

第15章 增加转载功能和转载统计

Edit Page

Page History

Clone URL

现在，我们来给博客添加转载文章的功能和转载数统计。

我们的设计思路是：当在线用户满足特定条件时，文章页面才会显示 `转载` 链接字样，当用户点击 `转载` 后，复制一份存储当前文章的文档，修改后以新文档的形式存入数据库，而不是单纯的添加一条指向被转载的文档的"引用"，这种设计是合理的，因为这样我们也可以将转载来的文章进行修改，类似于 `GitHub` 上的 `Fork`。

首先，我们来完成转载文章的功能。

打开 `post.js`，将 `Post.prototype.save` 内的

```
var post = {
  name: this.name,
  head: this.head,
  time: time,
  title:this.title,
  tags: this.tags,
  post: this.post,
  comments: [],
  pv: 0
};
```

修改为

```
var post = {
  name: this.name,
  head: this.head,
  time: time,
  title:this.title,
  tags: this.tags,
  post: this.post,
  comments: [],
  reprint_info: {},
  pv: 0
};
```

这里我们给文档里添加了 `reprint_info` 键，最多为以下形式：

```
{
  reprint_from: {name: xxx, day: xxx, title: xxx},
  reprint_to: [
    {name: xxx, day: xxx, title: xxx},
    {name: xxx, day: xxx, title: xxx},
    ...
  ]
}
```

`reprint_from` 表示转载来的原文章的信息，`reprint_to` 表示该文章被转载的信息。为了节约存储空间，我们初始设置 `reprint_info` 键为 `{}`，而不是以下形式：

```
{
  reprint_from: {},
  reprint_to: []
}
```

这是因为大多数文章是没有经过任何转载的，所以为每个文档都添加以上形式的 `reprint_info` 是有点浪费的。假如某篇文章是转载来的，我们只需给 `reprint_info` 添加上 `reprint_from` 键即可，假如某篇文章被转载了，我们只需给 `reprint_info` 添加上 `reprint_to` 键即可，假如某篇文章是转载来的且又被转载了，那我们就给 `reprint_info` 添加上 `reprint_from` 和 `reprint_to` 键即可。

打开 `post.js`，在最后添加如下代码：

```
//转载一篇文章
Post.reprint = function(reprint_from, reprint_to, callback) {
  mongodb.open(function (err, db) {
    if (err) {
      return callback(err);
    }
    db.collection('posts', function (err, collection) {
      if (err) {
        mongodb.close();
        return callback(err);
      }
      //找到被转载的原文档
      collection.findOne({
        "name": reprint_from.name,
        "time.day": reprint_from.day,
        "title": reprint_from.title
      }, function (err, doc) {
        if (err) {
          mongodb.close();
          return callback(err);
        }

        var date = new Date();
        var time = {
          date: date,
          year : date.getFullYear(),
          month : date.getFullYear() + "-" + (date.getMonth()+1),
          day : date.getFullYear() + "-" + (date.getMonth()+1) + "-" + date.getDate(),
          minute : date.getFullYear() + "-" + (date.getMonth()+1) + "-" + date.getDate() + " " + date.getHours() + ":" + date.ge

        }

        delete doc._id;

        doc.name = reprint_to.name;
        doc.head = reprint_to.head;
        doc.time = time;
        doc.title = (doc.title.search(/[[转载]]/) > -1) ? doc.title : "[转载]" + doc.title;
        doc.comments = [];
        doc.reprint_info = {"reprint_from": reprint_from};
        doc.pv = 0;

        //更新被转载的原文档的 reprint_info 内的 reprint_to
        collection.update({
          "name": reprint_from.name,
          "time.day": reprint_from.day,
          "title": reprint_from.title
        }, {
          $push: {
            "reprint_info.reprint_to": {
              "name": reprint_to.name,
              "day": time.day,
              "title": doc.title
            }
          }
        }, function (err, result) {
          if (err) {
            mongodb.close();
            return callback(err);
          }
        });

        //将转载生成的副本修改后存入数据库，并返回存储后的文档
        collection.insert(doc, {
          safe: true
        }, function (err, post) {
          mongodb.close();
          if (err) {
            return callback(err);
          }
          callback(err, post[0]);
        });
      });
    });
  });
};
```

```
};
```

这里我们在 `Post.reprint()` 内实现了被转载的原文的更新和转载后文章的存储。

打开 `index.js`，在 `app.get('/remove/:name/:day/:title')` 后添加如下代码：

```
app.get('/reprint/:name/:day/:title', checkLogin);
app.get('/reprint/:name/:day/:title', function (req, res) {
  Post.edit(req.params.name, req.params.day, req.params.title, function (err, post) {
    if (err) {
      req.flash('error', err);
      return res.redirect(back);
    }
    var currentUser = req.session.user,
        reprint_from = {name: post.name, day: post.time.day, title: post.title},
        reprint_to = {name: currentUser.name, head: currentUser.head};
    Post.reprint(reprint_from, reprint_to, function (err, post) {
      if (err) {
        req.flash('error', err);
        return res.redirect('back');
      }
      req.flash('success', '转载成功!');
      var url = '/u/' + post.name + '/' + post.time.day + '/' + post.title;
      //跳转到转载后的文章页面
      res.redirect(url);
    });
  });
});
```

