. As a result, these questions will not appear in the review screen.

You develop an ASP.NET web application that is self-hosted using Open Web Interface for .NET (OWIN) in a Microsoft Azure Worker role. The web application throws exceptions. You need to resolve the exceptions.

Solution: Use standard `HttpModule` and `HttpHandler` types.

Does the solution meet the goal?

A. Yes

B. No

**Answer: B**

**Explanation:**

Open Web Interface for .NET (OWIN) defines an abstraction between .NET web servers and web applications. OWIN decouples the web application from the server, which makes OWIN ideal for self-hosting a web application in your own process, outside of IIS-for example, inside an Azure worker role.

#### **NEW QUESTION 237**

**Note:** This question is part of a series of questions that present the same scenario. Each question in the series contains a unique solution that might meet the stated goals. Some questions sets might have more than one correct solution, while others might not have a correct solution. After you answer a question in this section, you will NOT be able to return to it. As a result, these questions will not appear in the review screen.

You develop an ASP.NET web application that is self-hosted using Open Web Interface for .NET (OWIN) in a Microsoft Azure Worker role. The web application throws exceptions. You need to resolve the exceptions.

Solution: Reference `System.Web.dll` to run in a custom host.

Does the solution meet the goal?

A. Yes

B. No

**Answer: B**

**Explanation:**

Traditional ASP.NET (up to MVC 5) is tightly coupled to IIS through `System.Web.dll`. ASP.NET Core provides a separation between the web server and your web application. This allows web applications to be portable between different web servers and also allows web servers to be self-hosted, which means you can start a web server in your own process, as opposed to a process that is owned by dedicated web server software such as IIS.

<https://docs.microsoft.com/en-us/azure/service-fabric/service-fabric-reliable-services-communication-aspnetcore>

**NEW QUESTION 238**

**Note:** This question is part of a series of questions that present the same scenario. Each question in the series contains a unique solution that might meet the stated goals. Some questions sets might have more than one correct solution, while others might not have a correct solution. After you answer a question in this section, you will NOT be able to return to it. As a result, these questions will not appear in the review screen.

You develop an ASP.NET Core MVC web application. You have a legacy business system that sends data to the web application by using Web API. The legacy business system uses proprietary data formats. You need to handle the proprietary data format.

Solution: Add an instance of a custom formatter class to the OutputFormatters collection in MVC. Does the solution meet the goal?

- A. Yes
- B. No

**Answer: B**

**Explanation:**

We need to add the custom formatter class to the InputFormatters collection in MVC.

<https://www.c-sharpcorner.com/article/custom-formatters-in-asp-net-core-mvc-web-api/>

**NEW QUESTION 239**

You define a startup task in the ServiceDefinition.csdef file. The task consists of a batch file that runs a Windows PowerShell script. The script places configuration files in local storage for use in a worker role. The worker role needs this information before starting. The worker role does not start after the startup task runs. You need to ensure that the worker role starts. What should you do?

- A. Use environment variables based on members of the RoleEnvironment class instead of static environment variables.
- B. Configure the task to use the directory specified by the TEMP environment variable.
- C. Ensure the task completes with an errorlevel of 0.
- D. Change the task from simple to foreground.

**Answer: C**

**Explanation:**

Startup tasks must end with an errorlevel (or exit code) of zero for the startup process to complete. If a startup task ends with a non-zero errorlevel, the role will not start. Startup tasks are actions that are taken before your roles begin and are defined in the ServiceDefinition.csdef file by using the Task element within the Startup element. Frequently startup tasks are batch files, but they can also be console applications, or batch files that start PowerShell scripts.

<https://docs.microsoft.com/en-us/azure/cloud-services/cloud-services-startup-tasks>

**NEW QUESTION 240**

You are developing an ASP.NET Core MVC web application that will be deployed to Microsoft Azure App Services Web App. Scheduled downtime during deployment of new features is not permitted. You need to ensure that deployments do not result in downtime. What should you do?

