DOWNLOAD PDF MVC 2.0 TUTORIAL FOR BEGINNERS IN NET

Chapter 1 : calendrierdelascience.com Core MVC & Razor Pages for Beginners: How to Build a Website â

The new tutorial uses calendrierdelascience.com Core MVC, which provides many improvements over this tutorial. This tutorial teaches calendrierdelascience.com Core MVC with controllers and views. Razor Pages is a new alternative in calendrierdelascience.com Core , a page-based programming model that makes building web UI easier and more productive.

A background to ASP. Is a composition of Model, View and Controller. Is an open source and the code is available in Code Plex In it was released under the Microsoft Public License and in it was released under the Apache License 2. Associated views and the controller are notified by the Model as in changes of its state. The Model provides a specific result as an output upon the request of the View to represent: In simple words, we can say that MVC is providing a facility to make our layers separate that are separated and interacting with each other. Net for web development over ASP. No view state performance is better in ASP. Integration with the client side easily handshakes with client-side scripting like jQuery and so on. There are two classes for one thing. A designer page has all the server control and aspx pages, that contain visualization of our server controls UI part and finally our cs page class contains all the functionality and other logic. Our class is based on an ASP. NET Page lifecycle for complete details refer to: NET MVC framework that already uses this pattern and provides everything required to create web apps? Overview of Model View Controller "This section of the session was a part of great discussion, what we learned until now. Please comment in this article what we discussed all queries and so on. Both the controller and view can access the Model. A Model can be used to pass data from a Controller to the View. The main purpose of a View is to display the data in a page using the Model. Within this class, methods can be implemented and are called action methods. The Controller class can access and use the Model class to pass data to the Views. The first request comes to the Route Tables then passes to a specific controller before interaction with the Model Binding. It goes through Authentication and Authorization and afterwards fires a specific Action Method and executes the results to the View. Overview of routes collection In simple words, the routes collection we also can say Route table is nothing but a collection of routes. Routes A system works on a pattern matching mechanism. A route matches an incoming request to meet a specific pattern and allows once matched else denied. What happens at runtime At runtime, the routing engine uses a route table contains various routes to match incoming URLs with the defined route patterns URL patterns. Example of typical route collections:

Chapter 2 : calendrierdelascience.com MVC 2 | The calendrierdelascience.com Site

These tutorials are designed for beginners and professionals who want to learn calendrierdelascience.com MVC 5 step by step. Prerequisites Basic knowledge calendrierdelascience.com Framework/, C#, Visual Studio is required.

Razor Pages is a page-based programming model that makes building web UI easier and more productive. The Razor Pages tutorial: Is easier to follow. For example, the scaffolding code has been significantly simplified. Provides more EF Core best practices. Uses more efficient queries. Is more current with the latest API. Is the preferred approach for new application development. If you choose this tutorial over the Razor Pages version, let us know why in this GitHub discussion. Razor Pages is a new alternative in ASP. If you choose this tutorial over the Razor Pages version, let us know why in this GitHub issue. The sample application is a web site for a fictional Contoso University. It includes functionality such as student admission, course creation, and instructor assignments. This is the first in a series of tutorials that explain how to build the Contoso University sample application from scratch. Download or view the completed application. For information about how to choose between EF 6. If you choose EF 6. Note For the ASP. Windows, Linux, or macOS: NET and web development workload. For a list of common errors and how to solve them, see the Troubleshooting section of the last tutorial in the series. Tip This is a series of 10 tutorials, each of which builds on what is done in earlier tutorials. Consider saving a copy of the project after each successful tutorial completion. Then if you run into problems, you can start over from the previous tutorial instead of going back to the beginning of the whole series. Users can view and update student, course, and instructor information. Enter ContosoUniversity as the name and click OK. Wait for the New ASP. This tutorial requires ASP. Make sure Authentication is set to No Authentication. Click OK Set up the site style A few simple changes will set up the site menu, layout, and home page. Change each occurrence of "ContosoUniversity" to "Contoso University". There are three occurrences. Add menu entries for Students, Courses, Instructors, and Departments, and delete the Contact menu entry. The changes are highlighted. This package is included in the Microsoft. This package and its dependencies Microsoft. Relational provide runtime support for EF. For information about other database providers that are available for Entity Framework Core, see Database providers. In other words, a student can be enrolled in any number of courses, and a course can have any number of students enrolled in it. The Student entity In the Models folder, create a class file named Student. The Enrollments property is a navigation property. Navigation properties hold other entities that are related to this entity. In this case, the Enrollments property of a Student entity will hold all of the Enrollment entities that are related to that Student entity. The Enrollment entity In the Models folder, create Enrollment. Ordinarily you would choose one pattern and use it throughout your data model. Here, the variation illustrates that you can use either pattern. The Grade property is an enum. The question mark after the Grade type declaration indicates that the Grade property is nullable. The StudentID property is a foreign key, and the corresponding navigation property is Student. An Enrollment entity is associated with one Student entity, so the property can only hold a single Student entity unlike the Student. Enrollments navigation property you saw earlier, which can hold multiple Enrollment entities. The CourseID property is a foreign key, and the corresponding navigation property is Course. An Enrollment entity is associated with one Course entity. The Course entity In the Models folder, create Course. A Course entity can be related to any number of Enrollment entities. Basically, this attribute lets you enter the primary key for the course rather than having the database generate it. Create the Database Context The main class that coordinates Entity Framework functionality for a given data model is the database context class. You create this class by deriving from the Microsoft. In your code you specify which entities are included in the data model. You can also customize certain Entity Framework behavior. In this project, the class is named SchoolContext. In the project folder, create a folder named Data. In the Data folder create a new class file named SchoolContext. In Entity Framework terminology, an entity set typically corresponds to a database table, and an entity corresponds to a row in the table. The Entity Framework would include them implicitly because the Student entity references the Enrollment entity and the Enrollment entity references the Course entity. When the database is created, EF creates tables that have names the same as the DbSet property