这里我们给 `转载` 链接添注册了路由响应。

注意：我们需要通过 `Post.edit()` 返回一篇文章 `markdown` 格式的文本，而不是通过 `Post.getOne` 返回一篇转义后的 `HTML` 文本，因为我们还要将修改后的文档存入数据库，而数据库中应该存储 `markdown` 格式的文本。

最后，我们在文章页（`article.ejs`）添加转载链接。

打开 `article.ejs`，在

```
<% if (user && (user.name == post.name)) { %>
  <span><a class="edit" href="/edit/<%= post.name %>/<%= post.time.day %>/<%= post.title %>">编辑</a></span>
  <span><a class="edit" href="/remove/<%= post.name %>/<%= post.time.day %>/<%= post.title %>">删除</a></span>
<% } %>
```

后添加如下代码

```
<% var flag = 1 %>
<% if (user && (user.name != post.name)) { %>
  <% if ((post.reprint_info.reprint_from != undefined) && (user.name == post.reprint_info.reprint_from.name)) { %>
    <% flag = 0 %>
  <% } %>
  <% if ((post.reprint_info.reprint_to != undefined)) { %>
    <% post.reprint_info.reprint_to.forEach(function (reprint_to, index) { %>
      <% if (user.name == reprint_to.name) { %>
        <% flag = 0 %>
      <% } %>
    <% }) %>
  <% } %>
<% } else { %>
  <% flag = 0 %>
<% } %>
<% if (flag) { %>
  <span><a class="edit" href="/reprint/<%= post.name %>/<%= post.time.day %>/<%= post.title %>">转载</a></span>
<% } %>
```

以上代码的意思是：我们设置一个 `flag` 标志，如果用户是游客，或者不是该文章的目前作者，或者是该文章的上一级作者，或者已经转载过该文章，都会将 `flag` 设置为 `0`，即不能转载该文章。最后判断 `flag` 为 `1` 时才会显示 `转载` 链接，即才可以转载这篇文章。

最后，我们需要添加一个 `原文链接` 来指向被转载的文章。

打开 `index.ejs`、`user.ejs`、`article.ejs`，在第一个 `<p class="info">` 里最后添加如下代码：

```
<% if (post.reprint_info.reprint_from) { %>
  <br><a href="/u/<%= post.reprint_info.reprint_from.name %>/<%= post.reprint_info.reprint_from.day %>/<%= post.reprint_info.reprint_from.title %>"><%= post.reprint_info.reprint_from.name %> <%= post.reprint_info.reprint_from.day %> <%= post.reprint_info.reprint_from.title %> </a>
<% } %>
```

现在我们就给博客添加了转载功能。

接下来我们添加转载统计。

添加转载统计就简单多了，我们只需使用 `reprint_info.reprint_to.length` 即可。

打开 `index.ejs`、`user.ejs`、`article.ejs`，将

```
<p class="info">阅读: <%= post.pv %> | 评论: <%= post.comments.length %></p>
```

修改为

```
<p class="info">
  阅读: <%= post.pv %> |
  评论: <%= post.comments.length %> |
  转载:
  <% if (post.reprint_info.reprint_to) { %>
    <%= post.reprint_info.reprint_to.length %>
  <% } else { %>
    <%= 0 %>
  <% } %>
</p>
```

现在我们就给博客添加了转载统计的功能。但工作还没有完成，假如我们要删除一篇转载来的文章时，还要将被转载的文章所在的文档的 `reprint_to` 删除遗留的转载信息。

打开 `post.js`，将 `Post.remove` 修改为:

```
//删除一篇文章
Post.remove = function(name, day, title, callback) {
  //打开数据库
  mongodb.open(function (err, db) {
    if (err) {
      return callback(err);
    }
    //读取 posts 集合
    db.collection('posts', function (err, collection) {
      if (err) {
        mongodb.close();
        return callback(err);
      }
      //查询要删除的文档
      collection.findOne({
        "name": name,
        "time.day": day,
        "title": title
      }, function (err, doc) {
        if (err) {
          mongodb.close();
          return callback(err);
        }
        //如果有 reprint_from, 即该文章是转载来的, 先保存下来
        var reprint_from = "";
        if (doc.reprint_info.reprint_from) {
          reprint_from = doc.reprint_info.reprint_from;
        }
        if (reprint_from != "") {
          //更新原文章所在文档的 reprint_to
          collection.update({
            "name": reprint_from.name,
            "time.day": reprint_from.day,
            "title": reprint_from.title
          }, {
            $pull: {
              "reprint_info.reprint_to": {
                "name": name,
                "day": day,
                "title": title
              }
            }
          }, {
            multi: true
          }, function (err) {
            if (err) {
              return callback(err);
            }
            //删除原文章
            collection.remove({
              "name": name,
              "time.day": day,
              "title": title
            }, function (err) {
              if (err) {
                return callback(err);
              }
              callback();
            });
          });
        } else {
          callback();
        }
      });
    });
  });
}
```

```
    })
  }, function (err) {
    if (err) {
      mongodb.close();
      return callback(err);
    }
  });
}

//删除转载来的文章所在的文档
collection.remove({
  "name": name,
  "time.day": day,
  "title": title
}, {
  w: 1
}, function (err) {
  mongodb.close();
  if (err) {
    return callback(err);
  }
  callback(null);
});
});
});
});
};
```

注意：我们使用了 `$pull` 来删除数组中的特定项，关于 `$pull` 的使用详见《MongoDB 权威指南》。

至此，我们给博客添加了转载功能和转载统计。

Last edited by nswbmw, 9 days ago

