Modulo 1 - Introdução as ASP NET MVC

Ano 2000 - Criação do Asp Net usando a tecnologia Web Forms.

Ano 2009 - Surgimento ASP .NET MVC

Vantagens

- Facilidade de desenvolvimento;
- Mais leve se comparado com o Asp NET Forms por não possuir view state;
- Conseguimos ter controle sobre o que mandamos para o browser;

Desvantagem

- "Peso" da aplicação tanto no cliente quanto no servidor.
- Conhecimento especializado no ASP .NET MVC

Algumas definições:

Comunicação client/server:

Protocolo HTTP.

GET - Obter dados do servidor;

POST - Enviar dados para o servidor;

PUT -

DELETE -

View State - Característica que auxilia o desenvolvedor.

ASP .NET

Para cada página é criada uma classe correspondente. Quando o usuário clica em algum botão, a página é submetida ao servidor IIS que verifica a extensão do arquivo e busca pela classe no diretório de arquivos.

M - Model -> Sua responsabilidade é representar o dado que será trocado entre a view e p controller. Classe que contém propriedades e sua função é passar dados entre o COntroller e a View e vice-versa.

V - View -> É a página que o usuário visualiza

C - Controller -> Sua responsabilidade é receber a ação do usuário e direcionar a view que será exibida para o usuário.

O método View() é herdado da Classe Controller. Ele possui 4 sobrecargas, e por default ele devolve a View que possui o mesmo do da Action

Ciclo de vida de uma aplicação ASP .NET MVC

POST

1 - Envio do form HTML para o controller;

2 - Transformação dos dados do form em Model;

3 - Persistência dos dados da Model na base de dados.

GET

1 - Envio do form para o controller;

2 - Controller efetua a pesquisa na base de dados;

4 - Contoller recebe a resposta do banco de dados;

5 - Controller tranforma os dados em model;

6 - Model é passada para a view ;

7 - Envio da resposta em forma de HTML para o browser;

Persistir em Model em Banco de Dados

Qual técnica ?

1. Usando ADO.NET (Aumenta o tempo de trabalho/alocação de recursos)
2. Usando Ferramenta de Mapeamento Objeto Relacional(redução de esforço/adptação à metodologia de trabalho das feramentas)

   Ex.: Entity Framework disponível a partir do .NET 3.5

   Linq to SQL - legado.

Padrão Repository

Facilita manutenção futura na camada de dados, ex: troca de tecnologia da base de dados.

Padroniza o acesso à base de dados.

O padrão repository é formado por 3 partes, sendo elas:

Interface de repositório(operações que deseja-se realizar na base de dados com o model);

Objeto que será persistido (Model no caso de projetos MVC);

Classe que implementa a interface(Única para cada model);

Principais classes do ADO .NET (SQL Server)

SqlConnetion -Estabelece conexão com a base de dados

ConectionString;

Open();

Close();

SqlCommand

Connection; => recebe um objeto SqlConnection(composisção)

Parammeters;

ExecuteNonQuery(); => Executa qualquer consulta que não retorne linha

ExecuteReader(); => Para retornar registros; (retorna um SqlDataReader)

ExecuteScalar(); =>

SqlDataReader

Read();

Close();

["nomedacoluna"]

Obs.: Sempre abrir a conexão mais tarde possível e fecha-la o mais cedo possível.

Formas de transferir dados entre View/Controller

Passagem de Parâmetros

- Request.Form["form"];
- FormCollection["inputname];
- Model Binder => Usa-se com parametro da tela o Model Ex.:IncluirCliente(Cliente cliente)

Formas de transferir dados entre Controller/View

- ViewBag - Utiliza propertiedades dinâmicas. Ex.: ViewBag.Mesagem = "Cliente cadastrado com sucesso';
- ViewData - é uma collection de object Ex.: ViewData["mensagem"] = "Cliente Cadastrado com sucesso";
- View Tipada - é quando a view consegue trabalhar com um model naturalmente;
- Através do mpetodo View()

EntityFramework

Ferramenta de Mapeamento objeto Relacional.