names. Property names for collections are typically plural Students rather than Student , but developers disagree about whether table names should be pluralized or not. To do that, add the following highlighted code after the last DbSet property. ToTable "Course" ; modelBuilder. ToTable "Enrollment" ; modelBuilder. NET Core implements dependency injection by default. Services such as the EF database context are registered with dependency injection during application startup. Components that require these services such as MVC controllers are provided these services via constructor parameters. To register SchoolContext as a service, open Startup. GetConnectionString "DefaultConnection" ; services. For local development, the ASP. NET Core configuration system reads the connection string from the appsettings. Add using statements for ContosoUniversity. EntityFrameworkCore namespaces, and then build the project. EntityFrameworkCore; Open the appsettings. By default, LocalDB creates. Add code to initialize the database with test data The Entity Framework will create an empty database for you. In the Data folder, create a new class file named DbInitializer. Models; using System; using System. Get a database context instance from the dependency injection container. Call the seed method, passing to it the context. Dispose the context when the seed method is done. LogError ex, "An error occurred while seeding the database. Data; In older tutorials, you may see similar code in the Configure method in Startup. We recommend that you use the Configure method only to set up the request pipeline. Application startup code belongs in the Main method. Now the first time you run the application, the database will be created and seeded with test data. Whenever you change your data model, you can delete the database, update your seed method, and start afresh with a new database the same way. The automatic creation of CRUD action methods and views is known as scaffolding. When you need to customize generated code, you use partial classes or you regenerate the code when things change. Update Visual Studio to the latest version. Visual Studio versions prior to In the Add Scaffold dialog box: In the Add Controller dialog box:

Chapter 3 : calendrierdelascience.com MVC Series For Beginners: Part 1

Step-by-step calendrierdelascience.com MVC Tutorial for Beginners Programming with Mosh. Building Web APIs with calendrierdelascience.com Core - Duration: (Model view controller) step by step in 2 days (16 hours).

