Extending the WordPress REST API

iThemes | timothybjacobs.com

History

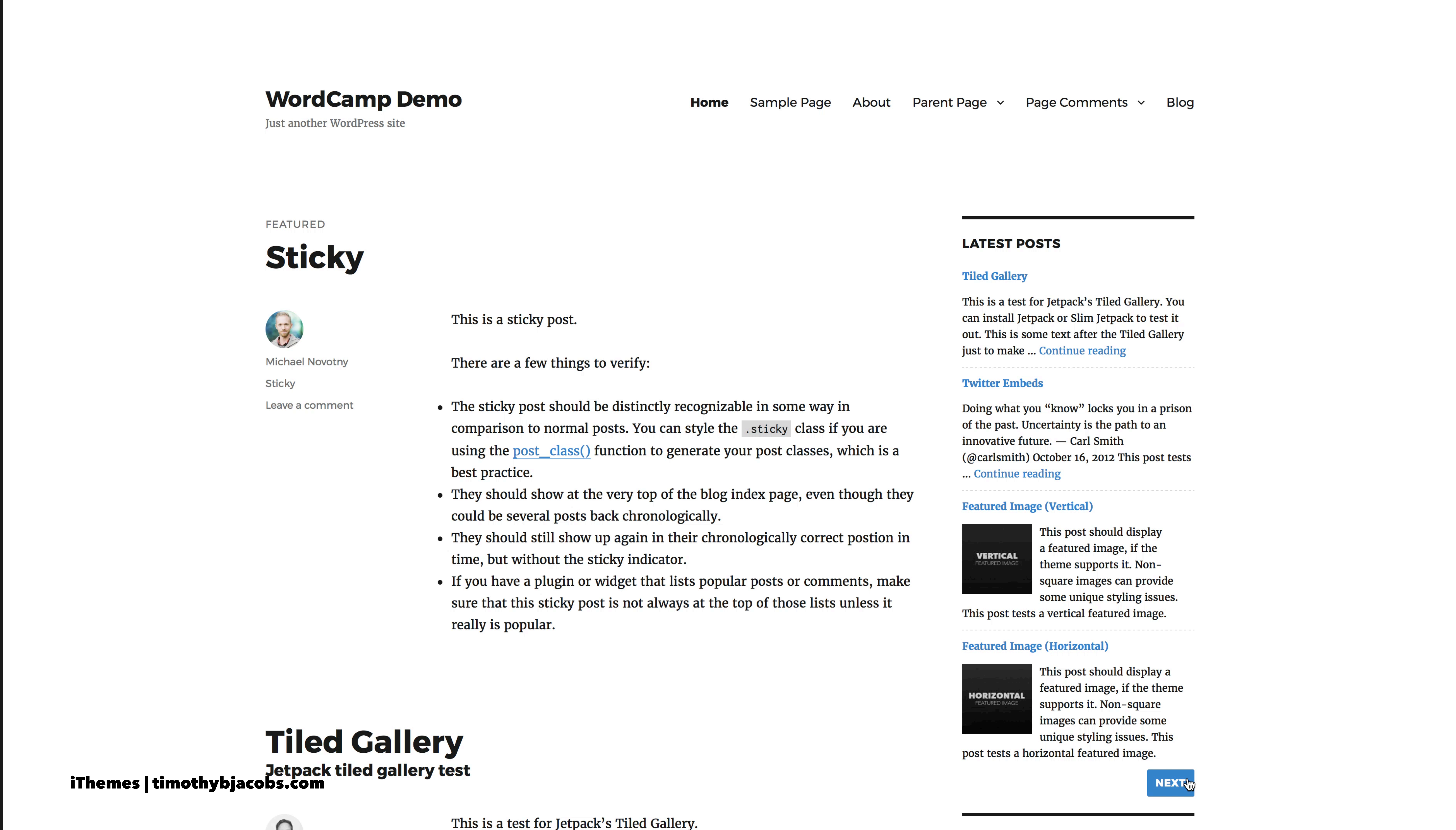
- REST API framework introduced in WordPress 4.4
  - Plugin authors can begin to use the REST API.
  - Content endpoints available in a plugin.
- Content endpoints introduced in WordPress 4.7
  - Additional infrastructure code makes it much easier to build custom endpoints.
  - Easy integration with custom post types, taxonomies, etc…

