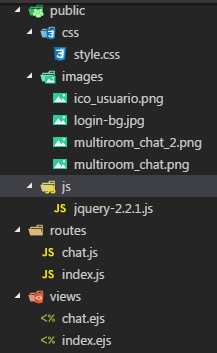
- Baixar o designer da aplicação do chat

- colocar os arquivos em seus devidos lugares dentro da pasta app > public > [css, js, images]

- na pasta de app > views vai o index.ejs e chat.ejs



- para testar as views com vou mudar a rota index apontando para o index.ejs da view, e criar um nova rota chamada chat

/\* app [application] é passada pelo autoload do consign \*/

module.exports = function (application) {

application.get('/', function (req, res) {

res.render('index.ejs');

});

}

module.exports = function (application) {

application.post('/chat', function (req, res) {

res.render('chat');

});

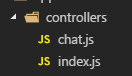
application.get('/chat', function (req, res) {

res.render('chat');

});

}

**- Configurar as rotas para que utilizem os controllers (onde vai ficar a logica)**



App > routes > index

module.exports = function (application) {

application.get('/', function (req, res) {

application.app.controllers.index.home(application, req, res);

});

}

App > controller > index

module.exports.home = function (application, req, res) {

res.render('index.ejs');

}

App > routes > chat

module.exports = function (application) {

application.post('/chat', function (req, res) {

application.app.controllers.chat.iniciaChat(application, req, res);

});

application.get('/chat', function (req, res) {

application.app.controllers.chat.iniciaChat(application, req, res);

});

}

App > controller > chat

module.exports.iniciaChat = function (application, req, res) {

res.render('chat');

}

**Seguindo…**

<form> do index.ejs – colocando action=”/chat” e o melhod=”POST”

<form accept-charset="UTF-8" role="form" class="form-signin" action="/chat" method="POST">

Lembrando que para recuperar a informação passada via fomulario, configurei como middleware o bodyParser, lá no config > server.js

- fazendo a validação do apelido inserido no formulário

module.exports.iniciaChat = function (application, req, res) {

/\* vai ter um json com os campos e valores enviados do formulario atraves do 'name' \*/

var dadosForm = req.body;

/\* funcao assert vem do expressValidator (esta no app -> application) \*/

req.assert('apelido', 'Nome ou apelido é obrigatório').notEmpty();

req.assert('apelido', 'Nome ou apelido deve ter entre 3 até 15 caracteres').len(3, 5);

var erros = req.validationErrors();

if (erros) {

res.send('Existem erros no formulario');

return;

}

res.render('chat');

}

- renderizando a mesma tela com os erros

if (erros) {

res.render('index', { validacao: erros });

return;

}

- Recuperando os erros na view

<!-- recuperando os erros -->

<div>

<% if(validacao.length > 0) { %>

<div class="row">

<div class="col-md-12">

<div class="alert alert-danger">

<strong>Atenção!</strong>

<ul>

<% for(var i = 0; i < validacao.length; i++) { %>

<li><%= validacao[i].msg %></li>

<% } %>

</ul>

</div>

</div>

</div>

<% } %>

</div>

- No controller index, que vai abrir simplesmente o index, eu tenho que colocar a validação associado com o obj vazio

module.exports.home = function (application, req, res) {

res.render('index', { validacao: {} });

}

