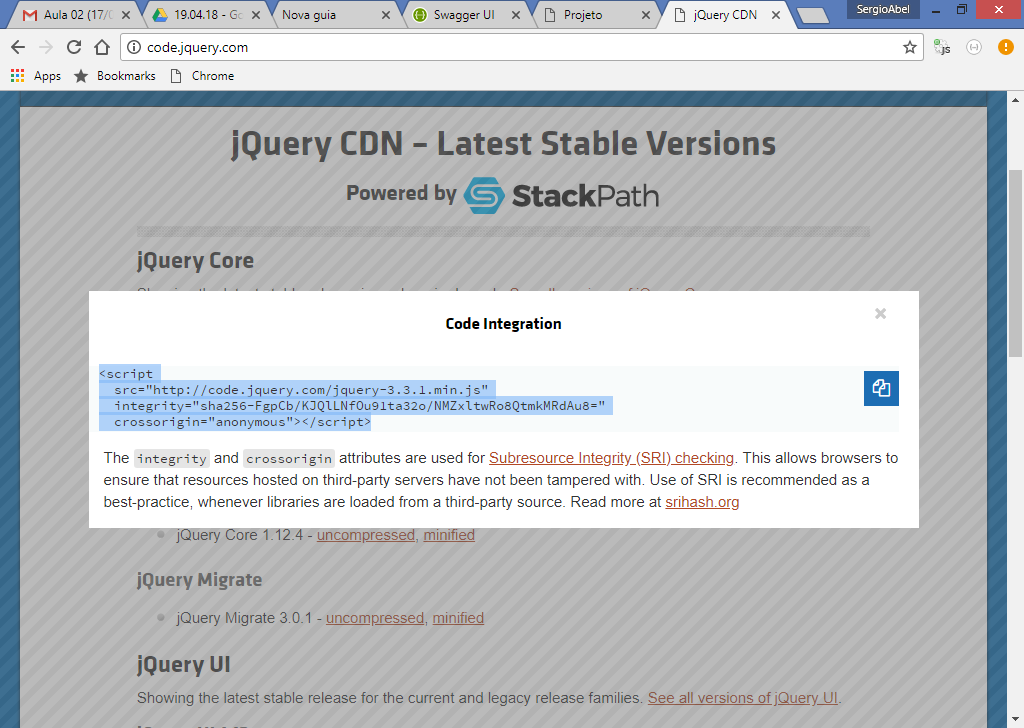
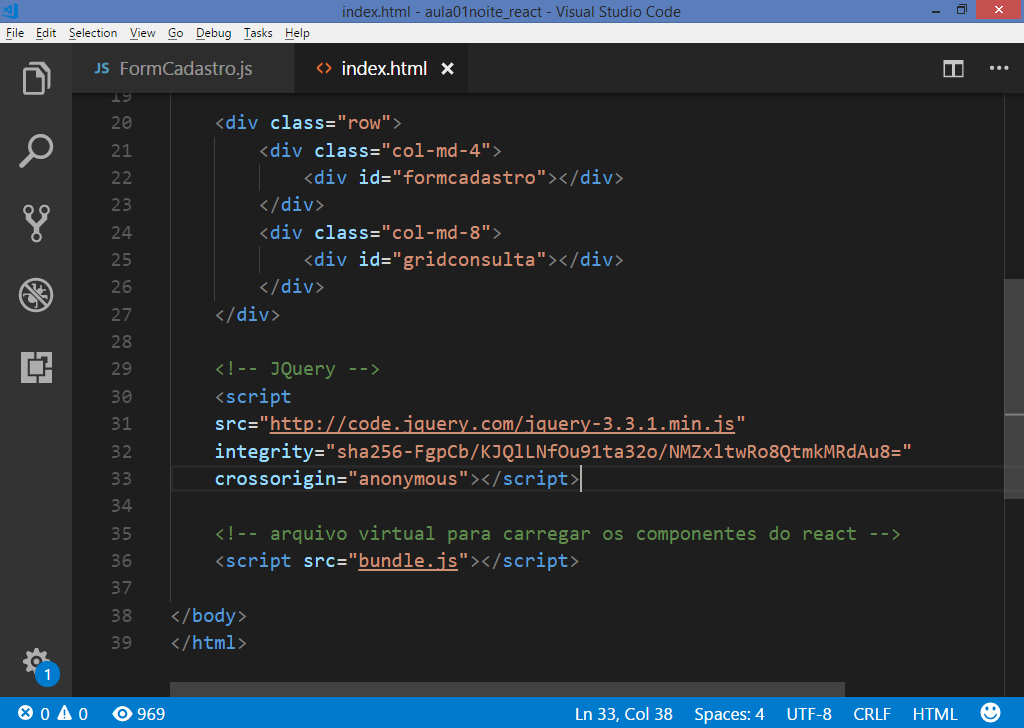
**Continuando...**