- A. Add additional upgrade domains.
- B. Use deployment slots during deployments.
- C. Convert the Web App to run in a standalone Docker container.
- D. Upgrade to a Premium App Service plan.

**Answer: B**

**Explanation:**

<https://blogs.msdn.microsoft.com/mvpawardprogram/2017/05/16/deploy-app-azure-app-service/>

**NEW QUESTION 241**

You are developing an ASP.NET Core MVC web application that provides assets to external websites including images, JavaScript scripts and text files. The external websites must be able to make full use of assets provided to them by the web application, including JavaScript asynchronous HTTP requests. Pages that refer to the images and text files do not load. External websites are unable to use assets. You need to resolve the issues. What should you do?

- A. Add a MVC controller and action to load the files from the folder.
- B. In the startup Configure method, call `IApplicationBuilder`.  
Use `StaticFiles` for the folder that contains the files.
- C. In the startup `ConfigureServices` method, call `IServiceCollection`.  
Add `DirectoryBrowser` for the folder that contains the files.
- D. Mark vendor provided files as content and set the value for `Copy to Output Directory` to `CopyAlways`.

**Answer: B**

**Explanation:**

<https://docs.microsoft.com/en-us/aspnet/core/fundamentals/middleware/?view=aspnetcore-2.1&tabs=aspnetcore2x>

**NEW QUESTION 242**

You are developing an application that uses ASP.NET Core Identity for authorization. The application must use an existing Microsoft Azure Table Storage instance to store user information. You create a custom `UserStore` class. You need to register the class as a dependency. Which two interfaces should you implement? (Each correct answer presents part of the solution. Choose two.)

- A. `IUserSecurityStampStore`
- B. `IUserLoginStore`
- C. `IQueryableUserStore`
- D. `IUserStore`
- E. `IUserPasswordStore`

**Answer: BD**

**Explanation:**

Create a `UserStore` class that provides the methods for all data operations on the user. This class is equivalent to the `UserStore` class. In your `UserStore` class, implement `IUserStore<TUser>` and the optional interfaces required. You select which optional interfaces to implement based on the functionality provided in your app. Interfaces to implement when customizing user store:

-- `IUserStore`:

The `IUserStore<TUser>` interface is the only interface you must implement in the user store. It defines methods for creating, updating, deleting, and retrieving users.

-- `IUserLoginStore`:

The `IUserLoginStore<TUser, TKey>` defines the methods you must implement in your user store to enable external authentication providers. It contains methods for adding, removing and retrieving user logins, and a method for retrieving a user based on the login information.

<https://docs.microsoft.com/en-us/aspnet/core/security/authentication/identity-custom-storageQQ-providers?view=aspnetcore-2.1>

**NEW QUESTION 243**

You are designing an ASP.NET Core MVC application that handles multiple customers. A user may log on to the site to perform activities such as checking balances, performing transactions,

and other activities that must be done securely. The application must store secure information that is specific to an individual user. The data must be automatically and securely purged when the user logs off. You need to save transient information in a secure data store. Which data store should you use?

- A. ASP.NET session state
- B. ASP.NET profile properties
- C. shared database
- D. ASP.NET application state

**Answer: C**

**NEW QUESTION 244**

You are designing a ASP.NET Core MVC application that runs on the Microsoft Azure platform. The application must store a small amount of information that is shared across all users and does not change frequently. You need to configure the application to meet the requirements. Which server-side state management options will achieve the goal? (Each correct answer presents a complete solution. Choose two.)

- A. HTTP Cookie
- B. Azure SQL Database
- C. ASP.NET session state
- D. ASP.NET application state

**Answer: BD**

**NEW QUESTION 245**

**Drag and Drop**

You deploy an ASP.NET MVC e-commerce application to a Microsoft Azure App Services Web App. Users report that the Orders page displays incorrect date and time information. You are unable to reproduce the issue locally. You need to configure Remote Debugging for the web application. Which three actions should you perform in sequence? (To answer, move the appropriate actions from the list of actions to the answer area and arrange them in the correct order.)

