CRUD

Nesse exemplo vamos explorar as opções de CRUD (*Create Read Update e Delete*) da tabela Clientes. Nesse exemplo usamos os seguintes recursos:

- Servidor Node.js
- Metodologia MVC (Model View Controler)
- Retorno dos dados do banco no formato JSON

Estrutura do site

O site será estruturado da seguinte forma:

/	index.php	Página principal do site
/estilo	estilo.css	CSS do site
/imagens		Imagens do site
/js	alunos.js init.js materialize.js materialize.min.js	Arquivos Javascript para trabalhar com dados obtidos pelo Node.js e exibi-los no navegador. Além das bibliotecas do framework Materialize
http://localhost:3000	appSQL.js	Servidor Node.js implementado em uma API que implementa as operações CRUD no banco de dados SQL Server

A tabela usada nesse exemplo será a tabela Aluno com a seguinte estrutura:

```
CREATE TABLE Clientes (
ID INT(5) PRIMARY KEY NOT NULL  ,
Nome VARCHAR( 150 ) ,
CPF VARCHAR( 11 ) ) ;
```

Servidor Node.js

```
const express = require('express');
const app = express();
const bodyParser = require('body-parser');
const porta = 3000; //porta padrão
const sql = require('mssql');
const conexaoStr = "Server=regulus;Database=BD18000;User Id= BD18000;Password= BD18000;";

//conexao com BD
sql.connect(conexaoStr)
  .then(conexao => global.conexao = conexao)
  .catch(erro => console.log(erro));

// configurando o body parser para pegar POSTS mais tarde
app.use(bodyParser.urlencoded({ extended: true}));
```

```
app.use(bodyParser.json());
//acrescentando informacoes de cabecalho para suportar o CORS
app.use(function(req, res, next) {
  res.header("Access-Control-Allow-Origin", "*");
  res.header("Access-Control-Allow-Headers", "Origin, X-Requested-With, Content-Type, Accept");
  res.header("Access-Control-Allow-Methods", "GET, POST, HEAD, OPTIONS, PATCH, DELETE");
  next();
});
//definindo as rotas
const rota = express.Router();
rota.get('/', (requisicao, resposta) => resposta.json({ mensagem: 'Funcionando!'}));
app.use('/', rota);

//inicia servidor
app.listen(porta);
console.log('API Funcionando!');

function execSQL(sql, resposta) {
        global.conexao.request()
                                .query(sql)
                                .then(resultado => resposta.json(resultado.recordset))
      //.then(resultado => console.log(resultado.recordset))
                                .catch(erro => resposta.json(erro));
}

//o simbolo ? indica que id na rota abaixo é opcional
rota.get('/clientes/:id?', (requisicao, resposta) => {
        let filtro = '';
        if (requisicao.params.id)
                filtro = ' WHERE ID=' + parseInt(requisicao.params.id);
        execSQL('SELECT * from Clientes' + filtro, resposta);
})

// testar no POSTMAN
rota.delete('/clientes/:id', (requisicao, resposta) =>{
        execSQL('DELETE Clientes WHERE ID=' + parseInt(requisicao.params.id), resposta);
         resposta.end(resposta.json({ mensagem: 'Deletado!'}));
})

rota.post('/clientes', (requisicao, resposta) =>{
   const id = parseInt(requisicao.body.id);
   const nome = requisicao.body.nome.substring(0,150);
   const cpf = requisicao.body.cpf.substring(0,11);
   execSQL(`INSERT INTO Clientes(ID, Nome, CPF) VALUES(${id},'${nome}','${cpf}')`, resposta);
   resposta.end(resposta.json({ mensagem: 'Incluído!'}));
})

rota.patch('/clientes/:id', (requisicao, resposta) =>{
   const id = parseInt(requisicao.params.id);
   const nome = requisicao.body.nome.substring(0,150);
   const cpf = requisicao.body.cpf.substring(0,11);
   execSQL(`UPDATE Clientes SET Nome='${nome}', CPF='${cpf}' WHERE ID=${id}`, resposta);
   resposta.end(resposta.json({ mensagem: 'Alterado!'}));
})
```

Lógica do site

O arquivo index.html será o arquivo que exibirá a lista de usuários do banco de dados por padrão. Essa página possui a estrutura da tabela no HTML e as janelas modais para opções de inclusão e alteração.