http://code.jquery.com/





using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.ComponentModel.DataAnnotations;

namespace Projeto.Services.Models

{

public class ProdutoCadastroViewModel

{

[MinLength(6, ErrorMessage = "Informe no mínimo {1} caracteres.")]

[MaxLength(50, ErrorMessage = "Informe no máximo {1} caracteres.")]

[Required(ErrorMessage = "Por favor, informe o nome do produto.")]

public string Nome { get; set; }

[Range(1, int.MaxValue,

ErrorMessage = "Informe valores entre {1} e {2}.")]

[Required(ErrorMessage = "Por favor, informe o preço do produto.")]

public decimal Preco { get; set; }

[Range(1, int.MaxValue,

ErrorMessage = "Informe valores entre {1} e {2}.")]

[Required(ErrorMessage = "Por favor, informe a quantidade do produto.")]

public int Quantidade { get; set; }

}

}

Implementando o cadastro dos produtos:

//importar as dependencias do React

var React = require('react');

var CreateReactClass = require('create-react-class');

//declarando a classe..

//nome da classe deverá sempre começar CAIXA-ALTA

var FormCadastro = CreateReactClass({

//getInitialState -> declarar os dados que serão

//armazenados pela classe React (atributos)

getInitialState: function () {

return {

nome: '',

preco: 0,

quantidade: 0,

mensagem: '',

erroNome: '',

erroPreco: '',

erroQuantidade: ''

}

},

//função para ler o valor inputado no campo 'Nome'

handleNome: function (e) { //executada em um evento onChange

//e -> variavel que contem o elemento que executou a função..

this.setState({ nome: e.target.value });

},

handlePreco: function (e) {

this.setState({ preco: e.target.value });

},

handleQuantidade: function (e) {

this.setState({ quantidade: e.target.value });

},

//função executada no evento onSubmit..

cadastrarProduto : function(e){ //e -> tag do evento..

e.preventDefault(); //cancelando a ação SUBMIT..

//definir mensagem..

this.setState({

mensagem : "Processando, aguarde...",

erroNome : "",

erroPreco : "",

erroQuantidade : ""

});

//criando um objeto JSON com os dados

//que serão enviados para a requisição..

var model = {

Nome : this.state.nome,

Preco : this.state.preco,

Quantidade : this.state.quantidade

};

//requisição AJAX com JQuery..

$.ajax({

type : "POST",

url : "http://localhost:51210/api/produto/cadastrar",

data : model,

success : function(d){

this.setState({

mensagem : d,

nome : '',

preco : 0,

quantidade : 0

});

}.bind(this),

error: function(e){

if(e.status == 400){ //BadRequest..

this.setState({

mensagem : "",

erroNome : e.responseJSON['model.Nome'],

erroPreco : e.responseJSON['model.Preco'],

erroQuantidade : e.responseJSON['model.Quantidade']

});

}

else if(e.status == 500){ //Internal Server Error..

this.setState({ mensagem : e.responseText });

}

}.bind(this)

});

},

//render -> função do React utilizado para retornar

//o conteúdo HTML que será exibido por esta classe

//na página web..