Built in Endpoints

- Posts & Pages
- Comments
- Terms
- Users
- Media
- Many more

Latest Posts

- WordPress has a built in Recent Posts widget.
- Doesn’t need to be rendered when the page loads.
- Can load latest posts asynchronously using JavaScript.
- Allows for increase interactivity.



FEATURED

Sticky

Michael Novotny

Sticky

Leave a comment

This is a sticky post.

There are a few things to verify:

- The sticky post should be distinctly recognizable in some way in comparison to normal posts. You can style the `.sticky` class if you are using the post_class() function to generate your post classes, which is a best practice.
- They should show at the very top of the blog index page, even though they could be several posts back chronologically.
- They should still show up again in their chronologically correct postion in time, but without the sticky indicator.
- If you have a plugin or widget that lists popular posts or comments, make sure that this sticky post is not always at the top of those lists unless it really is popular.

Tiled Gallery

Jetpack tiled gallery test

This is a test for Jetpack's Tiled Gallery.

LATEST POSTS

Tiled Gallery

This is a test for Jetpack's Tiled Gallery. You can install Jetpack or Slim Jetpack to test it out. This is some text after the Tiled Gallery just to make … Continue reading

Twitter Embeds

Doing what you "know" locks you in a prison of the past. Uncertainty is the path to an innovative future. — Carl Smith (@carlsmith) October 16, 2012 This post tests … Continue reading

Featured Image (Vertical)

This post should display a featured image, if the theme supports it. Non-square images can provide some unique styling issues. This post tests a vertical featured image.

Featured Image (Horizontal)

This post should display a featured image, if the theme supports it. Non-square images can provide some unique styling issues. This post tests a horizontal featured image.

Authors Widget

- API provides public access to post authors in the `users` endpoint.
- Could also be implemented in PHP.
- Has same benefits interactivity and performance benefits.

Supporting Metadata

- Example: A post subtitle or “featuring” authors.
- Two ways to include metadata.