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <link href="estilo/estilo.css" rel="stylesheet" type="text/css"/>
        <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>

        <title>CRUD</title>
    </head>
    <body>
        <div class="container">
        <div class="titulo">
                Cadastro de Usuários
            </div>
            <nav>
                <!-- Trigger/Open The Modal -->
                <a href="#!" id="btnInclusao">Inclusão</a>
            </nav>
        </div>
        <div class="container">
            <div class="dados">
            <table id="tblListar" class="highlight">
                <thead>
                  <tr>
                      <th data-field="acoes">-</th>
                      <th data-field="id">ID</th>
                      <th data-field="nome">Nome</th>
                      <th data-field="cpf">CPF</th>
                  </tr>
                </thead>

                <tbody>

                </tbody>
            </table>
            </div>

            <!-- Modal Inclusão -->
```

```
39              <div class="modal" id="modalInc">
40                  <div class="modal-content">
41                      <div class="modal-header">
42                          <span class="close">&times;</span>
43                          <h2>Incluir Cliente</h2>
44                      </div>
45                      <div class="modal-body">
46                          <form class="modal-form" action=""
method="" id="formCadastro">
47                              <label>ID:</label>
48                              <input type="text" name="id">
<br>
49                              <label>Nome:</label>
50                              <input type="text" name="nome">
<br>
51                              <label>CPF: </label>
52                              <input type="text" name="cpf">
<br>
53                          </form>
54                      </div>
55                      <div class="modal-footer">
56                          <a href="#!" id="btnCadastrar"
class="modal-link-rodape">CADASTRAR</a>
57                          <a href="#!" class="modal-link-
rodape" id="btnCancelar">CANCELAR / FECHAR</a>
58                      </div>
59                  </div>
60              </div> <!-- Fim Modal Inclusão -->
61
62              <!-- Modal ALteração -->
63              <div class="modal" id="modalAlt">
64                  <div class="modal-content">
65                      <div class="modal-header">
66                          <span class="close">&times;</span>
67                          <h2>Alterar Dados do Cliente</h2>
68                      </div>
69                      <div class="modal-body">
70                          <form class="modal-form" action=""
method="" id="formAlterar">
71                              <label>ID:</label>
72                              <input type="text" name="id">
<br>
73                              <label>Nome:</label>
74                              <input type="text" name="nome">
<br>
75                              <label>CPF: </label>
76                              <input type="text" name="cpf">
```

```
<br>
77                          </form>
78                      </div>
79                      <div class="modal-footer">
80                          <a href="#!" id="btnCadastrar"
class="modal-link-rodape">GRAVAR</a>
81                          <a href="#!" class="modal-link-
rodape" id="btnCancelar">CANCELAR / FECHAR</a>
82                      </div>
83                  </div>
84              </div> <!-- Fim Modal ALteração -->
85          </div>
86          <script src="js/clientes.js"
type="text/javascript"></script>
87          <script src="js/init.js"
type="text/javascript"></script>
88          <script src="js/materialize.js"></script>
89
90      </body>
91 </html>
```

Index.html

Observamos que no arquivo acima temos a tag `<div class="container">` onde serão exibidos os dados vindos do banco de dados. Dentro dessa tag temos uma tabela já estruturada com cabeçalhos, porém sem os dados.

Em seguida temos a tag `<div class="modal" id="modalInc">` onde criamos uma janela modal que será exibida quando o usuário clicar no link INCLUSÃO.

No final do arquivo são inseridos os arquivos contendo o Javascript necessário para esse site.

Listagem dos dados

O servidor appSQL.js em Node.js retornará os dados no formato JSON. O retorno será algo do tipo:

```
[{"ID":1,"Nome":"Luiz","CPF":"11111111111"},
{"ID":2,"Nome":"Maria","CPF":"22222222   "},
{"ID":3,"Nome":"José","CPF":"3333333333 "},
{"ID":4,"Nome":"João","CPF":"44444444444"}]
```

Retorno dos dados do banco em JSON

O arquivo clientes.js recebe esses dados em JSON e exibe no navegador.

