

A Related Matter:

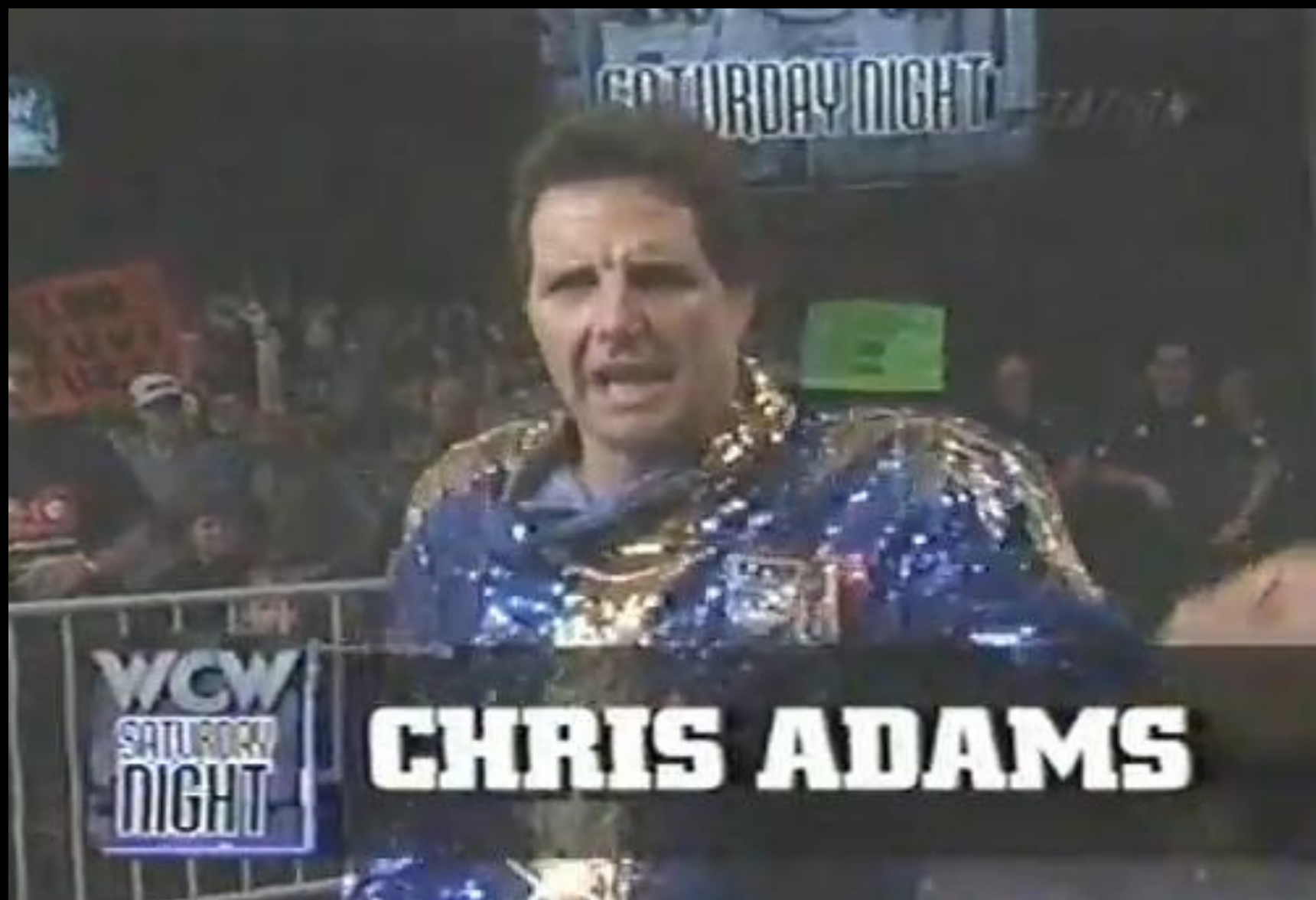
Optimizing your webapp by using django-debug-toolbar,
`select_related()`, and `prefetch_related()`

Christopher Adams
DjangoCon 2014

<https://github.com/adamsc64/a-related-matter>

Christopher Adams

- Software Engineer at Venmo
- Twitter/Github: @adamsc64
- I'm not Chris Adams (@acdha), who works at Library of Congress
- Neither of us are "The Gentleman" Chris Adams (90's-era Professional Wrestler)



The Django logo, featuring the word "django" in a white, lowercase, sans-serif font, centered within a dark green rounded rectangle.

django

Django is great

But Django is really a set of tools



Tools are great

But tools can be used in good or bad ways

The Django ORM:

A set of tools

Manage your own expectations for tools

- Many people approach a new tool with broad set of expectations as to what they think it will do for them.
- This may have little correlation with what the project actually has implemented.