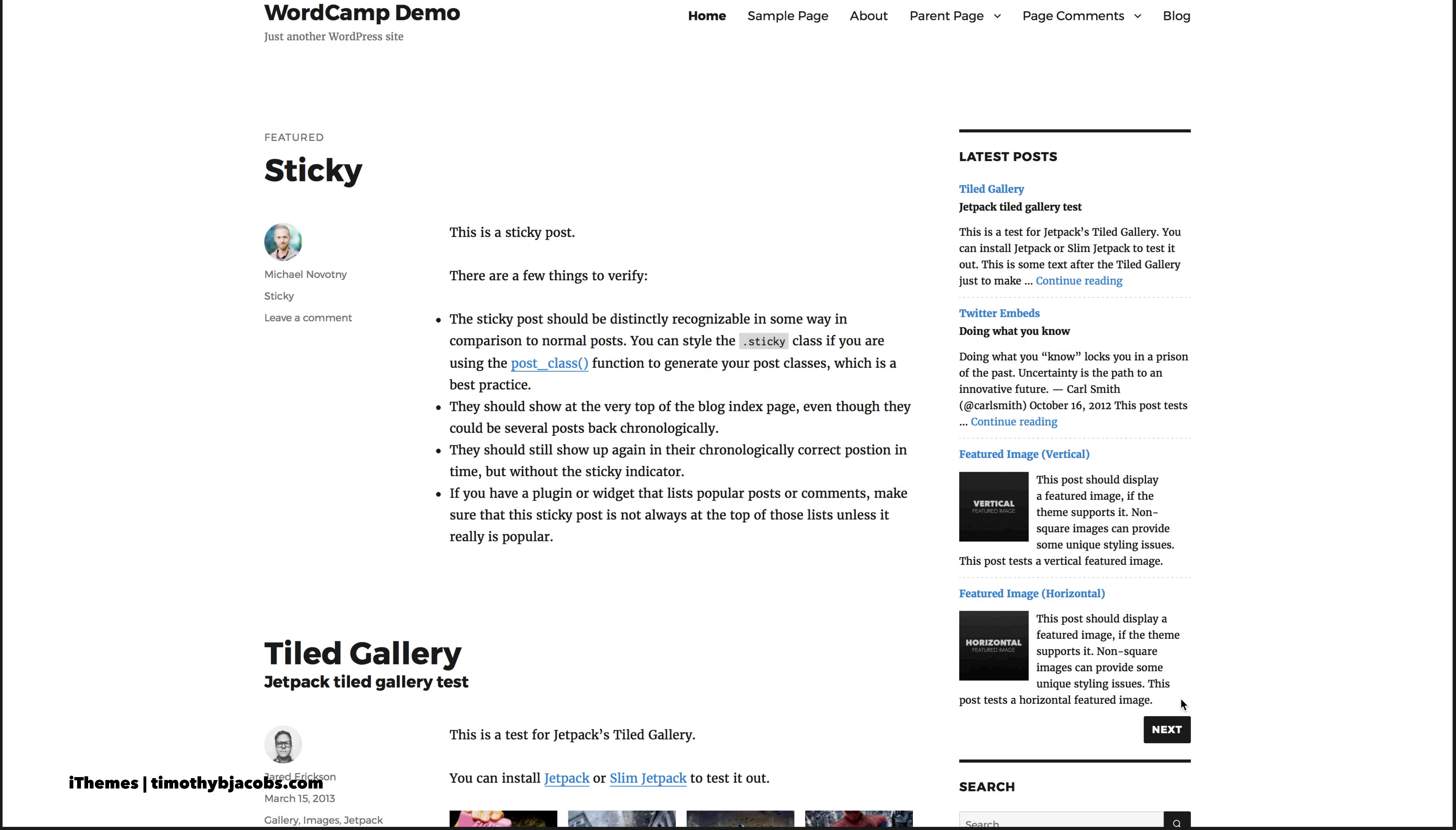
Register Meta

- `register_meta()`
- Expanded in WordPress 4.6

```
function register_meta( $object_type, $meta_key, $args ) {
    // ...
}

register_meta( 'post', '_subtitle', [
    'show_in_rest' => true,
    'single' => true,
    'type' => 'string',
] );
```

```
{
  "id": 5,
  "title": {
    "rendered": "The Post Title"
  },
  "meta": {
    "_subtitle": "The subtitle of the post."
  }
}
```



WordCamp Demo

Just another WordPress site

FEATURED

Sticky

Michael Novotny

Sticky

Leave a comment

This is a sticky post.

There are a few things to verify:

- The sticky post should be distinctly recognizable in some way in comparison to normal posts. You can style the `.sticky` class if you are using the post_class() function to generate your post classes, which is a best practice.
- They should show at the very top of the blog index page, even though they could be several posts back chronologically.
- They should still show up again in their chronologically correct postion in time, but without the sticky indicator.
- If you have a plugin or widget that lists popular posts or comments, make sure that this sticky post is not always at the top of those lists unless it really is popular.

Tiled Gallery

Jetpack tiled gallery test

Jared Erickson

March 15, 2013

Gallery, Images, Jetpack

This is a test for Jetpack's Tiled Gallery.

You can install Jetpack or Slim Jetpack to test it out.

LATEST POSTS

Tiled Gallery

Jetpack tiled gallery test

This is a test for Jetpack's Tiled Gallery. You can install Jetpack or Slim Jetpack to test it out. This is some text after the Tiled Gallery just to make … Continue reading

Twitter Embeds

Doing what you know

Doing what you "know" locks you in a prison of the past. Uncertainty is the path to an innovative future. — Carl Smith (@carlsmith) October 16, 2012 This post tests … Continue reading

Featured Image (Vertical)

This post should display a featured image, if the theme supports it. Non-square images can provide some unique styling issues. This post tests a vertical featured image.