```
//////////////////////////////////////
listarClientes = function(tbody) {
```

```
    $("tbody").empty();

    $.ajax({
        url : "http://localhost:3000/clientes"

    }).done(function(dados){

        $.each(dados, function(key, val) {

            //variável tr para dar um append de linha na tabela
            tr = $("<tr />");

            // --- criar botao ALTERAR que vai ser um link
            var btnAlterar = $("<a />").attr({
                    title: "Alterar registro",
                    href:  "#modalAlteracao"
            });

            icone = $("<img />").attr({ class: "botao",
                                        title: "Alterar cliente",
                                        src: "imagens/edit.png" });
            btnAlterar.append(icone);
            // fim do botao alterar

            // --- criar botao EXCLUIR que vai ser um link
            var btnExcluir = $("<a />").attr({
                                        title: "Excluir registro",
                                        href:  "#" });

            icone = $("<img />").attr({ class: "botao",
                                        title: "Excluir cliente",
                                        src: "imagens/delete.png" });
            btnExcluir.append(icone);
            btnExcluir.click(function(){
                excluir(val.ID);
            });
            // fim do botao excluir

            tdBotoes = $("<td />");

            tdBotoes.append(btnAlterar);
            tdBotoes.append(btnExcluir);

            tr.append(tdBotoes);

            //colunas da tabela
            tr.append($("<td />").text(val.ID));
            tr.append($("<td />").text(val.Nome));
            tr.append($("<td />").text(val.CPF));

            //adiciona a linha acima montada na tabela
            $("tbody").append(tr);
        });
    }); //ajax

};
```

Javascript para ler JSON e exibir dados na página

Exclusão

Para implementar a exclusão, basta colocar funcionalidade ao botão que é acrescentado na linha com os dados da tabela (inserido pela função listarClientes().

No arquivo init.js, vamos colocar as funcionalidades de acordo com os eventos da página. Quando o usuário clicar no botão da exclusão passar o ID do cliente para ser excluído.

```
////////////////////////////////////
excluir = function(id){
        if(confirm("Tem certeza que deseja excluir?")){
                $.ajax({
                    url: "http://localhost:3000/clientes/"+id,
                    type: 'DELETE'
                }).done(function(){
                    //chamar listarItem
                    listarClientes("tbody");

                }); //done
        }
};
```

Implementação da função exclusão no arquivo alunos.js

Inclusão

A inclusão será realizada através de um formulário numa janela modal já inserida no arquivo index.html

No arquivo init.js vamos acrescentar as funcionalidades da janela modal e da exclusão:

```
// Get the modal
var modal = document.getElementById('modalInc');

// Get the button that opens the modal
var btnInclusao = document.getElementById("btnInclusao");

// Get the <span> element that closes the modal
var span = document.getElementsByClassName("close")[0];

// Get the link CANCELAR
var btnCancelar = document.getElementById("btnCancelar");

var btnCadastrar = document.getElementById("btnCadastrar");

// Aponta para tabela do index.php
var tblItens = $("#tblListar");

// When the user clicks on the button, open the modal
btnInclusao.onclick = function() {
    modal.style.display = "block";
};

// When the user clicks on <span> (x), close the modal
span.onclick = function() {
    modal.style.display = "none";
};
```

```
// When the user clicks on <span> (x), close the modal
btnCancelar.onclick = function() {
    modal.style.display = "none";
};
// When the user clicks anywhere outside of the modal, close it
window.onclick = function(event) {
    if (event.target == modal) {
        modal.style.display = "none";
    }
};

// When the user clicks on the button, open the modal
btnCadastrar.onclick = function() {
    cadastrar($("#formCadastro"));
};

listarClientes("tbody");
```

Arquivo init.js

No arquivo clientes. js vamos acrescentar a funcionalidade da inclusão:

```
////////////////////////////////////
cadastrar = function(form){
    $.post( "http://localhost:3000/clientes/", form.serialize() ).done(function(data){
        if (!data.erro) {
            form.each(function(data){
                    //limpar formulário
                    this.reset();
            });

            $("#modalInc").closeModal();
        }
        alert(data.mensagem);

        //chamar listarItem
        listarClientes("tbody");
    });
};
```

Funcionalidade da inclusão (função cadastrar()) no arquivo clientes.js

Conclusão

Com esse exemplo exploramos alguns recursos tanto da programação *server side* como *cliente side*. Usamos o Node.js para acesso ao BD no *back-end* e javascript para *front-end*.

Os exemplos acima são extremamente simples. Vale observar que é necessária uma série de verificações e validações de dados, além de tratamento de erros em diversas instâncias.

Fonte

http://www.devmedia.com.br/crud-com-php-mysql-e-ajax-usando-jquery/32197

https://www.w3schools.com/js/js_json_php.asp

Modal: www.w3schools.com/howto/howto_css_modals.asp

Javascript Patterns O´Reilly