

WordPress Transients API

What we'll cover

- Introduction
- What's a query?
- Why reduce them?
- Testing speed
- Real example
- The transient API
- Other uses: connecting to other sites
- Things to look out for
- Questions?

Introduction

- Adam Maltpress – www.maltpress.co.uk
- PHP developer focussing on WordPress theme and plugin development
- @maltpress

What's a query?

- Database stores posts, settings... data, not presentational stuff
- User visits page – browser asks server for it
- PHP file contains the “questions” to ask the database
- Server looks through database for relevant stuff
- PHP file sticks results together with HTML and sends to browser

Why reduce queries?

- Every query takes time and resources
- More queries = longer time to first byte (TTFB) = slower site speed
- Slower speed is bad for users, bad for SEO

Testing speed

- Count the number of queries:
 - `get_num_queries();`
 - Doesn't necessarily reflect speed: some server setups can do lots of queries fast
- Time it:
 - `timer_stop($display = 0, $precision = 3);`
 - Better reflects actual speed
 - Doesn't give any clues about speeding it up

Testing speed

- In footer.php:
- `<?php echo get_num_queries(); ?> queries in <?php timer_stop(1); ?> second`

Testing speed

- Using native PHP functions:

```
• $start = microtime( true );  
  // do your functions  
  $time_elapsed_seconds = microtime( true ) -  
  $start;  
  echo $time_elapsed_seconds;
```


Real example

- Simple loop

```
if( have_posts() ) :  
while( have_posts() ) :  
the_post();  
?>  
<h1><?php the_title(); ?></h1>  
<p><?php the_excerpt(); ?></p>  
<span><?php echo get_post_meta( $post->  
ID, '_ex_meta_example', true ); ?  
></span>  
<?php endwhile; endif; ?>
```

Real example

- WP_Query (the basis for the loop) does one query to get matching posts
- This query triggers other queries – writing to WP internal caches, checking if pages are home pages etc, checking settings
- Most common template tags use existing \$post object...
- get_post_meta uses memcache but may do query... for each post

The transient API

- Is there a transient set? Is it in date? Get that string, use it
- If not, do your function to *create* the string
- Use that *new* string, and **also** store it as a transient
- Calling a transient uses 1 DB call... for a string which may originally have taken lots!

The transient API

- `set_transient($transient, $value, $expiration);`
- Expiration is in seconds
- You can use WP constants for times:
 - `MINUTE_IN_SECONDS = 60 (seconds)`
 - `HOUR_IN_SECONDS = 60 * MINUTE_IN_SECONDS`
 - `DAY_IN_SECONDS = 24 * HOUR_IN_SECONDS`
 - `WEEK_IN_SECONDS = 7 * DAY_IN_SECONDS`
 - `MONTH_IN_SECONDS = 30 * DAY_IN_SECONDS`
 - `YEAR_IN_SECONDS = 365 * DAY_IN_SECONDS`

A real example

- ```
if (false === ($query_results =
get_transient('query_results'))) {
if(have_posts()) :
while(have_posts()) : the_post();
$query_results =
"<h1>" . get_the_title() . "</h1>"
 . "<p>" . get_the_excerpt() . "</p>"
 . "" . get_post_meta($post->ID,
 'ex_meta_example', true) . "";
endwhile; endif;
set_transient('query_results',
$query_results, 12 * HOUR_IN_SECONDS);
}

echo $query_results;
```



# Other uses

- When connecting to other sites using `wp_remote_get`, `wp_remote_post` etc
  - Every connection to a remote site takes time
  - Busy sites can hit rate limits
  - Save the output of `wp_remote_post` (or processed output) to a transient
- Any data/processing heavy task which spits out a string can be saved to a transient!

# Things to look for

- Don't forget to update your transients if things change (a new post, for example) – hook `save_post` to `delete_transient`:
  - ```
function ex_update_transient() {  
    delete_transient( 'query_result' );  
}  
add_action( 'save_post', 'ex_update_transient' );
```
- AJAX and caching don't always work well
- ...and be careful storing carts in transients. Link to a user!
 - ```
set_transient('transient_' . $userid....);
```



# Questions?