render: function () {

return (

<div>

<form method="post" onSubmit={this.cadastrarProduto}>

<label>Nome do Produto</label>

<input type="text" className="form-control"

placeholder="Informe o nome"

onChange={this.handleNome}

value={this.state.nome} />

<span className="text-danger">

{this.state.erroNome}

</span>

<br />

<label>Preço:</label>

<input type="text" className="form-control"

placeholder="Informe o preço"

onChange={this.handlePreco}

value={this.state.preco} />

<span className="text-danger">

{this.state.erroPreco}

</span>

<br />

<label>Quantidade:</label>

<input type="text" className="form-control"

placeholder="Informe a quantidade"

onChange={this.handleQuantidade}

value={this.state.quantidade} />

<span className="text-danger">

{this.state.erroQuantidade}

</span>

<br />

<input type="submit" value="Cadastrar Produto"

className="btn btn-success" />

<br />

<br />

<div>

{this.state.mensagem}

</div>

</form>

</div>

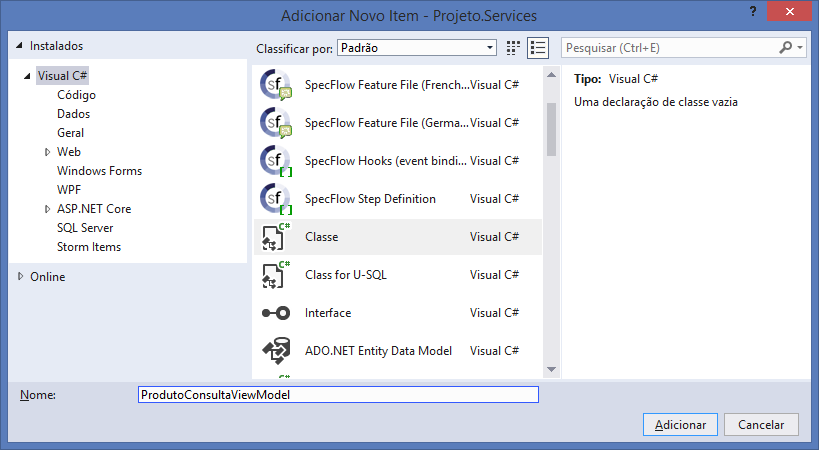
);

}

});

module.exports = FormCadastro;

**Implementando a consulta de Produtos:**



using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

namespace Projeto.Services.Models

{

public class ProdutoConsultaViewModel

{

public int IdProduto { get; set; }

public string Nome { get; set; }

public decimal Preco { get; set; }

public int Quantidade { get; set; }

public decimal Total { get; set; }

}

}

**Implementando os serviços:**

using Projeto.Entities;

using Projeto.Infra.Data.Contracts;

using Projeto.Services.Models;

using System;

using System.Collections;

using System.Collections.Generic;

using System.Linq;

using System.Net;

using System.Net.Http;

using System.Web.Http;

namespace Projeto.Services.Controllers

{

[RoutePrefix("api/produto")]

public class ProdutoController : ApiController

{

//atributo..

private readonly IProdutoRepository repository;

//construtor com entrada de argumentos

//utilizado pelo framework de injeção de dependência..

public ProdutoController(IProdutoRepository repository)

{

this.repository = repository;

}

[HttpPost] //Requisição HTTP POST

[Route("cadastrar")] //URL: /api/produto/cadastrar

public HttpResponseMessage Post(ProdutoCadastroViewModel model)

{

if(ModelState.IsValid) //passou nas validações?

{

try

{

Produto p = new Produto

{

Nome = model.Nome,

Preco = model.Preco,

Quantidade = model.Quantidade

};

repository.Insert(p); //gravando..

return Request.CreateResponse //HTTP 200 (OK)

(HttpStatusCode.OK, $"Produto {p.Nome},

cadastrado com sucesso.");

}

catch(Exception e)

{

return Request.CreateResponse

//HTTP 500 (Internal Server Error)

(HttpStatusCode.InternalServerError, e.Message);

}

}

else

{

Hashtable mapa = new Hashtable();

//varrer o ModelState..

foreach(var s in ModelState)

{

//verificar se existe erro..

if(s.Value.Errors.Count > 0)

{

mapa[s.Key] = s.Value.Errors

.Select(e => e.ErrorMessage).ToList();

}

}

return Request.CreateResponse(HttpStatusCode.BadRequest, mapa);

}

}

[HttpGet] //Requisição HTTP GET

[Route("consultar")] //URL: /api/produto/consultar

public HttpResponseMessage GetAll()

{

try

{

var lista = new List<ProdutoConsultaViewModel>();

foreach(var p in repository.FindAll())

{

var model = new ProdutoConsultaViewModel();

model.IdProduto = p.IdProduto;

model.Nome = p.Nome;

model.Preco = p.Preco;

model.Quantidade = p.Quantidade;

model.Total = p.Preco \* p.Quantidade;

lista.Add(model);

}

//retornando a lista.. HTTP 200

return Request.CreateResponse(HttpStatusCode.OK, lista);

}

catch(Exception e)

{

//retornar erro interno de servidor..

return Request.CreateResponse //HTTP 500..

