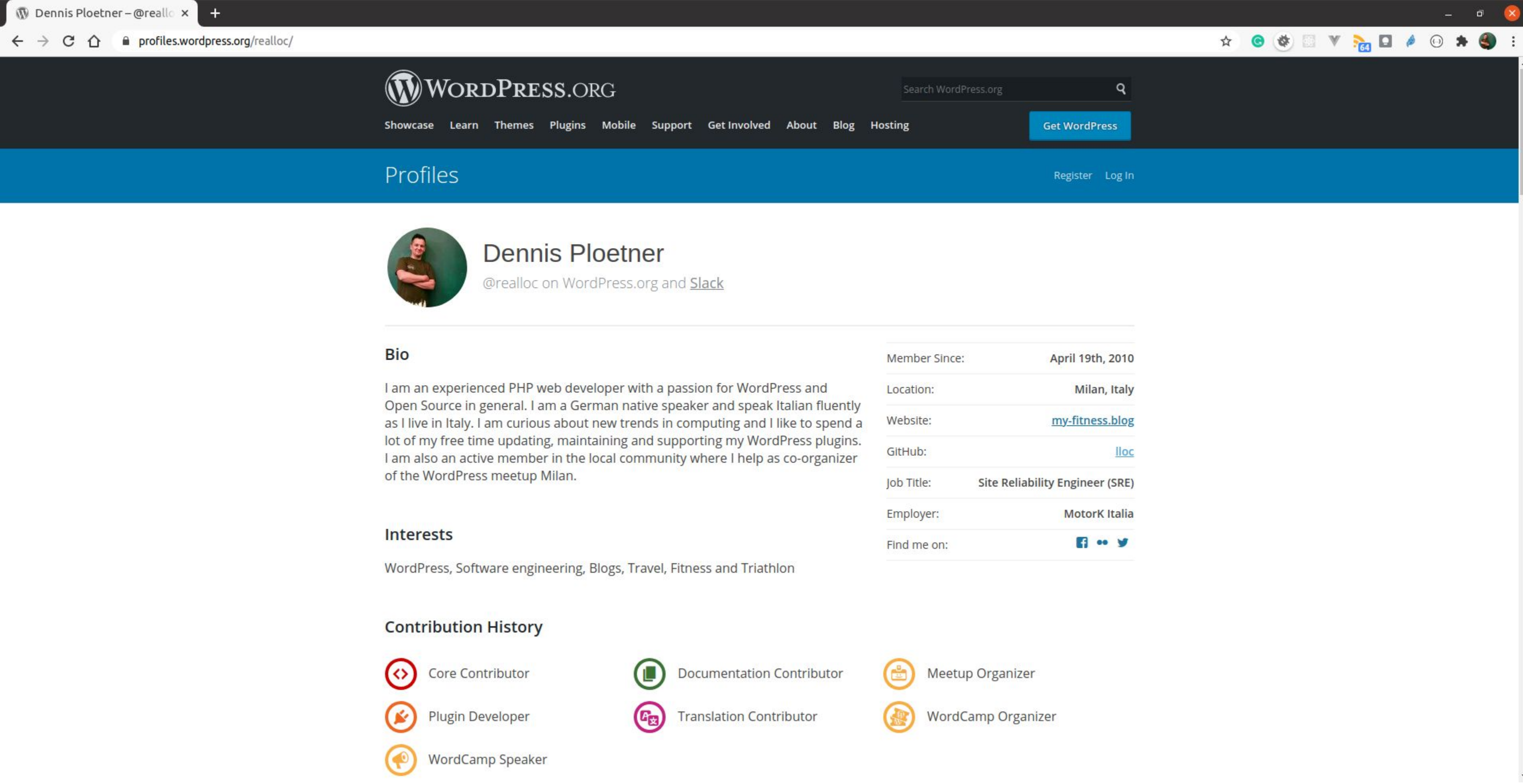
WordPress REST API: ne riparliamo?