But the truth is ASP. NET is a framework for creating web application while MVC is a great architecture to organize and arrange our code in a better way. Ok so if the new thing is ASP. Let me correct your vocabulary: NET Webforms has served and successfully delivered web application for past 12 years. If you see the success of Microsoft programming languages right from the days of VB visual basic it is due to RAD Rapid application development and visual programming approach. By using visual studio ,developers where able to drag drop UI elements on a designer area and at the backend , visual studio generates C or VB. NET code for those elements. In this code behind Developers can go and write logic to manipulate the UI elements. CS and so on. The main problem with ASP. NET Webform is performance, performance and performance. In web application there are two aspects which define performance: Let us try to understand why response time is slower when it comes to ASP. We did a small load testing experiment of Webform vs Asp. Net MVC and we found Asp. Net MVC to be twice faster. Read more on how this test was done from here Let us try to understand why ASP. Assume the ASPX code has the below simple text box. If you see the HTML output by doing view source it looks something as shown below. Try to get answers to the below questions: Do we really need to make those long server trips to get those simple HTML on the browser?. If you see for every request there is a conversion logic which runs and converts the server controls to HTML output. Due to this unnecessary conversion the response time get affected. Solution for this problem: Bandwidth consumption Viewstate has been a very dear and near friend of ASP. NET developers for past 10 years because it automatically saves states between post backs and reduces our development time. But this reduction in development time comes at a huge cost ,viewstate increases the page size considerably. In this load test we found viewstate increases the page size twice as compared to Asp. Below is the plot of the content length emitted from Webform and Asp. The size increase is because of extra bytes generated from viewstate , below is the snapshot of a viewstate. Lot of people can argue that viewstate can be disabled but then we all know how developers are , if there is option given they would definitely try that out. But the main thing is always performance. If you run the above code below are the respective generated HTML. The other great benefit of working directly with HTML is that your web designers can work very closely with the developer team. They can take the HTML code put in their favourite designer tool like dream weaver , front page etc and design independently. If we have server controls these designer tools do not identify them easily. Reusability of code behind class If you watch any professional ASP. NET Webform project you will notice that code behind class is where you have huge amount of code and the code is really complicated. This class is not a normal class which can be reused and instantiated anywhere. In other words you can never do something as shown below for a Webform class: From the code you can know how difficult it is to instantiate the same. Someone has to manually run the application and do the testing. If we read the four issues mentioned in the previous section with ASP. Below is root cause diagram I have drawn. In this I started with problems , what is the cause for it and the solution for the same. The solution is we need to move the code behind to a separate simple class library and get rid of ASP. In short the solution should look something as shown in the below image. As said the code behind and server controls are the root cause problem. NET classes which you can term as middle layer , business logic etc and the middle layer talks with data access layer. The code behind logic goes in to the controller. View is your ASPX i. You can see in the above diagram how those layers fit in. NET classes termed as controller. Net MVC request flow in general moves as follows: Model in turn calls the data access layer which fetches data in the model. Now that we have understood the different components of Asp. Let us first start with controllers as they are the most important and central part of the MVC architecture.

Chapter 4 : calendrierdelascience.com Core Books | calendrierdelascience.com | Page 2

You will learn calendrierdelascience.com Core by building three applications. The first application will be built using an empty template, the second with the MVC template, and the third with Razor Pages.

The new tutorial uses ASP. This tutorial teaches ASP. Razor Pages is a new alternative in ASP. The Razor Pages tutorial: Is easier to follow. Is the preferred approach for new application development. This tutorial teaches you the basics of building an ASP. The final source code for the tutorial is located on GitHub. This tutorial was written by Scott Guthrie twitter scottgu , Scott Hanselman twitter: Get started Start by installing Visual Studio Then, open Visual Studio. Visual Studio is an IDE, or integrated development environment. Create your first app On the Start page, select New Project. NET Framework project template. Name your project "MvcMovie" and then choose OK. In the New ASP. Visual Studio used a default template for the ASP. NET MVC project you just created, so you have a working application right now without doing anything! This is a simple "Hello World! Press F5 to start debugging. Notice that the address bar of the browser says localhost: When Visual Studio runs a web project, a random port is used for the web server. In the image below, the port number is Right out of the box this default template gives you Home, Contact, and About pages. Depending on the size of your browser window, you might need to click the navigation icon to see these links. The application also provides support to register and log in. The next step is to change how this application works and learn a little bit about ASP. For a list of current tutorials, see MVC recommended articles. See this app running on Azure Would you like to see the finished site running as a live web app? You can deploy a complete version of the app to your Azure account by simply clicking the following button. You need an Azure account to deploy this solution to Azure. Activate Visual Studio subscriber benefits - Your Visual Studio subscription gives you credits every month that you can use for paid Azure services.

Chapter 5 : calendrierdelascience.com Web API Tutorials

The newest and best calendrierdelascience.com MVC version is here, and it is Core Learn it in hours!. The newest Core version has many changes compared to MVC 5, so take this no-nonsense and to the point course to learn everything you need to know in the shortest amount of time possible.