(HttpStatusCode.InternalServerError, e.Message);

}

}

[HttpGet] //Requisição HTTP GET

[Route("consultarpornome")] //URL: /api/produto/consultarpornome?nome={0}

public HttpResponseMessage GetAllByNome(string nome)

{

try

{

var lista = new List<ProdutoConsultaViewModel>();

foreach (var p in repository.FindByNome(nome))

{

var model = new ProdutoConsultaViewModel();

model.IdProduto = p.IdProduto;

model.Nome = p.Nome;

model.Preco = p.Preco;

model.Quantidade = p.Quantidade;

model.Total = p.Preco \* p.Quantidade;

lista.Add(model);

}

//retornando a lista.. HTTP 200

return Request.CreateResponse(HttpStatusCode.OK, lista);

}

catch (Exception e)

{

//retornar erro interno de servidor..

return Request.CreateResponse //HTTP 500..

(HttpStatusCode.InternalServerError, e.Message);

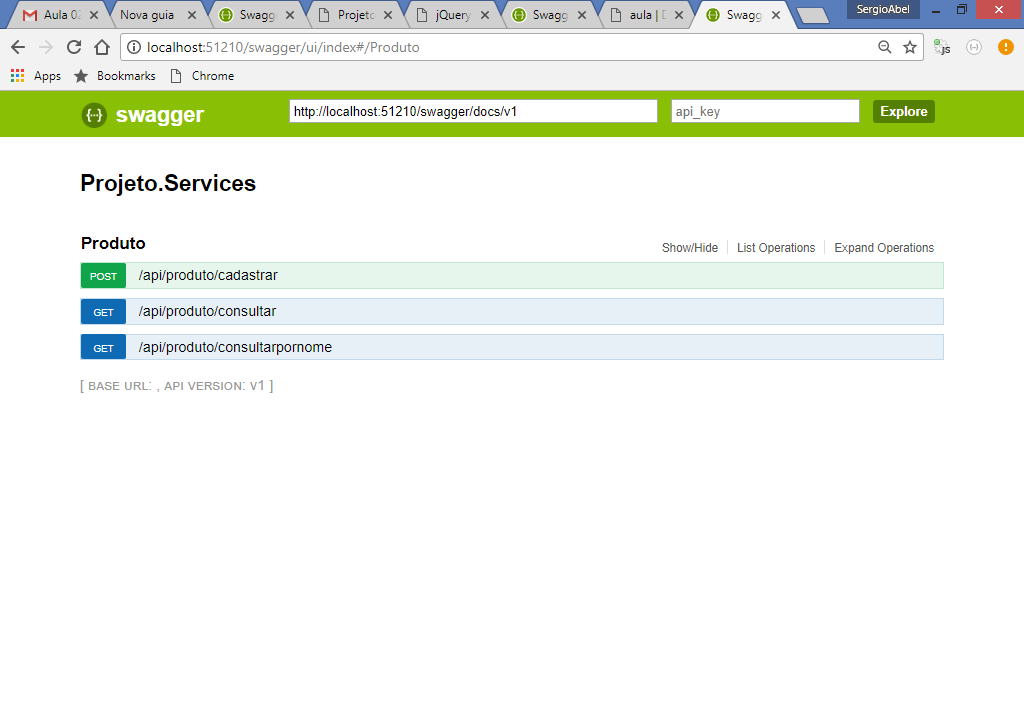
}

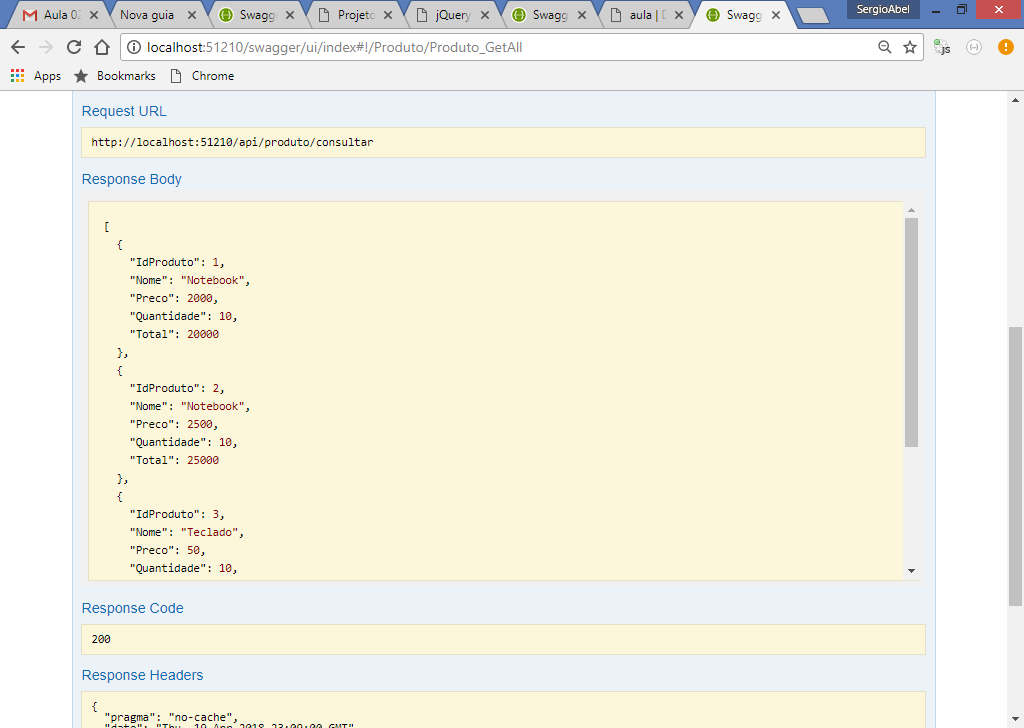
}

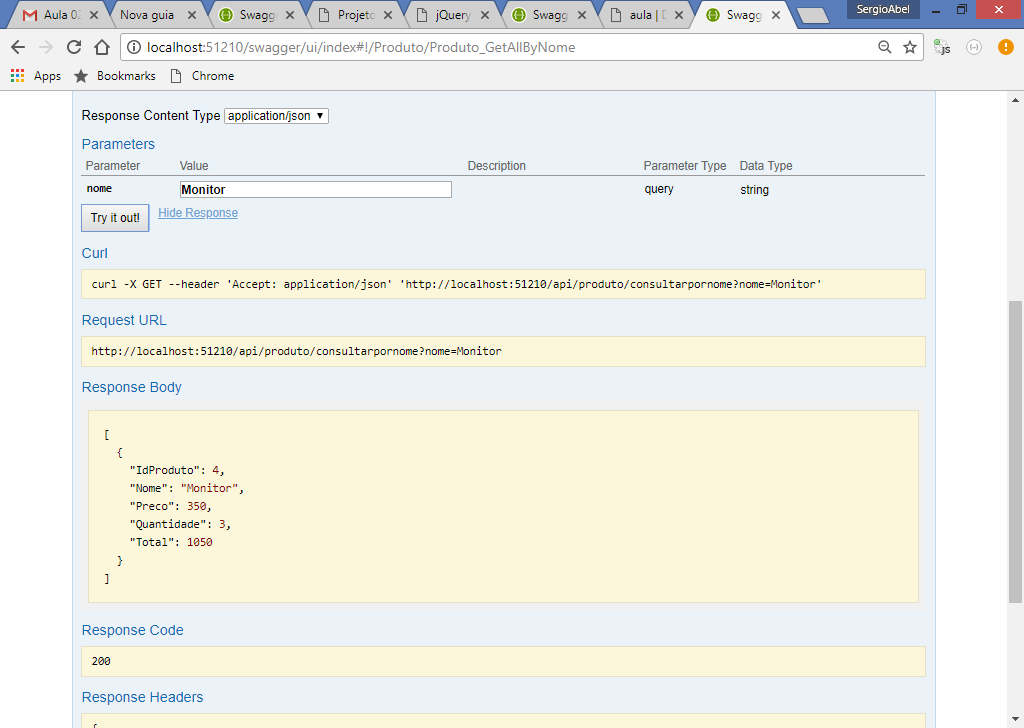
}

}

**Executando:**







**Implementando a consulta de Produtos:**

//importar as dependencias do React

var React = require('react');

var CreateReactClass = require('create-react-class');

//declarando a classe..