Dennis Ploetner @realloc



Profiles

Dennis Ploetner

@realloc on WordPress.org and Slack

Bio

I am an experienced PHP web developer with a passion for WordPress and Open Source in general. I am a German native speaker and speak Italian fluently as I live in Italy. I am curious about new trends in computing and I like to spend a lot of my free time updating, maintaining and supporting my WordPress plugins. I am also an active member in the local community where I help as co-organizer of the WordPress meetup Milan.

Member Since:	April 19th, 2010
Location:	Milan, Italy
Website:	my-fitness.blog
GitHub:	lloc
Job Title:	Site Reliability Engineer (SRE)
Employer:	MotorK Italia
Find me on:	

Interests

WordPress, Software engineering, Blogs, Travel, Fitness and Triathlon

Contribution History

- Core Contributor
- Plugin Developer
- WordCamp Speaker
- Documentation Contributor
- Translation Contributor
- Meetup Organizer
- WordCamp Organizer

Sources

Gimme the codes first!

https://github.com/lloc/simple-ping

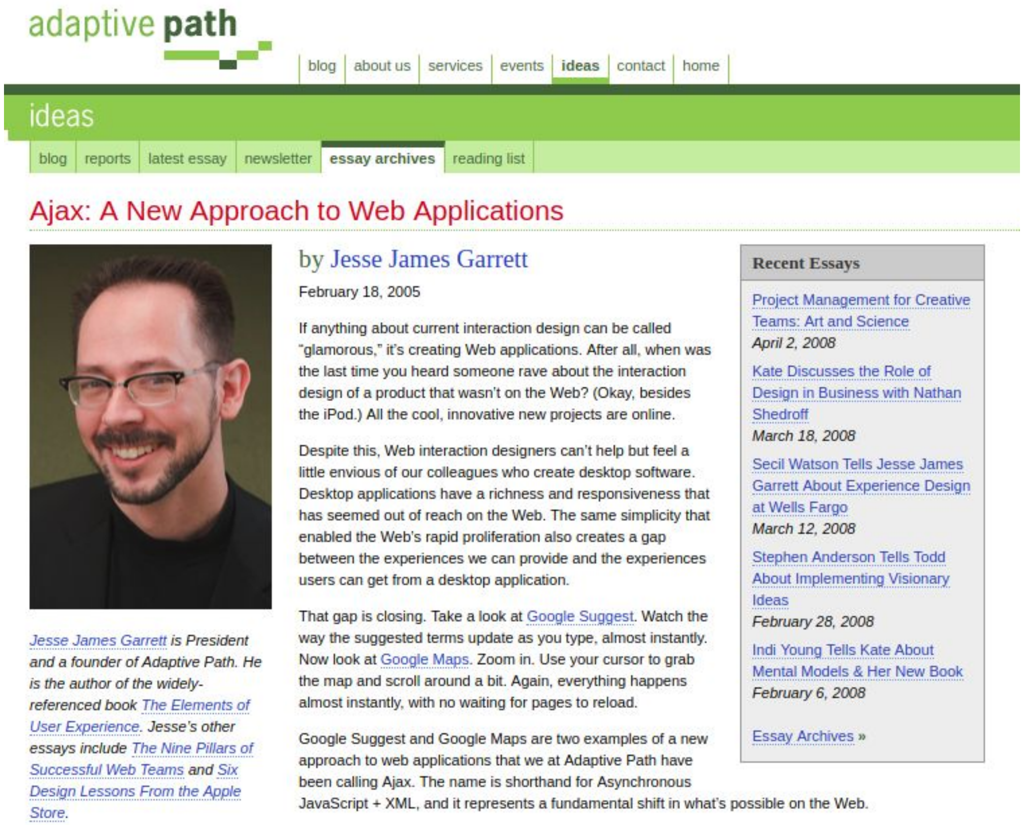
Checkout the branches

- ajax-endpoint
- rest-api
- master (restful)

PHP + some (obsolete 😁) JavaScript code for demonstration.