As amazing as it
would be if they did...

Unicorns don't exist



The Django ORM: An abstraction layer

Abstraction layers

- Great because they take us away from the messy details
- Risky because they take us away from the messy details



Don't forget

You're far from the ground

The QuerySet API

QuerySets are Lazy

QuerySets are
Immutable

Lazy: Does not evaluate
until it needs to

Immutable: Never
itself changes

Each a new QuerySet, none hit the database

- `queryset = Model.objects.all()`
- `queryset = queryset.filter(...)`
- `queryset = queryset.values(...)`

Hits the database (QuerySet is “evaluated”):

- `queryset = list(queryset)`
- `queryset = queryset[:]`
- `for model_object in queryset:`
 `do_something(...)`

Our app: a blog

Models

```
class Blog(models.Model):  
    submitter = models.ForeignKey('auth.User')  
  
class Post(models.Model):  
    blog = models.ForeignKey('blog.Blog', related_name="posts")  
    likers = models.ManyToManyField('auth.User')  
  
class PostComment(models.Model):  
    submitter = models.ForeignKey('auth.User')  
    post = models.ForeignKey('blog.Post', related_name="comments")
```

List View

```
def blog_list(request):  
    blogs = Blog.objects.all()  
    return render(request, "blog/blog_list.html", {  
        "blogs": blogs,  
    })
```

List Template

```
1 {% for blog in blogs %}
2     <li>
3         <a href="{% url 'blog_detail' blog.id %}">
4             {{ blog.name }}
5         </a>
6         | submitted by {{ blog.submitter.username }}
7     </li>
8 {% endfor %}
```

Blogs:

- [dolor](#) | submitted by James
- [Duis](#) | submitted by Thomas
- [qui](#) | submitted by Isabella
- [irure](#) | submitted by Jack
- [magna](#) | submitted by Lachlan
- [occaecat](#) | submitted by Emma
- [reprehenderit](#) | submitted by James
- [Excepteur](#) | submitted by Oliver
- [non](#) | submitted by Noah
- [dolor](#) | submitted by William
- [veniam,](#) | submitted by Oliver
- [proident,](#) | submitted by Ella
- [aute](#) | submitted by Olivia
- [non](#) | submitted by William
- [magna](#) | submitted by Thomas
- [ut](#) | submitted by Lachlan
- [laborum.](#) | submitted by Thomas
- [fugiat](#) | submitted by Olivia
- [Lorem](#) | submitted by Olivia
- [culpa](#) | submitted by Cooper
- [non](#) | submitted by Thomas
- [sint](#) | submitted by Emily
- [adipiscing](#) | submitted by Isabella

Detail View

```
def blog_detail(request, blog_id):  
    blog = get_object_or_404(Blog, id=blog_id)  
    posts = Post.objects.filter(blog=blog)  
    return render(request, "blog/blog_detail.html", {  
        "blog": blog,  
        "posts": posts,  
    })
```

Detail Template

```
1 <h1>Blog '{{ blog.name }}' by {{ blog.submitter.username }}:</h1>
2 <ul>
3 {% for post in posts %}
4   <li>Post '{{ post.name }}'
5     {% if post.likers.exists %}
6     (liked by
7     {% for liker in post.likers.all %}
8       {{ liker.username }}
9     {% endfor %})
10    {% endif %}
11   <ul>
12     {% for comment in post.comments.all %}
13       <li>Comment by {{ comment.submitter.username }}:
14         {{ comment.body|truncatechars:60 }}</li>
15     {% endfor %}
16   </ul>
17 </li>
18 {% endfor %}
```

Blog 'magna' by Thomas:

- Post 'proident,' (liked by Lachlan Cooper William superuser)
 - Comment by superuser: Lorem ipsum dolor sit amet, consectetur adipiscing elit, ...
 - Comment by James: Lorem ipsum dolor sit amet, consectetur adipiscing elit, ...
 - Comment by Oliver: Lorem ipsum dolor sit amet, consectetur adipiscing elit, ...
 - Comment by Lachlan: Lorem ipsum dolor sit amet, consectetur adipiscing elit, ...
 - Comment by superuser: Lorem ipsum dolor sit amet, consectetur adipiscing elit, ...
 - Comment by Lucas: Lorem ipsum dolor sit amet, consectetur adipiscing elit, ...
 - Comment by Thomas: Lorem ipsum dolor sit amet, consectetur adipiscing elit, ...
- Post 'exercitation' (liked by James)
 - Comment by Emma: Lorem ipsum dolor sit amet, consectetur adipiscing elit, ...
 - Comment by William: Lorem ipsum dolor sit amet, consectetur adipiscing elit, ...
 - Comment by superuser: Lorem ipsum dolor sit amet, consectetur adipiscing elit, ...
 - Comment by Thomas: Lorem ipsum dolor sit amet, consectetur adipiscing elit, ...
 - Comment by Lily: Lorem ipsum dolor sit amet, consectetur adipiscing elit, ...
 - Comment by Ava: Lorem ipsum dolor sit amet, consectetur adipiscing elit, ...
 - Comment by William: Lorem ipsum dolor sit amet, consectetur adipiscing elit, ...
- Post 'qui' (liked by Lachlan James Oliver Ella Lily Jack)
 - Comment by Sophia: Lorem ipsum dolor sit amet, consectetur adipiscing elit, ...
 - Comment by Oliver: Lorem ipsum dolor sit amet, consectetur adipiscing elit, ...
 - Comment by superuser: Lorem ipsum dolor sit amet, consectetur adipiscing elit, ...
 - Comment by Lily: Lorem ipsum dolor sit amet, consectetur adipiscing elit, ...
 - Comment by Lucas: Lorem ipsum dolor sit amet, consectetur adipiscing elit, ...
 - Comment by James: Lorem ipsum dolor sit amet, consectetur adipiscing elit, ...
 - Comment by Ava: Lorem ipsum dolor sit amet, consectetur adipiscing elit, ...
- Post 'commodo' (liked by Lucas Isabella Ava Emma)

SQL Queries?

If you can't measure
it...

...you'd never know if
there are problems.

Django Debug Toolbar

- [Installation](#)
 - [Getting the code](#)
 - [Quick setup](#)
 - [Explicit setup](#)
- [Configuration](#)
 - [DEBUG_TOOLBAR_PATCH_SETTINGS](#)
 - [DEBUG_TOOLBAR_PANELS](#)
 - [DEBUG_TOOLBAR_CONFIG](#)
- [Tips](#)
 - [The toolbar isn't displayed!](#)
 - [Using the toolbar offline](#)
 - [Performance considerations](#)
- [Panels](#)
 - [Default built-in panels](#)
 - [Non-default built-in panels](#)
 - [Third-party panels](#)
 - [API for third-party panels](#)
- [Commands](#)
 - `debugsqlshell`
- [Change log](#)
 - [1.2](#)
 - [1.1](#)
 - [1.0](#)
- [Contributing](#)

First view:
The blog list page

Blogs:

- [dolor](#) | submitted by James
- [Duis](#) | submitted by Thomas
- [qui](#) | submitted by Isabella
- [irure](#) | submitted by Jack
- [magna](#) | submitted by Lachlan
- [occaecat](#) | submitted by Emma
- [reprehenderit](#) | submitted by James
- [Excepteur](#) | submitted by Oliver
- [non](#) | submitted by Noah
- [dolor](#) | submitted by William
- [veniam,](#) | submitted by Oliver
- [proident,](#) | submitted by Ella
- [aute](#) | submitted by Olivia
- [non](#) | submitted by William
- [magna](#) | submitted by Thomas
- [ut](#) | submitted by Lachlan
- [laborum.](#) | submitted by Thomas
- [fugiat](#) | submitted by Olivia
- [Lorem](#) | submitted by Olivia
- [culpa](#) | submitted by Cooper
- [non](#) | submitted by Thomas
- [sint](#) | submitted by Emily
- [adipiscing](#) | submitted by Isabella

Time

CPU: 561.78ms (588.91ms)

Settings

Headers

Request

BLOG_LIST

SQL

52 QUERIES IN 41.55MS

Static files

0 FILES USED

Templates

BLOG/BLOG_LIST.HTML

Cache

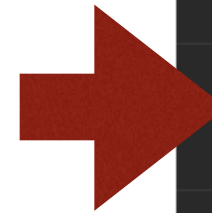
0 CALLS IN 0.00MS

Signals

8 RECEIVERS OF 12 SIGNALS

Blogs:

- [dolor](#) | submitted by James
- [Duis](#) | submitted by Thomas
- [qui](#) | submitted by Isabella
- [irure](#) | submitted by Jack
- [magna](#) | submitted by Lachlan
- [occaecat](#) | submitted by Emma
- [reprehenderit](#) | submitted by James
- [Excepteur](#) | submitted by Oliver
- [non](#) | submitted by Noah
- [dolor](#) | submitted by William
- [veniam,](#) | submitted by Oliver
- [proident,](#) | submitted by Ella
- [aute](#) | submitted by Olivia
- [non](#) | submitted by William
- [magna](#) | submitted by Thomas
- [ut](#) | submitted by Lachlan
- [laborum.](#) | submitted by Thomas
- [fugiat](#) | submitted by Olivia
- [Lorem](#) | submitted by Olivia
- [culpa](#) | submitted by Cooper
- [non](#) | submitted by Thomas
- [sint](#) | submitted by Emily
- [adipiscing](#) | submitted by Isabella



Time

CPU: 561.78ms (588.91ms)

Settings

Headers

Request

BLOG_LIST

SQL

52 QUERIES IN 41.55MS

Static files

0 FILES USED

Templates
















BLOG/BLOG_LIST.HTML

Cache

0 CALLS IN 0.00MS

Signals

8 RECEIVERS OF 12 SIGNALS

Query	Timeline
 SELECT ... FROM "auth_user" WHERE "auth_user"."id" = 1	
 SELECT ... FROM "blog_blog"	
 SELECT ... FROM "auth_user" WHERE "auth_user"."id" = 17	
 SELECT ... FROM "auth_user" WHERE "auth_user"."id" = 15	
 SELECT ... FROM "auth_user" WHERE "auth_user"."id" = 11	
 SELECT ... FROM "auth_user" WHERE "auth_user"."id" = 2	
 SELECT ... FROM "auth_user" WHERE "auth_user"."id" = 19	
 SELECT ... FROM "auth_user" WHERE "auth_user"."id" = 4	
 SELECT ... FROM "auth_user" WHERE "auth_user"."id" = 17	
 SELECT ... FROM "auth_user" WHERE "auth_user"."id" = 14	
 SELECT ... FROM "auth_user" WHERE "auth_user"."id" = 12	
 SELECT ... FROM "auth_user" WHERE "auth_user"."id" = 3	
 SELECT ... FROM "auth_user" WHERE "auth_user"."id" = 14	
 SELECT ... FROM "auth_user" WHERE "auth_user"."id" = 10	
 SELECT ... FROM "auth_user" WHERE "auth_user"."id" = 5	
 SELECT ... FROM "auth_user" WHERE "auth_user"."id" = 3	



```
SELECT "auth_user"."id", "auth_user"."password",  
"auth_user"."last_login", "auth_user"."is_superuser",  
"auth_user"."username", "auth_user"."first_name",  
"auth_user"."last_name", "auth_user"."email", "auth_user"."is_staff",  
"auth_user"."is_active", "auth_user"."date_joined" FROM "auth_user"  
WHERE "auth_user"."id" = 4
```

0.67

Sel

Expl

Connection: default

```
/Users/chris/coding/djangocon2014_project/adamsc64_djangocon2014/blog/views.py in blog_list(10)  
    "blogs": blogs,
```

```
7         <a href="{% url 'blog_detail' blog.id %}">  
8             {{ blog.name }}  
9         </a>  
10         | submitted by {{ blog.submitter.username }}  
11     </li>  
12 {% endfor %}  
13 </ul>
```

```
/Users/chris/coding/djangocon2014_project/adamsc64_djangocon2014/blog/templates/blog/blog_list.html
```



```
SELECT "auth_user"."id", "auth_user"."password",  
"auth_user"."last_login", "auth_user"."is_superuser",  
"auth_user"."username", "auth_user"."first_name",  
"auth_user"."last_name", "auth_user"."email", "auth_user"."is_staff",  
"auth_user"."is_active", "auth_user"."date_joined" FROM "auth_user"  
WHERE "auth_user"."id" = 4
```

0.67

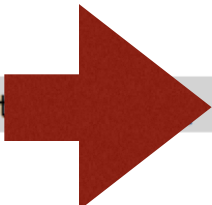
Sel

Expl

Connection: default

```
/Users/chris/coding/djangocon2014_project/adamsc64_djangocon2014/blog/views.py in blog_list(10)  
"blogs": blogs,
```

```
7         <a href="{% url 'blog_detail' blog.id %}">  
8             {{ blog.name }}  
9         </a>  
10         | submitter.username }}  
11     </li>  
12 {% endfor %}  
13 </ul>
```



```
/Users/chris/coding/djangocon2014_project/adamsc64_djangocon2014/blog/templates/blog/blog_list.html
```

select_related()

- **select_related** works by creating an SQL join and including the fields of the related object in the SELECT statement.
- For this reason, select_related gets the related objects in the **same database query**.
- However, to avoid the much larger result set that would result from joining across a 'many' relationship, select_related is limited to single-valued relationships - **foreign key and one-to-one**.

