Ournote

Rede Social

Artur Queiroz Luis Albuquerque

Resumo

Trabalho prático de Desemvolvimento Web

Conteúdo

Introdução

Já existem muitas redes sociais, umas mais voltadas para partilhar fotos, outras para partilhar o estado das pessoas, e outras até mais focadas no mundo de trabalho, no entanto não existem assim tantas focadas em desemvolvimento e partilha de notas entre alunos. Com isso em conta, desemvolvemos Ournote, que é uma rede social focada em alunos, organizada em grupos e em páginas, onde o contéudo pode ser organizado como num caderno, mas com a vantagem de aqui ser permitido facilmente trocar a ordem dos textos, imagens, ficheiros, etc. Os grupos são constituidos por páginas, e uma página esta organizada por uma meta linguagem, desemvolvida por nós, que permite distingir títulos de difrentes importâncias, criar paragrafos, listas de paragrafos. imagens, eventos, pdfs, e até ficheiros distintos para fazer download.

Modelo

No nosso modelo os nossos modelos foram distingidos em duas classes, users e groups.

Users

```
name: String,
email : {
         type: String,
         unique: true,
         lowercase: true,
         required: true
       },
password: {
        type: String,
        required: true
        },
favourite: [ String ],
```

Os *favourites* são um conjunto de caminhos de grupos

Groups

Cada evento tem um título e uma data.

```
var eventSchema = new mongoose.Schema({
    title : String,
    data  : String
});
```

Uma *card* é um objeto de uma página pode ser um parágrafo, um título uma imagem, etc.

```
var card_Schema = new mongoose.Schema({
    p: String,
    img: String,
    pdf: String,
    h1: String,
    h2: String,
    h3: String,
    a: String,
    file : String,
    list : [String],
    event : eventSchema,
```

Tudo acima, dentro de *card* são os objectos que uma página pode ter.

```
    comment : [String],
    tags : [String]
```

Lista de comentários e uma lista de *tags*.

```
});

var PATH  = String;
var EMAIL = String;
```

Um *group* é algo que tem uma página com várias *cards* e outros sub-grupos, com *tags* e umas permissões, tanto de leitura como de escrita.

```
var groupSchema = new mongoose.Schema({
    path : {
            type :String,
            unique: true,
            required: true
        },
    id_creator: EMAIL,
    name: {
             type: String,
             required: true
           },
    tags : [String],
    sub_groups : [ PATH ],
    read_perm  : [ EMAIL ],
    write_perm : [ EMAIL ],
    page       : [ card_Schema ],
  });
```

Esquema

Nós dividimos o trabalho em dois servidores (API e Interface) e clientes.

API

GET

```
/*?token=TOKEN                              --- dados do grupo
/profile?token=TOKEN&email=EMAIL           --- favoritos de um utilizador
/profile?token=TOKEN&email=EMAIL&tag=TAG --- Sitios que têm associado uma determinada tag
/user/:email?token=TOKEN                   --- dados de um utilizador
```

POST

```
/*?token=TOKEN&update=comment          --- adiciona um comentario a um objeto
/*?token=TOKEN&update=add              --- adiciona premissões a um ou mais utilizadores
/*?token=TOKEN&update=remove           --- remove premissões a um ou mais utilizadores
/*?token=TOKEN&type=file               --- upload de um ficheiro
/*?token=TOKEN                         --- adiciona um novo grupo

/login?token=TOKEN&email=EMAIL                 --- faz login
/favourite?token=TOKEN&email=EMAIL&path=PATH   --- adiciona uma página aos favoritos
/register                                      --- regista um utilizador
```

PUT

```
/*?token=TOKEN&type=TYPE               --- acrescenta um objeto à página
```

DELETE

```
/*?token=TOKEN                         --- remove um objeto da página
```

Interface

GET

```
/                                      --- responde com a página de login
/register                              --- responde com a página para registar
/profile?tag=TAG                       --- procura objectos que tenham a tag TAG
/logout                                --- termina a sessão de um utilizador
/*                                     --- página de um grupo
/*?json=true                           --- grupo em json
```

POST

```
/add_favourite?path=PATH               --- adiciona aos favoritos
/register                              --- regista um utilizador
/login                                 --- faz login
root/*?update=add                      --- acrescenta premissões
root/*?update=remove                   --- remove premissões
root/*?update=comment                  --- acrescenta um comentario
```

`root/*?type=TYPE` `--- acrescenta um ficheiro`

PUT

`root/*` `--- acrescenta um objeto`

DELETE

`root/*` `--- elimina um ou mais objetos`

Cliente

Conclusão

Ao longo desta jornada, encontramos diversos desafios, quer na discucão e esquematização da solução, como na própria implementação, como decidir qual seria a nossa abordgem e ao facto de termos usado javascript, e isso nos trazer muitos problemas na hora de debugging, uma vez que os erros muitas vezes ficam perdidos entre a API e a Interface, mas tendo o postman podemos testar individualmente a API, e com isso identificar mais facilmente a razão do problema. No fim de contas, conseguimos apresentar uma rede social funcional, com uma proposta bem original, que acreditamos que consiga ser utíl para estudantes organizarem seus trabalhos, e a sua vida escolar.

Trabalho futuro

No nosso trabalho, usamos uma meta linguagem para representar as páginas dos grupos, uma boa adição, seria adicionarmos elementos recursivas e outras funcionalidades, como canvas e suporte de videos, etc. Podiamos criar um parser, e com ajuda do pandoc convertermos as páginas em pdf, powerpoint, word, epub, etc. Infelizmente, não conseguimos fazer autênticação de ficheiros, apesar de termos tentado de diversas formas, acrescentar isso, seria fulcral para um bom funcionamento da aplicação. Outra coisa, menos importante, mas que seria mais conveniente para o utilizador, era ocultarmos as páginas que o utilizador não tem premissão de leitura. Neste momento, é possivel acrescentar favoritos, mas não elimina-los, isso seria uma boa adição.