**Actions**

In Solution Explorer, right-click the web application and select **Publish**.

In the Visual Studio Debugger menu, select **Attach Debugger** and then select **DFService.exe (Azure Development Fabric Service)**.

In Server Explorer, right-click the web application and select **Attach Debugger**.

Navigate to the Orders page.

**Answer Area**



**Answer:**

**Actions**

In the Visual Studio Debugger menu, select **Attach Debugger** and then select **DFService.exe (Azure Development Fabric Service)**.

**Answer Area**

In Solution Explorer, right-click the web application and select **Publish**.

In Server Explorer, right-click the web application and select **Attach Debugger**.

Navigate to the Orders page.



**NEW QUESTION 246**

**Hotspot**

You are developing an ASP.NET Core MVC web application. The web application must meet the following requirements:

- Allow users to create a user name and password.
- Use cookie-based authentication.
- Store user credentials in a Microsoft SQL Server database.

You need to implement ASP.NET Core Identity. How should you complete the code? (To answer, select the appropriate options in the answer area.)

## Answer Area

```
public class Startup
{
    ...
    public void ConfigureServices(IServiceCollection services)
    {
        services.AddDbContext<ApplicationDbContext>(options =>
            options.UseSqlServer(Configuration.GetConnectionString("DefaultConnection")));

        services.AddAuthentication<ApplicationUser, IdentityRole>()
            .AddIdentity()
            .AddAuthorization();

        services.AddEntityFrameworkStores<ApplicationDbContext>()
            .AddDefaultTokenProviders();

        services.Configure<IdentityOptions>(options =>
        {
            options.Password.RequireDigit = true;
            options.Lockout.DefaultLockoutTimeSpan = TimeSpan.FromMinutes(30);
        });

        services.ConfigureApplicationCookie(options =>
        {
            options.Cookie.HttpOnly = true;
            options.Cookie.Expiration = TimeSpan.FromDays(90);
            options.SlidingExpiration = true;
        });
    }

    public void Configure(IApplicationBuilder app, IHostingEnvironment env)
    {
        ...
        app.UseAuthentication();
        ...
        app.UseCookieAuthentication();
        app.UseIdentity();
    }
}
```

**Answer:**



## Answer Area

```
public class Startup
{
    ...
    public void ConfigureServices(IServiceCollection services)
    {
        services.AddDbContext<ApplicationDbContext>(options =>
            options.UseSqlServer(Configuration.GetConnectionString("DefaultConnection")));

        services.AddAuthentication<ApplicationUser, IdentityRole>()
            .AddIdentity()
            .AddAuthorization();

        services.AddEntityFrameworkStores<ApplicationDbContext>()
            .AddDefaultTokenProviders();

        services.Configure<IdentityOptions>(options =>
        {
            options.Password.RequireDigit = true;
            options.Lockout.DefaultLockoutTimeSpan = TimeSpan.FromMinutes(30);
        });

        services.ConfigureApplicationCookie(options =>
        {
            options.Cookie.HttpOnly = true;
            options.Cookie.Expiration = TimeSpan.FromDays(90);
            options.SlidingExpiration = true;
        });
    }

    public void Configure(IApplicationBuilder app, IHostingEnvironment env)
    {
        ...
        app.UseAuthentication();
        app.UseCookieAuthentication();
        app.UseIdentity();
    }
}
```

### Explanation:

<https://docs.microsoft.com/en-us/aspnet/core/security/authentication/identity?view=aspnetcore-2.1&tabs=visual-studio%2Caspnetcore2x>

### NEW QUESTION 247

.....

[Visit PassLeader and Download Full Version 70-486 Exam Dumps](#)