List View

```
def blog_list(request):  
    blogs = Blog.objects.all()  
    blogs = blogs.select_related("submitter")  
    return render(request, "blog/blog_list.html", {  
        "blogs": blogs,  
    })
```

SQL queries from 1 connection



default

5.49 ms (2 queries)

Query	Timeline	Time (ms)	Action
<div><div></div><div><div></div><div>SELECT ... FROM "auth_user" WHERE "auth_user"."id" = 1</div></div></div>		3.08	<div>SelExpl</div>
<div><div></div><div><div></div><div>SELECT ... FROM "blog_blog" INNER JOIN "auth_user" ON ("blog_blog"."submitter_id" = "auth_user"."id")</div></div></div>		2.40	<div>SelExpl</div>

Hide »

Versions

DJANGO 1.6.7

Time

CPU: 44.35ms (53.91ms)

Settings

Headers

Request

BLOG_LIST

SQL

2 QUERIES IN 5.49MS

Static files

0 FILES USED

Templates









BLOG/BLOG_LIST.HTML

SQL queries from 1 connection



default

5.49 ms (2 queries)

Query	Timeline	Time (ms)	Action
 SELECT ... FROM "auth_user" WHERE "auth_user"."id" = 1		3.08	 
 SELECT ... FROM "blog_blog" INNER JOIN "auth_user" ON ("blog_blog"."submitter_id" = "auth_user"."id")		2.40	 

Hide »

Versions

DJANGO 1.6.7

Time

CPU: 44.35ms (53.91ms)

Settings

Headers

Request

BLOG_LIST

SQL

2 QUERIES IN 5.49MS

Static files

0 FILES USED

Templates

BLOG/BLOG_LIST.HTML

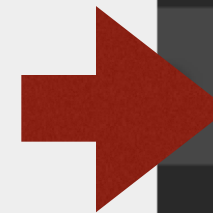
SQL queries from 1 connection



default

5.49 ms (2 queries)

Query	Timeline	Time (ms)	Action
SELECT ... FROM "auth_user" WHERE "auth_user"."id" = 1		3.08	
SELECT ... FROM "blog_blog" INNER JOIN "auth_user" ON ("blog_blog"."submitter_id" = "auth_user"."id")		2.40	



Hide »

Versions

DJANGO 1.6.7

Time

CPU: 44.35ms (53.91ms)

Settings

Headers

Request

BLOG_LIST

SQL

2 QUERIES IN 5.49MS

Static files

0 FILES USED

Templates

BLOG/BLOG_LIST.HTML

Second view:
The blog detail page

Blog 'magna' by Thomas:

- Post 'proident,' (liked by Lachlan Cooper William superuser)
 - Comment by superuser: Lorem ipsum dolor sit amet, consectetur adipiscing elit, ...
 - Comment by James: Lorem ipsum dolor sit amet, consectetur adipiscing elit, ...
 - Comment by Oliver: Lorem ipsum dolor sit amet, consectetur adipiscing elit, ...
 - Comment by Lachlan: Lorem ipsum dolor sit amet, consectetur adipiscing elit, ...
 - Comment by superuser: Lorem ipsum dolor sit amet, consectetur adipiscing elit, ...
 - Comment by Lucas: Lorem ipsum dolor sit amet, consectetur adipiscing elit, ...
 - Comment by Thomas: Lorem ipsum dolor sit amet, consectetur adipiscing elit, ...
- Post 'exercitation' (liked by James)
 - Comment by Emma: Lorem ipsum dolor sit amet, consectetur adipiscing elit, ...
 - Comment by William: Lorem ipsum dolor sit amet, consectetur adipiscing elit, ...
 - Comment by superuser: Lorem ipsum dolor sit amet, consectetur adipiscing elit, ...
 - Comment by Thomas: Lorem ipsum dolor sit amet, consectetur adipiscing elit, ...
 - Comment by Lily: Lorem ipsum dolor sit amet, consectetur adipiscing elit, ...
 - Comment by Ava: Lorem ipsum dolor sit amet, consectetur adipiscing elit, ...
 - Comment by William: Lorem ipsum dolor sit amet, consectetur adipiscing elit, ...
- Post 'qui' (liked by Lachlan James Oliver Ella Lily Jack)
 - Comment by Sophia: Lorem ipsum dolor sit amet, consectetur adipiscing elit, ...
 - Comment by Oliver: Lorem ipsum dolor sit amet, consectetur adipiscing elit, ...
 - Comment by superuser: Lorem ipsum dolor sit amet, consectetur adipiscing elit, ...
 - Comment by Lily: Lorem ipsum dolor sit amet, consectetur adipiscing elit, ...
 - Comment by Lucas: Lorem ipsum dolor sit amet, consectetur adipiscing elit, ...
 - Comment by James: Lorem ipsum dolor sit amet, consectetur adipiscing elit, ...
 - Comment by Ava: Lorem ipsum dolor sit amet, consectetur adipiscing elit, ...
- Post 'commodo' (liked by Lucas Isabella Ava Emma)

