

Aula 11 – Ajax

Disciplina: Programação Web

Prof. Allbert Velleniche de Aquino Almeida

E-mail: allbert.almeida@fatec.sp.gov.br

Site: <http://www.allbert.com.br>



/allbert.almeida

Objetivo

- O objetivo da aula é chamar métodos ASP .NET do lado do servidor a partir do cliente, usando JQuery com AJAX.

Conceito

- AJAX significa Asynchronous JavaScript and XML, ou JavaScript e XML Assíncronos, em bom português. Ele é um conjunto de técnicas de desenvolvimento voltado para a web que permite que aplicações trabalhem de modo assíncrono, processando qualquer requisição ao servidor em segundo plano;

Adicionar 2 pacotes para solução

- EntityFramework
- MySql Data Entity 6.9.11

EntityFramework por Microsoft

v6.2.0



Entity Framework is Microsoft's recommended data access technology for new applications.



MySql.Data.Entity por Oracle

v6.10.6



Entity Framework 6.0 supported

Adicionar a String de Conexão no WebConfig

```
<connectionStrings>
```

```
    <add name="StringConexao"  
providerName="MySql.Data.MySqlClient"  
connectionString="server=localhost;dat  
abase=aulaajax;uid=root;pwd='root'" />
```

```
</connectionStrings>
```

Modelo

Usuario

Id

Nome

Email

Status

Criar uma classe Usuario

```
public class Usuario
{
    public int Id { get; set; }
    [Required]
    public string Nome { get; set; }
    [EmailAddress]
    public string Email { get; set; }
    public bool Status { get; set; }
}
```

Criar uma classe Contexto

```
public class Contexto:DbContext
{
    public Contexto() :
base(nameOrConnectionString:
"StringConexao") { }

    public DbSet<Usuario> Usuario { get;
set; }
}
```


Compilar a solução

- Compilar a solução para que o modelo seja validado e seja possível criar o “CRUD” da classe Usuario

View Create - Usuario

- Inserir a tag span abaixo ValidationMessageFor do Email:

```
<span id="msg"></span>
```

- Inserir o atributo no submit (botão):

```
id="botao"
```

```
disabled="disabled"
```

View Create - Usuario

```
@Scripts.Render("~/bundles/jquery")
<script>
    $("#Email").focusout(function () {
        var email = $(this).val();
        $.ajax({
            dataType: "json",
            type: "POST",
            url: "/Usuarios/ValidarEmail",
            data: { email },
            success: function (retorno) {
                if (retorno == "s") {
                    $("#botao").attr('disabled', 'disabled');
                    $("#msg").text("E-mail já cadastrado");
                    $("#msg").removeAttr("class");
                    $("#msg").attr("class", "text-danger");
                }
                else {
                    $("#botao").removeAttr("disabled");
                    $("#msg").text("E-mail validado");
                    $("#msg").removeAttr("class");
                    $("#msg").attr("class", "text-success");
                }
            }
        });
    });
</script>
```

Controller Usuario

```
[AcceptVerbs(HttpVerbs.Post)]
[ValidateInput(false)]
public JsonResult ValidarEmail(string email) {
    Usuario u = db.Usuario.Where(t=>t.Email ==
email).FirstOrDefault();
    if (u != null) {
        return Json("s");
    } else {
        return Json("n");
    }
}
```

Compilar a solução

- Compilar a solução e testar se está sendo validado o e-mail em duplicidade;

View Index - Usuario

- Trocar a tag DisplayFor do Status para:

```
@Html.CheckBox("chk",item.Status, new{  
@class="status",@rel=@item.Id})
```

View Index - Usuario

- Adicionar nova coluna na tabela:

```
<td>
@if (item.Status) {
    <span id="status_@item.Id" class="text-success"
style="font-weight:bold">Ativo</span>
} else {
    <span id="status_@item.Id" class="text-danger"
style="font-weight:bold">Inativo</span>
}
</td>
```

View Index - Usuario

```
@Scripts.Render("~/bundles/jquery")
<script>
    $(".status").click(function () {
        var id = $(this).attr("rel");
        $.ajax({
            dataType: "json",
            type: "POST",
            url: "/Usuarios/AlterarStatus",
            data: { id },
            success: function (retorno) {
                if (retorno == "t") {
                    $(this).removeAttr('disabled');
                    $("#status_" + id).removeClass();
                    $("#status_" + id).addClass("text-success");
                    $("#status_" + id).text("Ativo");
                }
                else if (retorno == "f") {
                    $(this).attr('disabled', 'disabled');
                    $("#status_" + id).removeClass();
                    $("#status_" + id).addClass("text-danger");
                    $("#status_" + id).text("Inativo");
                }
            }
        });
    });
</script>
```