Featured Image (Horizontal)

This post should display a featured image, if the theme supports it. Non-square images can provide some unique styling issues. This post tests a horizontal featured image.

SEARCH

Validation & Sanitization

- Meta data is automatically sanitized and validated on update.
- `string`, `integer`, `number` and `boolean` types are supported.

Validation & Sanitization

- Meta data is automatically sanitized and validated on update.
- `string`, `integer`, `number` and `boolean` types are supported.
  - `integer` and `number` values can be passed as strings.
  - `boolean` accepts `0`, `1`, `'false'` and `'true'`.

Validation & Sanitization

- Meta data is automatically sanitized and validated on update.
- `string`, `integer`, `number` and `boolean` types are supported.
- Can express a more complex constraint with JSON Schema.

JSON Schema

- Provides more complex validation beyond just a data type.
- Starts with a data type.
- Extended with custom keywords.
- WordPress supports a limited subset of JSON Schema out of the box.

```
register_meta( 'post', '_price', [
    'show_in_rest' => [
        'schema' => [
            'type' => 'number',
            'minimum' => 1.00,
        ]
    ]
] );
```

```
register_meta( 'post', '_available_after', [
    'show_in_rest' => [
        'schema' => [
            'type' => 'string',
            'format' => 'date-time',
        ]
    ]
] );
```

https://timothybjacobs.com/2017/05/17/json-schema-and-the-wp-rest-api/

Validation & Sanitization

- Meta data is automatically sanitized and validated on update.
- `string`, `integer`, `number` and `boolean` types are supported.
- Can express a more complex constraint with JSON Schema.
- Provide own sanitization function with `sanitize_callback`.

Drawbacks

- Exposes implementation details.
- Makes it difficult to refactor or provide filtering support.

```
function get_the_subtitle( $id ) {
   return get_post_meta( $id, '_subtitle', true );
}
```

```
function get_the_subtitle( $id ) {
   return get_post( $id )->post_subtitle;
}
```

```
function get_the_subtitle( $id ) {
   $subtitle = get_post_meta( $id, '_subtitle', true );

   return apply_filters( 'subtitle', $subtitle );
}
```

Drawbacks

- Exposes implementation details.
- Makes it difficult to refactor or provide filtering support.
- Currently can only be registered across all post types.
- Can’t provide a custom validation function.
- Can’t support objects or arrays.

Register Rest Field

- Can be registered for a specific route.
- Objects and arrays are supported.
- More sensible naming.

```
function register_rest_field( $object_type, $attribute, $args ) {
    // ...
}
```

```
register_rest_field( 'post', 'subtitle', [
    'get_callback' => 'get_the_subtitle',
    'update_callback' => function( $value, $post ) {
        update_post_meta( $post['id'], '_subtitle', $value );
    },
    'schema' => [
        'type' => 'string',
        'args' => [
            'validate_callback' => function( $value ) {
                return ! empty( $value );
            }
        ]
] );
```

```json
{
  "id": 5,
  "title": {
    "rendered": "The Post Title"
  },
  "subtitle": "The subtitle of the post."
}
```

Drawbacks

- Requires more code.

Custom Post Types

- `show_in_rest` automatically registers routes in the `wp/v2` namespace.
- Includes the same fields as the posts route.
- Respects post type feature support.
- Use `register_rest_field` to provide custom data from post meta.

Default Routes

- `/wp/v2/posts/`
- `/wp/v2/posts/5/`
- `/wp/v2/posts/5/revisions/`
- `/wp/v2/posts/5/revisions/15/`

Customization

- `rest_base` customizes the routes.

```
/wp/v2/$rest_base/

/wp/v2/$rest_base/{post_id}/
```

Customization

- `rest_base` customizes the routes.
- `rest_prepare_{$post_type}` filters the response for each post.
- Can completely customize the route by subclassing `WP_REST_Posts_Controller`.