Hide »

Versions

DJANGO 1.6.7

Time

CPU: 464.95ms (500.62ms)

Settings

Headers

Request

BLOG_DETAIL

SQL

44 QUERIES IN 42.43MS

Static files

0 FILES USED

Templates

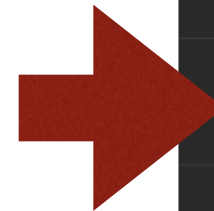
BLOG/BLOG_DETAIL.HTML

Cache

0 CALLS IN 0.00MS

Blog 'magna' by Thomas:

- Post 'proident,' (liked by Lachlan Cooper William superuser)
 - Comment by superuser: Lorem ipsum dolor sit amet, consectetur adipiscing elit, ...
 - Comment by James: Lorem ipsum dolor sit amet, consectetur adipiscing elit, ...
 - Comment by Oliver: Lorem ipsum dolor sit amet, consectetur adipiscing elit, ...
 - Comment by Lachlan: Lorem ipsum dolor sit amet, consectetur adipiscing elit, ...
 - Comment by superuser: Lorem ipsum dolor sit amet, consectetur adipiscing elit, ...
 - Comment by Lucas: Lorem ipsum dolor sit amet, consectetur adipiscing elit, ...
 - Comment by Thomas: Lorem ipsum dolor sit amet, consectetur adipiscing elit, ...
- Post 'exercitation' (liked by James)
 - Comment by Emma: Lorem ipsum dolor sit amet, consectetur adipiscing elit, ...
 - Comment by William: Lorem ipsum dolor sit amet, consectetur adipiscing elit, ...
 - Comment by superuser: Lorem ipsum dolor sit amet, consectetur adipiscing elit, ...
 - Comment by Thomas: Lorem ipsum dolor sit amet, consectetur adipiscing elit, ...
 - Comment by Lily: Lorem ipsum dolor sit amet, consectetur adipiscing elit, ...
 - Comment by Ava: Lorem ipsum dolor sit amet, consectetur adipiscing elit, ...
 - Comment by William: Lorem ipsum dolor sit amet, consectetur adipiscing elit, ...
- Post 'qui' (liked by Lachlan James Oliver Ella Lily Jack)
 - Comment by Sophia: Lorem ipsum dolor sit amet, consectetur adipiscing elit, ...
 - Comment by Oliver: Lorem ipsum dolor sit amet, consectetur adipiscing elit, ...
 - Comment by superuser: Lorem ipsum dolor sit amet, consectetur adipiscing elit, ...
 - Comment by Lily: Lorem ipsum dolor sit amet, consectetur adipiscing elit, ...
 - Comment by Lucas: Lorem ipsum dolor sit amet, consectetur adipiscing elit, ...
 - Comment by James: Lorem ipsum dolor sit amet, consectetur adipiscing elit, ...
 - Comment by Ava: Lorem ipsum dolor sit amet, consectetur adipiscing elit, ...
- Post 'commodo' (liked by Lucas Isabella Ava Emma)



Hide »

Versions

DJANGO 1.6.7

Time

CPU: 464.95ms (500.62ms)

Settings

Headers

Request

BLOG_DETAIL

SQL

44 QUERIES IN 42.43MS

Static files

0 FILES USED

Templates

BLOG/BLOG_DETAIL.HTML

Cache

0 CALLS IN 0.00MS

SQL queries from 1 connection



default

42.56 ms (44 queries)