//nome da classe deverá sempre começar CAIXA-ALTA

var GridConsulta = CreateReactClass({

//função para declarar os atributos da classe..

getInitialState : function(){

return {

produtos : [], //array..

qtdProdutos : 0,

mensagem : ''

}

},

//função para realizar a requisição de consulta..

consultarProdutos : function(){

//requisição AJAX..

$.ajax({

type: 'GET',

url: 'http://localhost:51210/api/produto/consultar',

data: {},

success: function(d){

this.setState({

mensagem : '',

produtos : d,

qtdProdutos : d.length

});

}.bind(this),

error: function(e){

this.setState({ mensagem : e.responseText });

}.bind(this)

});

},

//função executada sempre que a classe for renderizada..

componentDidMount : function(){

//executar a função de consulta de produtos..

this.consultarProdutos();

},

//render -> função do React utilizado para retornar

//o conteúdo HTML que será exibido por esta classe

//na página web..

render : function(){

return(

<div>

<table className="table table-hover">

<thead>

<tr>

<th>Código</th>

<th>Nome do Produto</th>

<th>Preço</th>

<th>Quantidade</th>

<th>Total</th>

</tr>

</thead>

<tbody>

{

this.state.produtos.map(

function(value, index){

return(

<tr key={index}>

<td>{value.IdProduto}</td>

<td>{value.Nome}</td>

<td>{value.Preco}</td>

<td>{value.Quantidade}</td>

<td>{value.Total}</td>

</tr>

)

}

)

}

</tbody>

<tfoot>

<tr>

<td colSpan="5">

Quantidade de registros: {this.state.qtdProdutos}

</td>

</tr>

</tfoot>

</table>

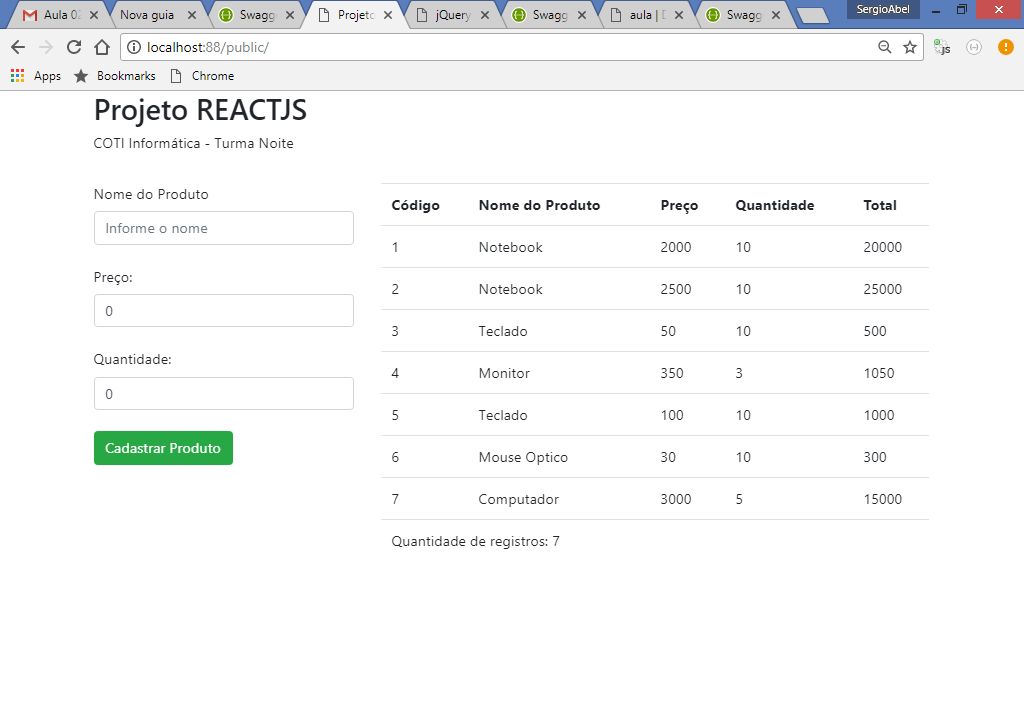
</div>

);