Controller Usuario

```
[AcceptVerbs(HttpVerbs.Post)]
[ValidateInput(false)]
public JsonResult AlterarStatus(string id) {
    Usuario u = db.Usuario.Find(Convert.ToInt32(id));
    if (u != null) {
        if (u.Status)
            u.Status = false;
        else
            u.Status = true;
        db.Entry(u).State = EntityState.Modified;
        db.SaveChanges();
        return Json(u.Status?"t":"f");
    } else {
        return Json("n");
    }
}
```

Compilar a solução

- Compilar a solução e testar se está sendo alterado o status do usuário;

View Index - Usuario

- Adicionar id na linha:

```
<tr id="linha_@item.Id">
```

- Adicionar depois de "delete":

```
| <a class="excluir" rel="@item.Id"><i  
class="glyphicon glyphicon-trash"></i></a>
```

View Index - Usuario

```
<script>
    $(".excluir").click(function () {
        var id = $(this).attr("rel");
        $.ajax({
            dataType: "json",
            type: "POST",
            url: "/Usuarios/ExcluirUsuario",
            data: { id },
            success: function (retorno) {
                if (retorno == "ok") {
                    var linha = $("#linha_"+id); //tr
                    linha.remove();
                }
                else if (retorno == "n") {
                }
            }
        });
    });
</script>
```

Controller Usuario

```
[AcceptVerbs(HttpVerbs.Post)]
[ValidateInput(false)]
public JsonResult ExcluirUsuario(string id) {
    try {
        Usuario u = db.Usuario.Find(Convert.ToInt32(id));
        if (u != null) {
            db.Usuario.Remove(u);
            db.SaveChanges();
            return Json("ok");
        } else
            return Json("n");
    } catch {
        return Json("n");
    }
}
```

Compilar a solução

- Compilar a solução e testar se está sendo excluído o usuário;

View Index - Usuario

- Adicionar id nas colunas nome e e-mail:

```
<td id="nome_@item.Id">  
<td id="email_@item.Id">
```

- Adicionar depois do "excluir":

```
| <a onclick="editar(@item.Id)" data-  
toggle="modal" data-  
target="#ModalEditar"><i class="glyphicon  
glyphicon-edit"></i></a>
```

View Index - Usuario (Modal)

```
<div id="ModalEditar" class="modal fade" role="dialog">
  <div class="modal-dialog">
    <div class="modal-content">
      <div class="modal-header">
        <button type="button" class="close" data-dismiss="modal">&times;</button>
        <h4 class="modal-title">Editar Usuário</h4>
      </div>
      <div class="modal-body">
        <input type="hidden" id="edit_id" />
        <div class="form-group">
          <label class="control-label col-md-2">Nome</label>
          <div class="col-md-10">
            <input type="text" id="edit_nome" class="form-control" />
          </div>
        </div><br /><br />
        <div class="form-group">
          <label class="control-label col-md-2">E-mail</label>
          <div class="col-md-10">
            <input type="text" id="edit_email" class="form-control" />
          </div>
        </div>
        <br /><br />
        <div class="form-group">
          <div class="col-md-12">
            <span id="msgerro" class="text-danger"></span>
          </div>
        </div>
      </div>
      <div class="modal-footer">
        <button type="button" class="btn btn-default pull-left" data-
dismiss="modal">Fechar</button>
        <button type="button" class="btn btn-primary pull-right" id="salvar">Salvar</button>
      </div>
    </div>
  </div>
</div>
```


View Index - Usuario (Script)

```
<script>
  function editar(id) {
    $("#edit_id").val(id);
    $("#edit_nome").val($("#nome_" + id).text().trim());
    $("#edit_email").val($("#email_" + id).text().trim());
    $("#msgerro").text("");
  }
  $("#salvar").click(function () {
    var id = $("#edit_id").val();
    var nome = $("#edit_nome").val();
    var email = $("#edit_email").val();
    $.ajax({
      dataType: "json",
      type: "POST",
      url: "/Usuarios/EditarUsuario",
      data: { id, nome, email },
      success: function (retorno) {
        if (retorno == "ok") {
          $("#nome_" + id).text(nome);
          $("#email_" + id).text(email);
          $('#ModalEditar').modal('toggle');
        }
        else if (retorno == "erro") {
          $("#msgerro").text("Preencha o nome e cadastre e-mail válido");
        }
        else if (retorno == "n") {
          $("#msgerro").text("Usuário não encontrado");
        }
      }
    });
  });
</script>
```

Controller Usuario

```
[AcceptVerbs(HttpVerbs.Post)]
[ValidateInput(false)]
public JsonResult EditarUsuario(string id, string nome, string email) {
    try {
        Usuario u = db.Usuario.Find(Convert.ToInt32(id));
        if (u != null) {
            u.Nome = nome;
            u.Email = email;
            if (TryValidateModel(u)) {
                db.Entry(u).State = EntityState.Modified;
                db.SaveChanges();
                return Json("ok");
            } else
                return Json("erro");
        } else
            return Json("n");
    } catch {
        return Json("n");
    }
}
```

Compilar a solução

- Compilar a solução e testar se está sendo editado o usuário;