分页实现

分页实现

- 1. 发布用户帖子
 - 1.1. 新建用户帖子表单类
 - 1.2. 将帖子表单添加到首页模板中
 - 1.3. 视图函数处理表单数据
 - 1.4. 启动服务测试
- 2. 新增发现页面展示所有用户的帖子
 - 2.1. 新增发现视图函数
 - 2.2. 注册发现视图
 - 2.3. 将帖子子模板中作者信息修改为链接
 - 2.4. 启动服务测试
- 3. 用户帖子分页处理
 - 3.1. 给主页及发现页视图函数增加分页处理
 - 3.2. 新增分页导航视图
 - 3.3. 新增分页导航模板
 - 3.4. 启动服务测试
- 4. 个人主页分页处理
 - 4.1. 视图函数增加分页处理
 - 4.2. 模板添加分页导航
 - 4.3. 启动服务测试
- 5. 调整分页时每页数据条数参数

1. 发布用户帖子

1.1. 新建用户帖子表单类

修改 app/forms.py 脚本,新增用户帖子表单类,用于帖子信息的提交。

```
class PostForm(FlaskForm):
    """帖子提交表单"""
    post = TextAreaField('内容', validators=[DataRequired(), Length(min=1, max=140)])
    submit = SubmitField('提交')
```

1.2. 将帖子表单添加到首页模板中

修改 app/templates/index/index.html 文件, 增加帖子提交表单。

```
{% extends 'base.html' %}

{% block content %}
    <h1>Hi, {{ current_user.username }}!</h1>
    <form action="" method="post">
```

1.3. 视图函数处理表单数据

修改 app/index.py 脚本,修改原视图函数处理,增加表单数据的提交及查询展示。其中通过 followed_posts 方法可以获取本人以及其关注者对应的帖子信息。

```
from flask.views import View
from flask import render_template, flash, redirect, url_for
from flask_login import login_required, current_user
from app import db
from app.forms import PostForm
from app.models import Post
class IndexView(View):
   methods = ['GET', 'POST']
   decorators = [login_required]
   def dispatch_request(self):
       form = PostForm()
        if form.validate_on_submit():
            post = Post(body=form.post.data, author=current_user)
            db.session.add(post)
            db.session.commit()
            flash('你的帖子已提交!')
            return redirect(url_for('index'))
       # 获取本人及关注者对应的帖子信息
       posts = current_user.followed_posts().all()
        return render_template('index/index.html', title='ion', form=form, posts=posts)
```

1.4. 启动服务测试

通过john用户登录首页。

← → C ① 127.0.0.1:5000

博客: 首页 个人资料 退出

Hi, john!

内容	

提交

发表一篇帖子信息。

← → C ① 127.0.0.1:5000

博客: 首页 个人资料 退出

• 你的帖子已提交!

Hi, john!

内容	
	//

提交



2. 新增发现页面展示所有用户的帖子

2.1. 新增发现视图函数

修改 app/index.py 脚本,增加发现视图函数。

```
class ExploreView(View):
    """发现视图"""
    methods = ['GET']
    decorators = [login_required]

def dispatch_request(self):
    posts = Post.query.order_by(Post.timestamp.desc()).all()
    return render_template('index/index.html', title='发现', posts=posts)
```

2.2. 注册发现视图

修改 app/__init__.py 脚本,增加发现视图注册。

```
def create_app(test_config=None):
.....

# 注册Index首页视图URL
from app.index import IndexView, ExploreView
application.add_url_rule('/', view_func=IndexView.as_view('index'))
# 注册Explore发现视图URL
application.add_url_rule('/explore', view_func=ExploreView.as_view('explore'))
.....
```

2.3.首页增加是否传入表单判断

修改 app/templates/index/index.html 文件,增加表单是否传入判断,若未传入表单,则不展示帖子表单提交处理。

2.4. base页面增加发现页面链接

修改 app/templates/base.html 文件, 增减发现页面链接。

2.3. 将帖子子模板中作者信息修改为链接

修改 app/templates/common/post.html 文件,将作者用户名修改为链接方式。

2.4. 启动服务测试

通过用户john登录。

← → C ① 127.0.0.1:5000

博客: 首页 发现 个人资料 退出

Hi, john!

内容	

提交



通过用户susan登录,并新增帖子。

博客: 首页 发现 个人资料 退出

Hi, susan!

内容		
		//

提交

 \leftarrow \rightarrow \bigcirc (i) 127.0.0.1:5000

博客: 首页 发现 个人资料 退出

• 你的帖子已提交!

Hi, susan!

内容	

提交



点击发现链接。

← → C (i) 127.0.0.1:5000/explore

博客: 首页发现个人资料退出

Hi, susan!