}

});

module.exports = GridConsulta;



------------------------------

**Renderizando a consulta ao cadastrar um novo produto:**

//importar as dependencias do React

var React = require('react');

var ReactDOM = require('react-dom');

var CreateReactClass = require('create-react-class');

//importando a classe que monta a consulta de produtos..

var GridConsulta = require('./GridConsulta');

//criando uma variavel para executar a renderização

//da consulta de produtos..

var Grid = ReactDOM.render(

<GridConsulta/>,

document.getElementById('gridconsulta')

);

//declarando a classe..

//nome da classe deverá sempre começar CAIXA-ALTA

var FormCadastro = CreateReactClass({

//getInitialState -> declarar os dados que serão

//armazenados pela classe React (atributos)

getInitialState: function () {

return {

nome: '',

preco: 0,

quantidade: 0,

mensagem: '',

erroNome: '',

erroPreco: '',

erroQuantidade: ''

}

},

//função para ler o valor inputado no campo 'Nome'

handleNome: function (e) { //executada em um evento onChange

//e -> variavel que contem o elemento que executou a função..

this.setState({ nome: e.target.value });

},

handlePreco: function (e) {

this.setState({ preco: e.target.value });

},

handleQuantidade: function (e) {

this.setState({ quantidade: e.target.value });

},

//função executada no evento onSubmit..

cadastrarProduto : function(e){ //e -> tag do evento..

e.preventDefault(); //cancelando a ação SUBMIT..

//definir mensagem..

this.setState({

mensagem : "Processando, aguarde...",

erroNome : "",

erroPreco : "",

erroQuantidade : ""

});

//criando um objeto JSON com os dados

//que serão enviados para a requisição..

var model = {

Nome : this.state.nome,

Preco : this.state.preco,

Quantidade : this.state.quantidade

};

//requisição AJAX com JQuery..

$.ajax({

type : "POST",

url : "http://localhost:51210/api/produto/cadastrar",

data : model,

success : function(d){

this.setState({

mensagem : d,

nome : '',

preco : 0,

quantidade : 0

});

//executando e renderizando a consulta..

Grid.consultarProdutos();

}.bind(this),

error: function(e){

if(e.status == 400){ //BadRequest..

this.setState({

mensagem : "",

erroNome : e.responseJSON['model.Nome'],

erroPreco : e.responseJSON['model.Preco'],

erroQuantidade : e.responseJSON['model.Quantidade']

});

}

else if(e.status == 500){ //Internal Server Error..

this.setState({ mensagem : e.responseText });

}

}.bind(this)

});

},

//render -> função do React utilizado para retornar

//o conteúdo HTML que será exibido por esta classe

//na página web..

render: function () {

return (

<div>

<form method="post" onSubmit={this.cadastrarProduto}>

<label>Nome do Produto</label>

<input type="text" className="form-control"

placeholder="Informe o nome"

onChange={this.handleNome}

value={this.state.nome} />

<span className="text-danger">

{this.state.erroNome}

</span>

<br />

<label>Preço:</label>

<input type="text" className="form-control"

placeholder="Informe o preço"

onChange={this.handlePreco}

value={this.state.preco} />

<span className="text-danger">

{this.state.erroPreco}

</span>

<br />

<label>Quantidade:</label>

<input type="text" className="form-control"

placeholder="Informe a quantidade"

onChange={this.handleQuantidade}

value={this.state.quantidade} />

<span className="text-danger">

{this.state.erroQuantidade}

</span>

<br />

<input type="submit" value="Cadastrar Produto"

className="btn btn-success" />

<br />

<br />

<div>

{this.state.mensagem}

</div>

</form>

</div>

);

}

});

module.exports = FormCadastro;