Asynchronous JavaScript and XML

Brief history...



Ajax: A New Approach to Web Applications

[Jesse James Garrett](#) is President and a founder of Adaptive Path. He is the author of the widely-referenced book The Elements of User Experience. Jesse's other essays include The Nine Pillars of Successful Web Teams and Six Design Lessons From the Apple Store.

by Jesse James Garrett

February 18, 2005

If anything about current interaction design can be called "glamorous," it's creating Web applications. After all, when was the last time you heard someone rave about the interaction design of a product that wasn't on the Web? (Okay, besides the iPod.) All the cool, innovative new projects are online.

Despite this, Web interaction designers can't help but feel a little envious of our colleagues who create desktop software. Desktop applications have a richness and responsiveness that has seemed out of reach on the Web. The same simplicity that enabled the Web's rapid proliferation also creates a gap between the experiences we can provide and the experiences users can get from a desktop application.

That gap is closing. Take a look at Google Suggest. Watch the way the suggested terms update as you type, almost instantly. Now look at Google Maps. Zoom in. Use your cursor to grab the map and scroll around a bit. Again, everything happens almost instantly, with no waiting for pages to reload.

Google Suggest and Google Maps are two examples of a new approach to web applications that we at Adaptive Path have been calling Ajax. The name is shorthand for Asynchronous JavaScript + XML, and it represents a fundamental shift in what's possible on the Web.

Recent Essays

Project Management for Creative Teams: Art and Science
April 2, 2008

Kate Discusses the Role of Design in Business with Nathan Shedroff
March 18, 2008

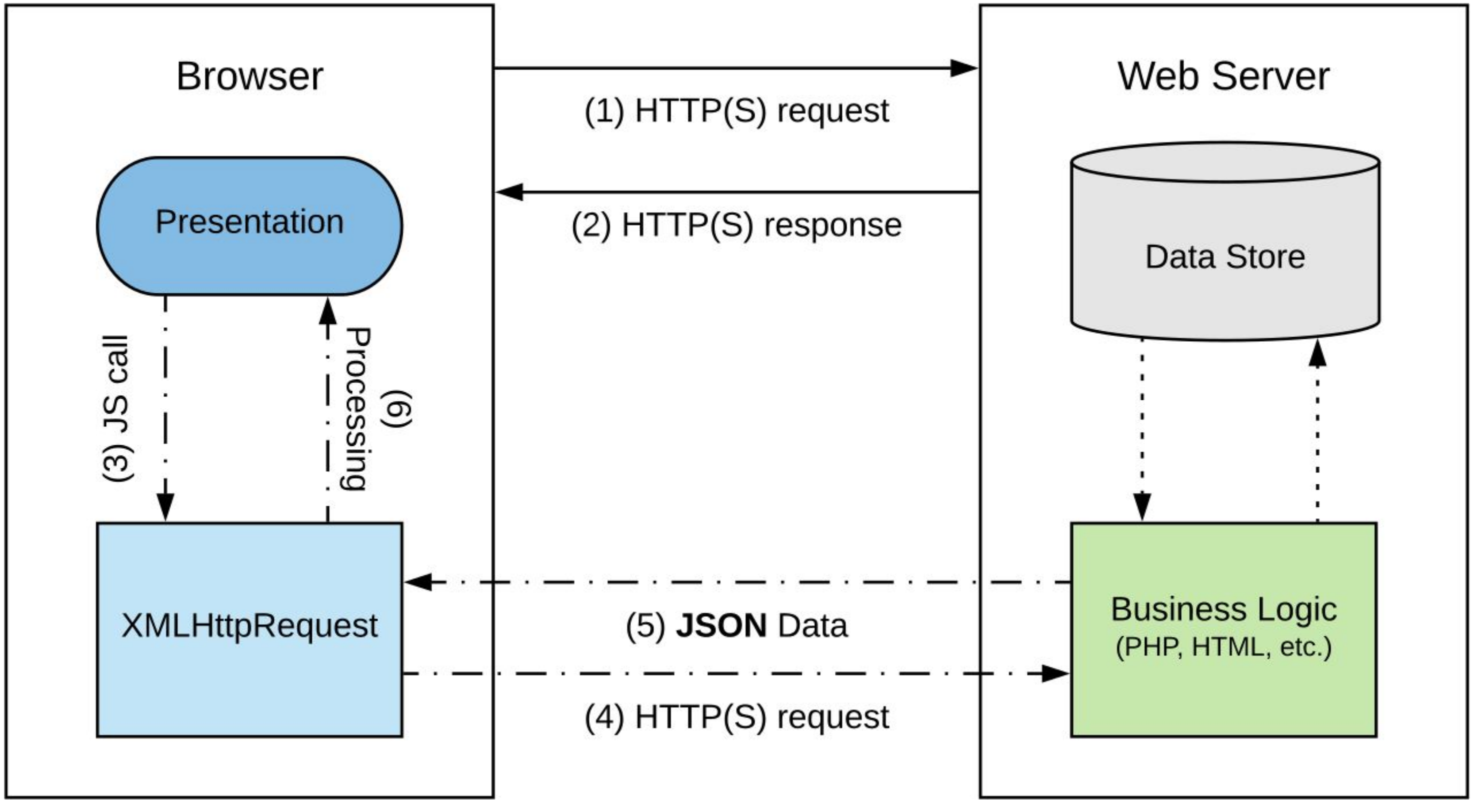
Secil Watson Tells Jesse James Garrett About Experience Design at Wells Fargo
March 12, 2008