Model First

DataBase First

CodeFirst

Classe Model (Cliente)

DBContext -> Executa comandos na base de dados;

Injeção de dependência

Usando esse padrão a sua aplicação trabalha de forma desacoplada. A injeção de dependencia é muito usada para desacoplar camadas.

Testes

Teste Unitário

Características:

-Usado para testar métodos

-Permite criar cenários positivos e negativos

-Pode ser usado no ASP .NET MVC para testar as actions dos controlers

Exemplo:

Aplicação bancária

Cliente

- Codigo
- Nome

ContaCorrente

- Numero
- Titular
- Saldo
- Depositar (valor)
- Sacar (valor)

Gerenciamento de Estado no ASP .NET

Server

- Application
- Session

Client

- Cookie

Application:

- Armazena object;
- Está disponível para toda a aplicação
- É única para todos os usuários
- Nunca expira

Session

- Armazena object;
- Está disponível para toda aplicação
- Cada usuário possui a sua Session
- Por padrão expira aos 20 minutos de inatividade do cliente

Quando a aplicação ASP .NET inicia?

1) Um processo(*w3wp.exe*) é criado no servidor web (IIS);

2) As configurações do *web.config* são colocadas em memória;

3) Eventos do Global.asax são disparados;

- Application_Start
- Application_End - Disparado quando o processo do Asp.NET(*w3wp.exe*) é encerrado.
    - Parar o IIS;
    - Parar o ApplicationPool;
    - Quando alterar a DLL da Aplicação ASP.Net
    - Quando alterar o web.config;
- Session_Start
- Session_End
    - Quando a sessão expira;
    - Via Session.Abandon();

Cliente

- Cookies
    - Persistentes
    - Em memória

Armazenam informações usando o conceito de chave e valor.
Um cookie não pode armazenar mais que 4kb de informações.

Deploy

Application Pool

Passos para deploy

1. Criar Application Pool, caso necessário.
2. Definir a pasta que vaio acomodar os arquivosd da aplicação
3. Publicar a aplicação pelo Visual Studio
4. Configurar a aplicação no IIS

Solicitações AJAX (Assincronus Java Script And XML)

- Permite o envio de dados ao servidor sem o envolvimento do browser, sem o uso de form html;
- Diminui o tráfego de dados entre o cliente e o servidor;
- Diminui o processamento do servidor e do cliente;

WCF

Troca de informações

- Soap - XML
- Binário
- Fila

Transacional

SOA

Para construir um WCF:

1. Contrato;
2. Implementação do Contrato;
3. EndPoint;
4. Host;

1) Contrato

É no contrato que estão as operações que o ser4viço oferece.(Interface C#);

2) Implementação

É o serviço implementado respeitando o contexto;

3)EndPoint

- *Adress* - Endereço do serviço;
- *Binding* - Define o protocolo de comunicação e o formato de dados;
- *Contract* - Define o contrato que será exposto pelo endpoint;
- *Host* - Onde o Serviço fica hospedado (IIS);

O cliente que vai consumir um WCF deve montar um proxy. As chamadas do WCF são feitas através desse proxy.

ASP NET WEB API

Características:

- Baseado na arquitetura REST
- Pode ser consumido por qualquer cliente. Até mesmo uma aplicação HTML;
- Não é transacional;
- Só trabalha dom protocolo HTTP

HTTP

1) GET - Obter dados
2) POST - Inserção de dados
2) PUT - Atualizar dados
4) DELETE - Excluir dados

Rota: Diz qual o formato de endereçamento deve ser usado pelo cliente, para acessar um serviço Web API ou uma aplicação ASP .NET MVC.

Autenticação

Windows

- Basic
    - Pode ser usado com ou sem domínio;
- Digest
    - Cria um hash com usuário e senha;
    - Funciona somente Internet Explorer;
    - Funciona somente em ambientes com domínio;

Forms

- Cookies
- Filtro de actions