In this article, we will try to understand the concept of middleware in ASP. We will see how middleware plays an important part in request response pipeline and how we can write and plug-in our custom middleware. Background Before we could get into the what middleware is and the value it brings, we need to understand how the request response worked in classic ASP. In earlier days the request and response objects in ASP. This was a problem because some of the values in these objects are filled by the IIS request-response pipeline and unit testing such bloated objects was a very big challenge. So the first problem that needed to be solved was to decouple the applications from web servers. This was very nicely defined by a community owned standards called Open Web Interface for. Since the older ASP. NET applications were dependent on System. Web DLL which internally had very tight coupling with IIS, it was very difficult to decouple the applications from web servers. OWIN primarily defines following actors in its specifications: Servers may require an adapter layer that converts to OWIN semantics. Middleware â€" Pass through components that form a pipeline between a server and application to inspect, route, or modify request and response messages for a specific purpose. Host â€" The process an application and server execute inside of, primarily responsible for application startup. Some Servers are also Hosts. Since OWIN is just a standard, there are multiple implementation for this in last few years starting from Katana to the present day implementation in ASP. We will now focus on how the middleware implementation looks like in ASP. Before that, lets try to understand what a middleware is. For the developer coming from ASP. These are used to intercept the request-response pipeline and implement our custom logic by writing custom modules or handlers. In the OWIN world, the same thing is achieved by the middleware. OWIN specifies that the request coming from web server to the web application has to pass through multiple components in a pipeline sort of fashion where each component can inspect, redirect, modify or provide a response for this incoming request. The response then will get passed back to the web server in the opposite order back to the web server which then can be served back to the user. Following image visualizes this concept: If we look at the above diagram we can see that the request passes through a chain of middleware and then some middleware decides to provide a response for the request and then response travels back to web server passing through all the same middleware it passed through while request. So a middleware typically can: Process the request and generate the response. Monitor the request and let it pass thorough to next middleware in line. Monitor the request, Modify it and then let it pass through to next middleware in line. If we try to find the middleware with actual use cases defined above: Process the request and generate the response: MVC itself is a middleware that typically gets configured in the very end of middleware pipeline Monitor the request and let it pass through to next middleware in line: Logging middleware which simply logs the request and response details Monitor the request, Modify it and then let it pass through to next middleware in line: Routing and Authentication module where we monitor the request decide which controller to call routing and perhaps update the identity and Principle for authorization Auth-Auth. Using the code In this article, we will create 2 owin middleware. First one to demonstrate the scenario where we are not altering the request. Second one to check the incoming response, find a specific header value to determine which tenant is calling the code and then returning back if the tenant is not valid â€" MyTenantValidator. Before we get started with the sample implementation, its good to highlight the point that middleware is an implementation of pipes and filter pattern. Pipes and filter pattern says that if we need to performs a complex processing that involves a series of separate activity, its better to separate out each activity as a separate task that can be reused. This gives us benefits in terms of reusability, performance and scalability. Lets start by looking at how the middleware class definition should look like. There are two ways to define our custom middleware:

Chapter 6 : calendrierdelascience.com Core MVC with Entity Framework Core - Tutorial 1 of 10 | Microsoft

calendrierdelascience.com MVC Series For Beginners: Part 1 released under the Apache License). We can simply say that calendrierdelascience.com MVC is a framework that provides us the ability to.

Chapter 7 : Tutorial for visual C# calendrierdelascience.com MVC 2 - Stack Overflow

Our calendrierdelascience.com tutorials can help you create a dynamic website that includes web form pages, external CSS pages, and SQL server data. Start watching and learn how to develop a web application from.

Chapter 8 : Getting Started with calendrierdelascience.com MVC 5 | Microsoft Docs

calendrierdelascience.com Web API Tutorials calendrierdelascience.com Web API is a framework for building HTTP services that can be accessed from any client including browsers and mobile devices. It is an ideal platform for building RESTful applications on calendrierdelascience.com Framework.

Chapter 9 : calendrierdelascience.com MVC Tutorials

Lot of calendrierdelascience.com developers who start MVC for the first time think that MVC is different new, fresh from calendrierdelascience.com But the truth is calendrierdelascience.com is a framework for creating web application while MVC is a great architecture to organize and arrange our code in a better way.