Stephen Anderson Tells Todd About Implementing Visionary Ideas
February 28, 2008

Indi Young Tells Kate About Mental Models & Her New Book
February 6, 2008

Essay Archives »

AJAX flowchart



Classic AJAX implementation

Enqueue script and define ajaxurl in your theme’s function.php or a custom plugin

```
add_action( 'wp_enqueue_scripts', function () {
  $handle = 'simple-ping';
  $src    = plugins_url( 'simple-ping.js', __FILE__ );

  wp_enqueue_script( $handle, $src, [ 'jquery' ] );
  wp_localize_script( $handle, 'LLOC', [
     'ajaxurl' => admin_url( 'admin-ajax.php' )
  ] );
} );
```

Classic AJAX implementation

jQuery event listener in a file named simple-ping.js

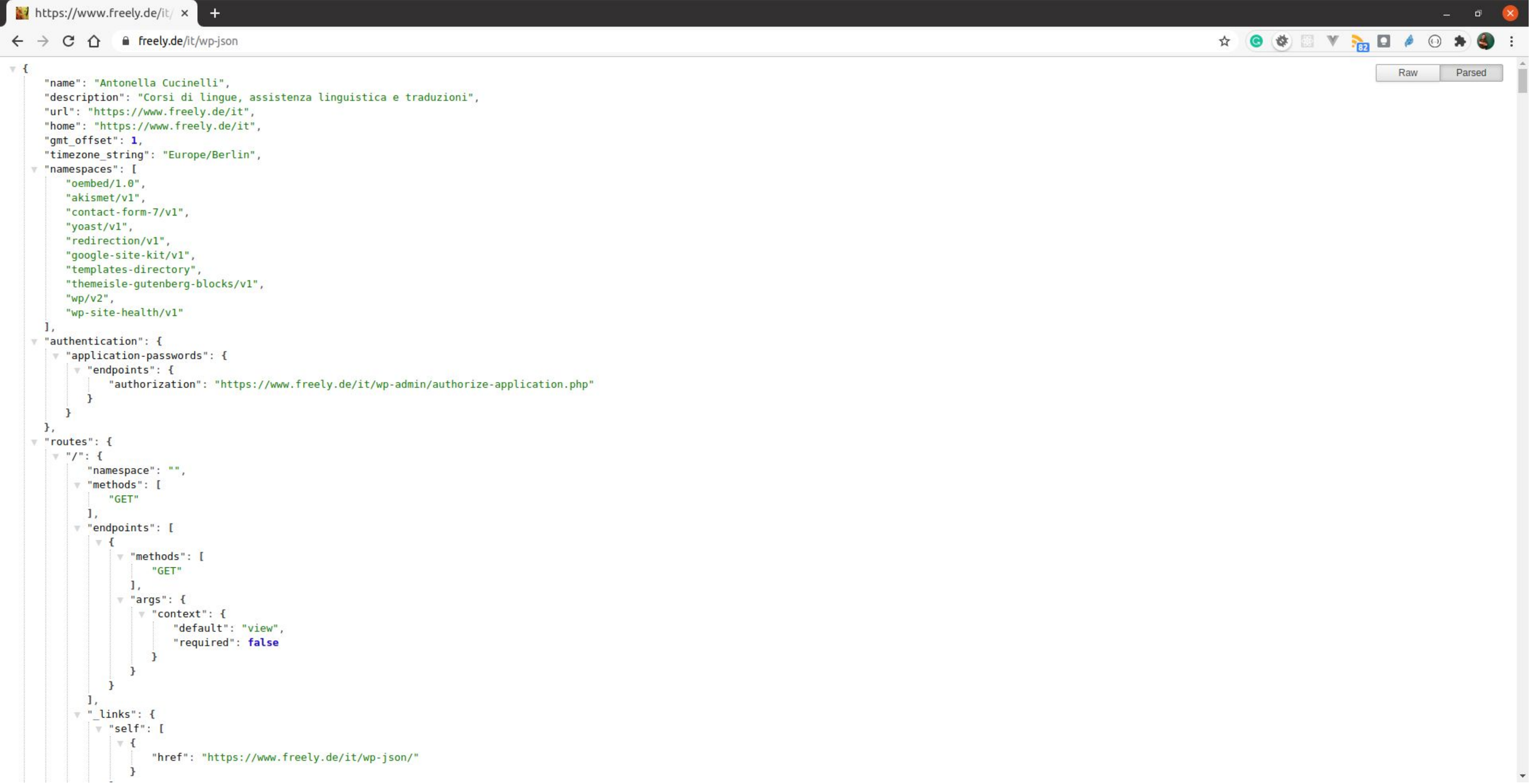
```
jQuery(document).ready(function($) {
 'use strict';

 $('body').click(function() {
   var settings = {
     url: LLOC.ajaxurl,
     data: {action: 'simple_ping', msg: 'ping'}
   };
   $.ajax(settings).done(function(response) {
     console.debug(response.data);
   });
 });
});
```

Classic AJAX implementation

PHP callback definition in your theme’s function.php or a custom plugin

```
add action( 'wp_ajax_simple_ping', 'simple_ping' );
add_action( 'wp_ajax_nopriv_simple_ping', 'simple_ping' );

function simple ping(): void {
  if ( 'ping' == $ REQUEST['msg'] ) {
     wp_send_json_success( 'pong' );
  }

  wp_send_json_error( 'Huhh?' );
}
```