Query	Timeline	Time (ms)	Action
SELECT ... FROM "blog_blog" WHERE "blog_blog"."id" = 15		2.45	Sel Expl
SELECT ... FROM "auth_user" WHERE "auth_user"."id" = 1		2.45	Sel Expl
SELECT ... FROM "auth_user" WHERE "auth_user"."id" = 15		0.76	Sel Expl
SELECT ... FROM "blog_post" WHERE "blog_post"."blog_id" = 15		1.50	Sel Expl
SELECT ... FROM "auth_user" INNER JOIN "blog_post_likers" ON ("auth_user"."id" = "blog_post_likers"."user_id") WHERE "blog_post_likers"."post_id" = 60 LIMIT 1		1.75	Sel Expl
SELECT ... FROM "auth_user" INNER JOIN "blog_post_likers" ON ("auth_user"."id" = "blog_post_likers"."user_id") WHERE "blog_post_likers"."post_id" = 60		1.57	Sel Expl
SELECT ... FROM "blog_postcomment" WHERE "blog_postcomment"."post_id" = 60		1.26	Sel Expl
SELECT ... FROM "auth_user" WHERE "auth_user"."id" = 1		0.77	Sel Expl
SELECT ... FROM "auth_user" WHERE "auth_user"."id" = 17		0.95	Sel Expl
SELECT ... FROM "auth_user" WHERE "auth_user"."id" = 14		0.73	Sel Expl
SELECT ... FROM "auth_user" WHERE "auth_user"."id" = 19		0.84	Sel Expl
SELECT ... FROM "auth_user" WHERE "auth_user"."id" = 1		0.99	Sel Expl
SELECT ... FROM "auth_user" WHERE "auth_user"."id" = 18		0.67	Sel Expl
SELECT ... FROM "auth_user" WHERE "auth_user"."id" = 15		1.18	Sel Expl
SELECT ... FROM "auth_user" INNER JOIN "blog_post_likers" ON ("auth_user"."id" = "blog_post_likers"."user_id") WHERE "blog_post_likers"."post_id" = 59 LIMIT 1		1.06	Sel Expl
SELECT ... FROM "auth_user" INNER JOIN "blog_post_likers" ON ("auth_user"."id" = "blog_post_likers"."user_id") WHERE "blog_post_likers"."post_id" = 59		1.27	Sel Expl
SELECT ... FROM "blog_postcomment" WHERE "blog_postcomment"."post_id" = 59		0.64	Sel Expl

```
- SELECT "auth_user"."id", "auth_user"."password",  
  "auth_user"."last_login", "auth_user"."is_superuser",  
  "auth_user"."username", "auth_user"."first_name",  
  "auth_user"."last_name", "auth_user"."email", "auth_user"."is_staff",  
  "auth_user"."is_active", "auth_user"."date_joined" FROM "auth_user"  
WHERE "auth_user"."id" = 14
```

0.73

Sel

Expl

Connection: default

```
/Users/chris/coding/djangocon2014_project/adamsc64_djangocon2014/blog/views.py in blog_detail(21)  
    "posts": posts,
```

```
12         {% endif %}  
13     <ul>  
14         {% for comment in post.comments.all %}  
15             <li>Comment by {{ comment.submitter.username }}:  
16                 {{ comment.body|truncatechars:60 }}</li>  
17         {% endfor %}  
18     </ul>
```

```
/Users/chris/coding/djangocon2014_project/adamsc64_djangocon2014/blog/templates/blog/blog_detail.html
```

```
- SELECT "auth_user"."id", "auth_user"."password",  
  "auth_user"."last_login", "auth_user"."is_superuser",  
  "auth_user"."username", "auth_user"."first_name",  
  "auth_user"."last_name", "auth_user"."email", "auth_user"."is_staff",  
  "auth_user"."is_active", "auth_user"."date_joined" FROM "auth_user"  
WHERE "auth_user"."id" = 14
```

0.73


Sel

Expl

Connection: default

```
/Users/chris/coding/djangocon2014_project/adamsc64_djangocon2014/blog/views.py in blog_detail(21)  
    "posts": posts,
```

```
12         {% endif %}  
13     <ul>  
14         {% for comment in post.comments.all %}  
15         <li>Comment by {{ comment.submitter.username }}:  
16             {{ comment.body|truncatechars:60 }}</li>  
17         {% endfor %}  
18     </ul>
```



```
/Users/chris/coding/djangocon2014_project/adamsc64_djangocon2014/blog/templates/blog/blog_detail.html
```