Customization: What Not to Do

- Don’t completely change the structure of the response.
- Keep the same semantics for fields and query parameters.
- Core in the future (Gutenberg) and other apps will expect different post types to be consistent.

Example

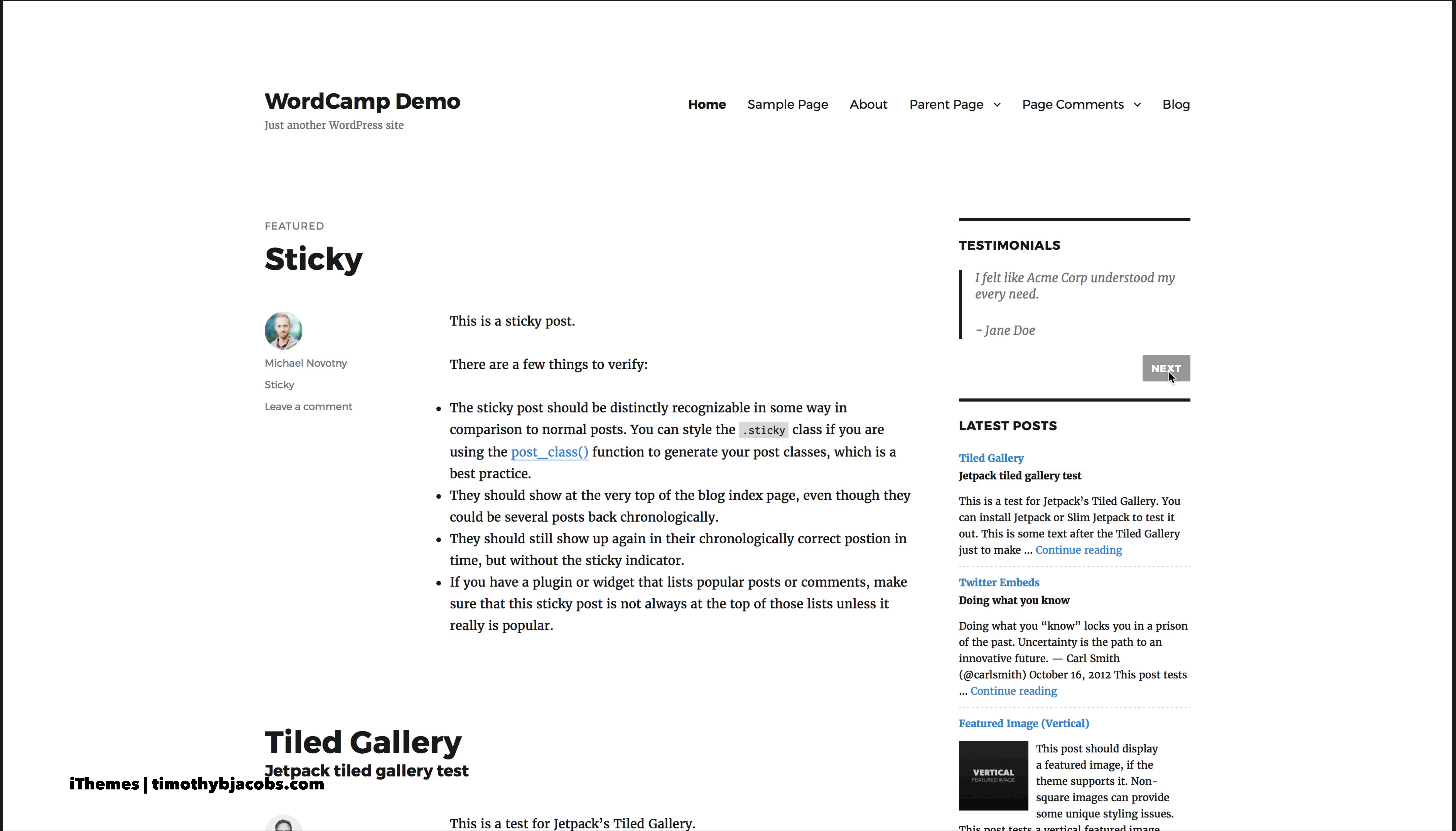
- Testimonials custom post type.
- Only supports the content field.

```
register_post_type( 'testimonial', [
    'supports'     => [ 'editor' ],
    'show_in_rest' => true,
    'rest_base'    => 'testimonials',
    'public'       => false,
    'show_ui'      => true,
] );
```

```
register_rest_field( 'testimonial', 'client', [
    'get_callback'    => function ( $post ) {
        return get_post_meta( $post['id'], '_client', true );
    },
    'update_callback' => function ( $value, $post ) {
        update_post_meta( $post['id'], '_client', wp_strip_all_tags( $value ) );
    },
    'schema'          => [
        'type'        => 'string',
        'required'    => true,
        'minLength'   => 1,
        'description' => 'The client who gave the testimonial',
    ]
] );
```

```
register_rest_field( 'testimonial', 'client_head_shot', [
    'get_callback'    => function ( $post ) {
        return get_post_meta( $post['id'], '_client_head_shot', true );
    },
    'update_callback' => function ( $value, $post ) {
        update_post_meta( $post['id'], '_client_head_shot', $value );
    },
    'schema'          => [
        'type'        => 'string',
        'format'      => 'uri',
        'description' => 'A head shot of the client.',
    ]
] );
```

```json
{
  "id": 1286,
  "content": {
    "rendered": "I felt like Acme Corp understood my every need.",
    "protected": false
  },
  "client": "Jane Doe",
  "client_head_shot": ""
}
```