Application Programming Interface



```
{
    "name": "Antonella Cucinelli",
    "description": "Corsi di lingue, assistenza linguistica e traduzioni",
    "url": "https://www.freely.de/it",
    "home": "https://www.freely.de/it",
    "gmt_offset": 1,
    "timezone_string": "Europe/Berlin",
    "namespaces": [
        "oembed/1.0",
        "akismet/v1",
        "contact-form-7/v1",
        "yoast/v1",
        "redirection/v1",
        "google-site-kit/v1",
        "templates-directory",
        "themeisle-gutenberg-blocks/v1",
        "wp/v2",
        "wp-site-health/v1"
    ],
    "authentication": {
        "application-passwords": {
            "endpoints": {
                "authorization": "https://www.freely.de/it/wp-admin/authorize-application.php"
            }
        }
    },
    "routes": {
        "/": {
            "namespace": "",
            "methods": [
                "GET"
            ],
            "endpoints": [
                {
                    "methods": [
                        "GET"
                    ],
                    "args": {
                        "context": {
                            "default": "view",
                            "required": false
                        }
                    }
                }
            ],
            "_links": {
                "self": [
                    {
                        "href": "https://www.freely.de/it/wp-json/"
                    }
```

Example of a root endpoint of the WordPress REST API using the JSON Formatter browser extension.

WordPress REST API implementation

Enqueue script and define the url of the API endpoint (function.php or custom plugin)

```
add action( 'wp_enqueue_scripts', function () {
  $handle   = 'simple-ping';
  $src      = plugins url( 'simple-ping.js', __FILE__ );
  $endpoint = '/wp-json/lloc/v1/';

  wp enqueue script( $handle, $src, [ 'jquery' ] );
  wp localize script( $handle, 'LLOC', [
     'apiurl' => site_url( $endpoint )
  ] );
} );
```

WordPress REST API implementation

Small change in jQuery event listener (still in simple-ping.js)

```
jQuery(document).ready(function($) {
 'use strict';

 $('body').click(function() {
   var settings = {
     url: LLOC.apiurl + 'ping',
     data: {msg: 'ping'}
   };
   $.ajax(settings).done(function(response) {
     console.debug(response);
   });
 });
});
```

WordPress REST API implementation

PHP callback definition in your theme’s function.php or a custom plugin

```php
add_action( 'rest_api_init', function () {
  register_rest_route( 'lloc/v1', 'ping', [
        'methods'  => 'GET',
        'callback' => 'simple_ping',
        'args'     => [ 'msg' => [ 'required' => true ] ],
     ]
  );
} );

function simple_ping( \WP_REST_Request $request ): \WP_REST_Response {
  $data = 'ping' == $request->get_param( 'msg' ) ? 'pong' : 'Huhh?';

  return rest_ensure_response( $data );
}
```

REpresentational State Transfer

WordPress RESTful implementation

PHP register routes (let’s use a custom plugin)

```
add_action( 'rest_api_init', function () {
  register_rest_route( 'lloc/v1', 'ping',
     [
        'methods'  => \WP_REST_Server::READABLE,
        'callback' => 'read_ping',
        'args'     => [ 'msg' => [ 'required' => true ] ],
     ],
     [
        'methods'  => \WP_REST_Server::EDITABLE,
        'callback' => 'edit_ping',
        'args'     => [ 'msg' => [ 'required' => true ] ],
     ]
  );
} );
```