 **SELECT** "auth_user"."id", "auth_user"."password",
"auth_user"."last_login", "auth_user"."is_superuser",
"auth_user"."username", "auth_user"."first_name",
"auth_user"."last_name", "auth_user"."email", "auth_user"."is_staff",
"auth_user"."is_active", "auth_user"."date_joined" **FROM** "auth_user"
INNER JOIN "blog_post_likers" **ON** ("auth_user"."id" =
"blog_post_likers"."user_id") **WHERE** "blog_post_likers"."post_id" = 58

1.06

Sel

Expl

Connection: default

/Users/chris/coding/djangocon2014_project/adamsc64_djangocon2014/blog/views.py in blog_detail(21)
"posts": posts,

```
6         <li>Post '{{ post.name }}'  
7             {% if post.likers.exists %}  
8             (liked by  
9                 {% for liker in post.likers.all %}  
10                 '{{ liker.username }}'  
11             {% endfor %})  
12             {% endif %}
```

/Users/chris/coding/djangocon2014_project/adamsc64_djangocon2014/blog/templates/blog/blog_detail.html

```
SELECT "auth_user"."id", "auth_user"."password",
"auth_user"."last_login", "auth_user"."is_superuser",
"auth_user"."username", "auth_user"."first_name",
"auth_user"."last_name", "auth_user"."email", "auth_user"."is_staff",
"auth_user"."is_active", "auth_user"."date_joined" FROM "auth_user"
INNER JOIN "blog_post_likers" ON ( "auth_user"."id" =
"blog_post_likers"."user_id" ) WHERE "blog_post_likers"."post_id" = 58
```

1.06

Sel

Expl

Connection: default

```
/Users/chris/coding/djangocon2014_project/adamsc64_djangocon2014/blog/views.py in blog_detail(21)
    "posts": posts,
```

```
6         <li>Post '{{ post.name }}'
7             {% if post.likers.exists %}
8             (liked by
9             {% for liker in post.likers.all %}
10                '{{ liker.username }}'
11            {% endfor %})
12            {% endif %}
```

```
/Users/chris/coding/djangocon2014_project/adamsc64_djangocon2014/blog/templates/blog/blog_detail.html
```

prefetch_related()

- **prefetch_related** does a separate lookup for each relationship, and does the 'joining' in Python.
- This allows it to prefetch many-to-many and many-to-one objects ... **in addition to** the foreign key and one-to-one relationships.
- It also supports prefetching of GenericRelation and GenericForeignKey.















Detail View

```
def blog_detail(request, blog_id):  
    blog = get_object_or_404(Blog, id=blog_id)  
    posts = Post.objects.filter(blog=blog)  
    posts = posts.prefetch_related(  
        "comments__submitter", "likers",  
    )  
    return render(request, "blog/blog_detail.html", {  
        "blog": blog,  
        "posts": posts,  
    })
```

SQL queries from 1 connection

default

12.16 ms (7 queries)

Query	Timeline	Time (ms)	Action
 SELECT ... FROM "blog_blog" WHERE "blog_blog"."id" = 15		3.58	Sel Expl
 SELECT ... FROM "auth_user" WHERE "auth_user"."id" = 1		2.09	Sel Expl
 SELECT ... FROM "auth_user" WHERE "auth_user"."id" = 15		0.69	Sel Expl
 SELECT ... FROM "blog_post" WHERE "blog_post"."blog_id" = 15		1.06	Sel Expl
 SELECT ... FROM "blog_postcomment" WHERE "blog_postcomment"."post_id" IN (60, 59, 58, 57)		1.70	Sel Expl
 SELECT ... FROM "auth_user" WHERE "auth_user"."id" IN (1, 2, 3, 4, 7, 8, 9, 10, 14, 15, 16, 17, 18, 19)		0.95	Sel Expl
 SELECT ... FROM "auth_user" INNER JOIN "blog_post_likers" ON ("auth_user"."id" = "blog_post_likers"."user_id") WHERE "blog_post_likers"."post_id" IN (60, 59, 58, 57)		2.09	Sel Expl

Hide »

Versions

DJANGO 1.6.7

Time

CPU: 86.12ms (100.47ms)

Settings

Headers

Request

BLOG_DETAIL

SQL

7 QUERIES IN 12.16ms

Static files

0 FILES USED

Templates

BLOG/BLOG_DETAIL.HTML

Cache

Summary

- The QuerySet API methods `select_related()` and `prefetch_related()` automate some best practices to avoid extra queries in views/templates.
- Use `select_related()` for one-to-many or one-to-one relations.
- Use `prefetch_related()` for many-to-many or many-to-one (e.g. reverse foreign key) relations.

Thanks!

Christopher Adams (@adamsc64)

<https://github.com/adamsc64/a-related-matter>