FEATURED

Sticky

This is a sticky post.

There are a few things to verify:

- The sticky post should be distinctly recognizable in some way in comparison to normal posts. You can style the `.sticky` class if you are using the post_class() function to generate your post classes, which is a best practice.
- They should show at the very top of the blog index page, even though they could be several posts back chronologically.
- They should still show up again in their chronologically correct postion in time, but without the sticky indicator.
- If you have a plugin or widget that lists popular posts or comments, make sure that this sticky post is not always at the top of those lists unless it really is popular.

Tiled Gallery

Jetpack tiled gallery test

This is a test for Jetpack's Tiled Gallery.

TESTIMONIALS

> *I felt like Acme Corp understood my every need.*
>
> *– Jane Doe*

LATEST POSTS

Tiled Gallery

Jetpack tiled gallery test

This is a test for Jetpack's Tiled Gallery. You can install Jetpack or Slim Jetpack to test it out. This is some text after the Tiled Gallery just to make … Continue reading

Twitter Embeds

Doing what you know

Doing what you "know" locks you in a prison of the past. Uncertainty is the path to an innovative future. — Carl Smith (@carlsmith) October 16, 2012 This post tests … Continue reading

Featured Image (Vertical)

This post should display a featured image, if the theme supports it. Non-square images can provide some unique styling issues. This post tests a vertical featured image

Custom Taxonomies

- `show_in_rest` automatically registers routes in the `wp/v2` namespace.

Custom Endpoints

- WordPress supports completely custom endpoints that aren’t backed by a core data structure.
- Custom URL structure.
- Response and request formats are flexible.
- Don’t necessarily need to stick with REST semantics.

```
function register_rest_route( $namespace, $route, $args ) {
    // ...
}

register_rest_route( 'my-project/v1', 'testimonials', [
    'callback' => function( $request ) {
        // query testimonials
        $query = get_posts( /* ... */ );
        // format testimonials
        $data = array_walk( $query, function( $t ) { } );

        return new WP_REST_Response( $data );
    }
] );
```

```
class My_Project_Testimonials extends WP_REST_Controller {

    protected $namespace = 'my-project/v1';

}
```

```php
function register_routes() {
    register_rest_route( $this->namespace, 'testimonials', [
        'callback' => [ $this, 'get_items' ],
        'permission_callback' => [ $this, 'get_items_permissions_check' ],
        'schema' => [ $this, 'get_public_item_schema' ],
        'args' => $this->get_collection_params(),
    ] );
}
```

```
function get_items( $request ) {
    // query testimonials
    $query = get_posts( /* ... */ );
    // format testimonials
    foreach ( $query as $t )
        $data[] = $this->prepare_response_for_collection(
            $this->prepare_item_for_response( $t, $request );
        );

    return new WP_REST_Response( $data );
}
```

```
function get_items_permission_check() {
    return is_user_logged_in();
}
```

```
function get_item_schema() { }
```