WordPress RESTful implementation

jQuery event listener … some further changes in simple-ping.js

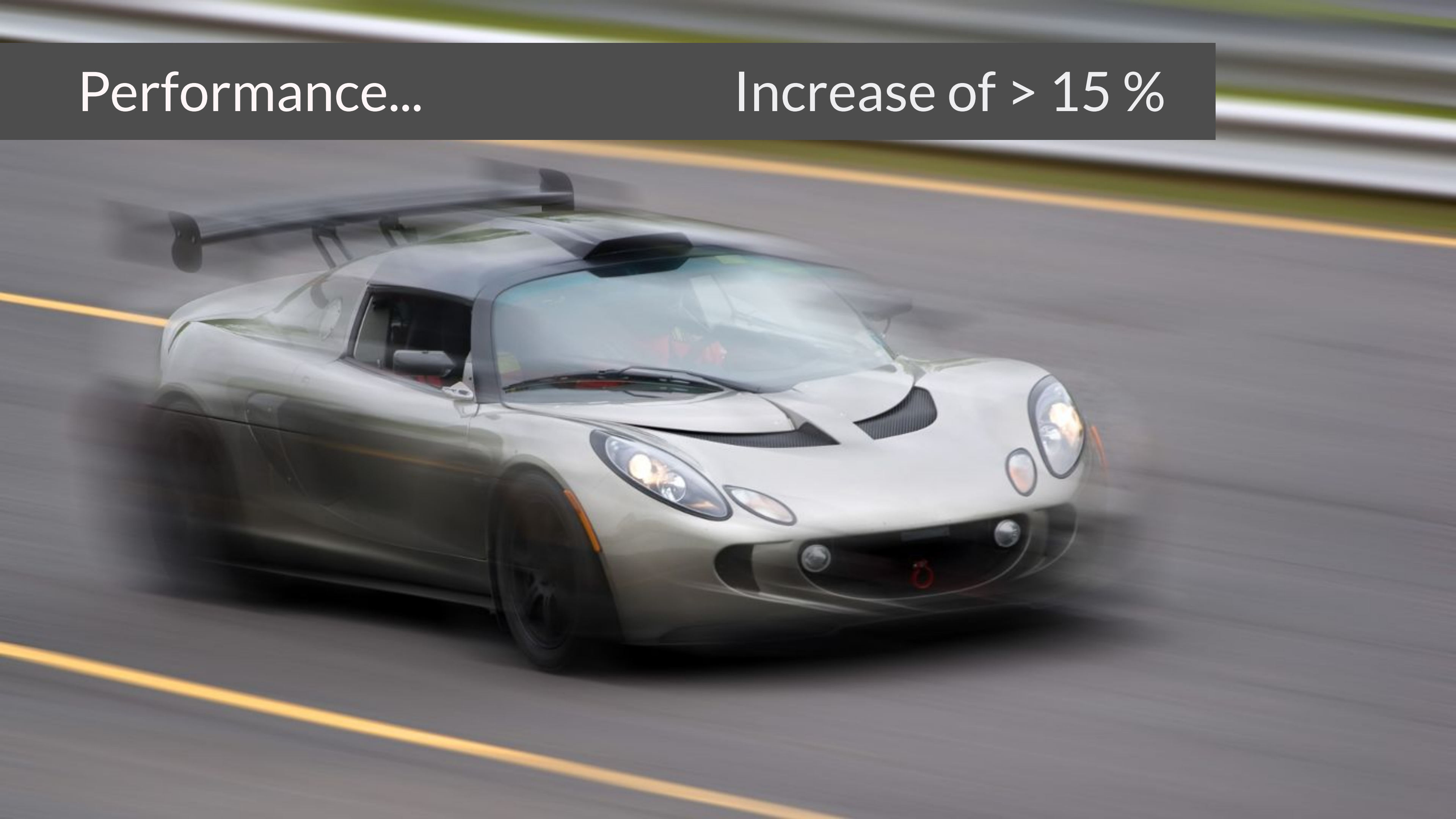
```
jQuery(document).ready(function($) {
 'use strict';

 $('body').click(function() {
   var settings = {
     url: LLOC.apiurl + 'ping',
     type: 'POST', // 'GET', 'DELETE', ...
     data: {msg: 'ping'}
   };
   $.ajax(settings).done(function(response) {
     console.debug(response);
   });
 });
});
```

WordPress REST API

Why should I use the WordPress REST API?

- easy to use
- easy to structure complete applications
- strong replacement for admin-ajax.php
- your imagination is the limit

Performance... Increase of > 15 %



Security

Any risks?

Security Giveaways

What should I know about the security of the WordPress REST API?

- You'd like to deactivate it? You need it probably activated.
- Don't expect security by obscurity working.
- Don't expose confidential data without protection.
- Use basic auth or better OATH, application passwords or JWT (Handbook).

Can I just... Ignore all this?

Thank you for your attention!

Dennis Ploetner @realloc