

What I Wish I'd Known When I Started

Erick Hitter
@ethitter
<https://ethitter.com/>

AUTOMATTIC



FIRST

Can't check if the user's
logged in until `init`

- May work before because WP's trying to protect you
- Don't try to conditionally load plugin files at `plugins_loaded`, for example

SECOND

The query isn't
available until wp

- Can't use query conditional tags earlier, like at init
- `is_admin()` is an exception

THIRD

Never build a manual link again

- WordPress is a CMS, it's dynamic
- Domains change, permalink structures change
- Plugins and themes shouldn't break because of a URL

Link Functions

- `get_permalink()`
- `get_page_by_path('about')`
- `get_post_type_archive_link('waffles')`
- `user_trailingslashit()`
- `get_year_link()`
- `get_adjacent_post()`

Link Functions

- `add_query_arg()`
- `remove_query_arg()`

Link Functions

- `home_url('/')`

Many more in
`wp-includes/link-template.php`.

FOURTH

Escaping and Sanitization

- Stephane speaks at 3:15pm
- WordPress has lots of functions to protect against common security vulnerabilities
- Never trust the user

FIFTH

Nonces

- If you're building a form, you should know about these. CSRF is bad.
- Again, Stephane speaks at 3:15pm

SIXTH

Ajax is easy with WordPress

- Natively supported

Ajax

- Hook your function to one of two variable actions
- Use the same action name with the request to `admin-ajax.php`
- `check_ajax_referer()` or use a nonce

Ajax

- `wp_ajax_{your_action}`
- `wp_ajax_nopriv_{your_action}`

- Hook to both if logged-in state isn't relevant.

SEVENTH

Database Interactions

- Don't

Database Interactions

- Use custom post types and custom taxonomies instead.
- If you must, always `$wpdb->prepare()` your queries.

- CPTs and CTs have extensive APIs, much easier to deal with
- More portable that way, too
- Many caching benefits in the right environments
- If you must, `$wpdb->prepare()` uses `sprintf`-style notation to secure things

Database Interactions

- `$wpdb->get_var()`
- `$wpdb->get_col()`
- `$wpdb->get_row()`
- `$wpdb->insert($table, $data, $format)`
- `$wpdb->update($table, $data, $where, $format, $where_format)`
- `$wpdb->query()`

- Just some examples
- Simplify interactions, make it easy to deal with sets of data
- Simpler interactions than writing SQL

EIGHTH

Enqueue All The Things

- Rarely a reason to write a script or link tag for static assets

Register, then enqueue!

- `wp_register_style()`
- `wp_register_script()`

- Makes them ready for reuse, but doesn't load them yet
- Then use with conditional tags, or in page, to load when needed
- Pagespeed and other tools consider the number of files loaded

Enqueue!

- `wp_enqueue_style()`
- `wp_enqueue_script()`

- Use already-registered scripts
- Either your own, or those Core includes
- Enqueue a script you know you need

But why?

- Reusable
- Dependencies
- Versioning
- Minification
- Concatenation
- CDN

- Won't load seven copies of same script
- Ensure jQuery, Backbone, or whatever is loaded
- Bust caches on update
- Many hosts optimize assets when loaded this way

NINTH

Cache All The Things

- Performance
- Less resource usage, potential cost savings

Caching

- Transients
- Object Cache

- Explain difference
- Persistent cache implication
- Use transients unless you know object cache will always be there

TENTH

Remote Requests

- Always use the API!

WP HTTP API

- `wp_remote_get()`
- `wp_remote_post()`
- `wp_remote_head()`
- `wp_remote_request()`
- `wp_remote_retrieve_response_code()`

- Cached
- Selects best available method

ELEVENTH

Miscellany

Miscellany

- `get_queried_object()`
- `get_queried_object_id()`
- `wp_parse_args()`

- `get_queried_object()/get_queried_object_id()` – useful in a template that's shared across multiple views
- `wp_parse_args()` – defaults, avoid notices when keys aren't defined

Thanks

<https://eth.pw/wcywg15>

Erick Hitter
@ethitter
<https://ethitter.com/>

AUTOMATTIC