- Is a full JSON Schema.
- Indicates to clients what data is included in the response.
- *Write* endpoints use it to automatically validate and sanitize.

```
function get_collection_params() {
    $p = parent::get_collection_params();

    $p['after'] = [
        'type' => 'string',
        'format' => 'date-time',
        'description' => 'Show testimonials written after this date.',
    ];

    return $p;
}
```

Use Cases

- Proxying access to an API that requires a private API key or other sensitive data.
- Data stored in custom tables.
- Providing a highly customized view for a post type.

Comment Voting

- WordPress provides a comment route.
- Could just use comment metadata.

```json
{
  "id": 5,
  "content": {
    "rendered": "This is my insightful comment!"
  },
  "meta": {
    "votes": 5
  }
}
```

Comment Voting

- WordPress provides a comment route.
- Could use just comment metadata.
  - How would we prevent duplicate votes?
  - Votes isn’t consistent the whole time.
- Could use `register_rest_field` and accept a vote direction.

```
{
  "id": 5,
  "content": {
    "rendered": "This is my insightful comment!"
  },
  "votes": 5,
  "vote": "up"
}
```

Comment Voting

- WordPress provides a comment route.
- Could use just comment metadata.
- Could use `register_rest_field` and accept a vote direction.
    - An update should describe the resulting state, not an action.
- Great use case for custom endpoint.

```
POST /wp-json/reddit-clone/v1/vote/5 HTTP/1.1
{
  "dir": "up"
}
```

```
register_rest_route( 'reddit-clone/v1', 'vote/(?P<id>[\d]+)', [
    'methods' => 'POST',
    'callback' => function ( $request ) {
        // ...
    },
    'permission_callback' => function ( $request ) {
        // ...
    }
] );
```

Callback

```
function ( $request ) {
    $dir     = $request['dir'];
    $comment = $request['id'];
    $uid     = get_current_user_id();

    $votes = get_comment_meta( $comment, '_votes', true ) ?: 1;
    $votes = $dir === 'up' ? $votes + 1 : $votes - 1;
    update_comment_meta( $comment, '_votes', $votes );
    update_user_meta( $uid, "_voted_{$comment}", $dir );

    return new WP_REST_Response( [ 'votes' => $votes ], 204 );
}
```

Permissions

```
function ( $request ) {

    $comment = $request['id'];
    $uid     = get_current_user_id();

    return $uid && ! get_user_meta( $uid, "_voted_{$comment}" . true );
}
```

... Continue reading

WP Daily

March 14, 2013 at 11:23 am / Edit

We are totally going to blog about these tests!

Reply Up 9 Down

Michael Novotny 👤

March 14, 2013 at 11:30 am / Edit

Thanks for all the comments, everyone!

Reply Up 6 Down

Eryn Erickson

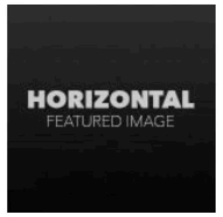
March 14, 2013 at 9:56 am / Edit

Image comment.

Featured Image (Vertical)

This post should display a featured image, if the theme supports it. Non-square images can provide some unique styling issues. This post tests a vertical featured image.

Featured Image (Horizontal)



This post should display a featured image, if the theme supports it. Non-square images can provide some unique styling issues. This post tests a horizontal featured image.

NEXT

SEARCH

Search ... 🔍

Questions?