点击用户john的用户名链接。

← → C ① 127.0.0.1:5000/user/info/john

博客: 首页 发现 个人资料 退出



用户: john

最后访问于:

2019-05-05 08:45:59.377292

粉丝 0, 关注 0

关注





3. 用户帖子分页处理

3.1. 给主页及发现页视图函数增加分页处理

修改 app/index.py 脚本,增加分页处理。其中,查询页数通过参数 page 以查询字符串参数方式指定,该参数可以通过 request.args 对象获取,默认为第一页。

分页处理函数 paginate:

- 第一个参数为当前查询页码数
- 第二个参数为每页查询数据条数 (设置到 config.py 文件中, 测试先设置为2条)
- 第三个参数为错误处理布尔标记,如果是 True ,当请求范围超出已知范围时自动引发404错误;如果是 False ,则会返回一个空列表

```
class IndexView(View):
   methods = ['GET', 'POST']
   decorators = [login_required]
   def dispatch_request(self):
       # 获取本人及关注者对应的帖子信息
       page = request.args.get('page', 1, type=int)
       # 分页处理
       posts = current_user.followed_posts().paginate(page,
current_app.config['POSTS_PER_PAGE'], False)
       return render_template('index/index.html', title='首页', form=form, posts=posts)
class ExploreView(View):
   """发现视图"""
   methods = ['GET']
   decorators = [login_required]
   def dispatch_request(self):
       page = request.args.get('page', 1, type=int)
       # 分页处理
       posts = Post.query.order_by(Post.timestamp.desc()).paginate(page,
current_app.config['POSTS_PER_PAGE'], False)
       return render_template('index/index.html', title='发现', posts=posts)
```

3.2. 新增分页导航视图

paginate 方法返回一个 Pagination 的实例,该实例有几个属性可以用来构建分页链接:

- items: 请求内容的数据列表
- has_next: 当前页之后存在后续页面时为真
- has_prev: 当前页之前存在前置页面时为真
- next_num:下一页的页码
- prev_num:上一页的页码

通过这几个元素,修改 app/index.py 脚本,可以生成上一页和下一页的链接并将其传入模板以渲染。url_for() 函数有一个特点,可以添加任何关键字参数,如果这些参数的名字没有直接在URL中匹配使用,那么Flask将它们设置为URL的查询字符串参数。

```
class IndexView(View):
    methods = ['GET', 'POST']
    decorators = [login_required]

def dispatch_request(self):
```

```
. . . . . .
       # 获取本人及关注者对应的帖子信息
       page = request.args.get('page', 1, type=int)
       # 分页处理
       posts = current_user.followed_posts().paginate(page,
current_app.config['POSTS_PER_PAGE'], False)
       # 下一页
       next_url = url_for('index', page=posts.next_num) if posts.has_next else None
       # 上一页
       prev_url = url_for('index', page=posts.prev_num) if posts.has_prev else None
       return render_template('index/index.html', title='首页', form=form,
posts=posts.items,
                              next_url=next_url, prev_url=prev_url, page=page)
class ExploreView(View):
   """发现视图"""
   methods = ['GET']
   decorators = [login_required]
   def dispatch_request(self):
       page = request.args.get('page', 1, type=int)
       # 分页处理
       posts = Post.query.order_by(Post.timestamp.desc()).paginate(page,
current_app.config['POSTS_PER_PAGE'], False)
       # 下一页
       next_url = url_for('explore', page=posts.next_num) if posts.has_next else None
       # 上一页
       prev_url = url_for('explore', page=posts.prev_num) if posts.has_prev else None
       return render_template('index/index.html', title='发现', posts=posts.items,
                              next_url=next_url, prev_url=prev_url, page=page)
```

3.3. 新增分页导航模板

修改 app/templates/index/index.html 文件, 增加分页导航展示。

3.4. 启动服务测试

通过用户susan登录,该用户有两条帖子,只展示当前页数。

← → C ① 127.0.0.1:5000

博客: 首页发现个人资料退出

Hi, susan!

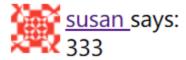
内容	
提交	
susan says:	
susan says: 2222	
第1页	

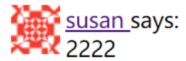
点击发现链接,展示所有用户的帖子信息,总共有三条,则会展示下一页链接。

← → C ① 127.0.0.1:5000/explore

博客: 首页 发现 个人资料 退出

Hi, susan!





第1页 下一页

点击下一页链接,展示另外一条帖子信息,并显示上一页链接。

← → C ① 127.0.0.1:5000/explore?page=2

博客: 首页发现个人资料退出

Hi, susan!



上一页 第2页

博客: 首页 发现 个人资料 退出

Hi, susan!



第1页 下一页

4. 个人主页分页处理

4.1. 视图函数增加分页处理

修改 app/user.py 脚本,增加分页处理。

4.2. 模板添加分页导航

修改 app/templates/user/user_info.html 文件, 增减分页导航链接。

```
•••••
```

4.3. 启动服务测试

通过用户susan登录并新增一条帖子。

← → C (i) 127.0.0.1:5000

博客: 首页发现个人资料退出

Hi, susan!

内容		
		,

提交



第1页 <u>下一页</u>

点击个人资料链接, 也会有分页处理。

博客: 首页 发现 个人资料 退出



用户: susan

最后访问于:

2019-05-05 09:54:07.116215

粉丝 0, 关注 0

编辑您的资料

susan says:

<mark>susan says:</mark>

第1页 下一页

← → C ① 127.0.0.1:5000/user/info/susan?page=2

博客: 首页 发现 个人资料 退出



用户: susan

最后访问于:

2019-05-05 09:54:20.476830

粉丝 0, 关注 0

编辑您的资料



<u>-页</u> 第2页

5. 调整分页时每页数据条数参数

修改 app/config.py 脚本,将条数调整为每页20条。

每页展示数据条数

POSTS_PER_PAGE = 20