ezjscore JSON endpoints in PHP

Writing ezjscore server functions in PHP - the [ezjscServer_<name>] block, the static method that receives the arguments array, the answer envelope, permissions with Functions[] and RequirePolicy, POST writes with the...

Source file: `doc/eng-us/help/php-developers/ezjscore-json-endpoints-in-php.md` Last changed 2026-11-04
https://doc.exponential.earth/learninminutes/exphelp/viewer/help/php-developers/ezjscore-json-endpoints-in-php.md

This page is for PHP developers who give their own JavaScript data or actions over HTTP: autocomplete, counters, small saves. `ezjscore` (`extension/ezjscore`) calls a registered static PHP method and answers in JSON, XML or text. The JavaScript side and a first example are on JSON endpoints with ezjscore.

Contents

1. How a call reaches your method
2. Registering the class
3. The method
4. Permissions
5. Writes
6. Several calls in one request
7. expservices: the structured form
8. Testing

How a call reaches your method

```
<root>/ezjscore/call/<name>::<function>[::<arg>[::<arg>...]]?ContentType=json
```

The router `ezjscServerRouter` (`extension/ezjscore/classes/ezjscserverrouter.php`) refuses any name without a `[ezjscServer_<name>]` block, calls `<class>::<function>( array $args )` with the `::`-separated arguments as strings, and wraps the return value:

JSON

```
{"error_text": "", "content": <what your method returned>}
```

The format follows the `Accept` header or `?ContentType=json|xml|text`. The kernel ships a test call: `<root>/ezjscore/call/ezjsc::time`.

Registering the class

`extension/acmereport/settings/ezjscore.ini.append.php`:

INI

```
<?php /* #?ini charset="utf-8"?
[ezjscServer_acmereport]
Class=acmeReportServerFunctions
Functions[]=acmereport
RequirePolicy=enabled
*/ ?>
```

Key	Meaning
`Class`	the class; without it the block name is used
`File`	optional; without it the autoload map finds the class
`TemplateFunction`	answer with a template instead of a PHP method
`Functions[]`	the `[ezjscServer] FunctionList[]` entries the user needs (policy `ezjscore/call`, limitation `FunctionList`)
`PermissionPrFunction`	check one `FunctionList` entry per function (`<name>_<function>`)
`RequirePolicy`	not set: refused calls answer with an error text; `enabled`: a request whose calls are all refused answers 403; `disabled`: not checked

The comments of `[ezjscServer]` in `extension/ezjscore/settings/ezjscore.ini` describe each key. Add your entry to `[ezjscServer] FunctionList[]` when you use `Functions[]`.

The method

PHP

```
<?php
class acmeReportServerFunctions
{
    /**
     * acmereport::count::<status>
     *
     * @param array $args the ::-separated arguments, as strings
     * @return array
     */
    public static function count( $args )
    {
        $access = eZUser::currentUser()->hasAccessTo( 'acmereport', 'read' );
        if ( $access['accessWord'] === 'no' )
            throw new Exception( 'Access denied' );

        $status = isset( $args[0] ) ? (string)$args[0] : '';
        if ( !in_array( $status, array( 'open', 'closed' ), true ) )
            throw new Exception( 'Unknown status' );

        return array( 'count' => (int)eZPersistentObject::count( acmeReportItem::definition(),
                                                                 array( 'status' => $status ) ) )
;
    }
}
```

- Every argument is a string from the URL: cast it and check it against an allow list.
- Return data (arrays, numbers, strings), not HTML. If you must return markup, escape it.
- An exception becomes the `error_text` of the answer; keep its message free of internals.

Permissions

`ezjscore/call` does not check a policy for your function by itself. Use both layers:

1. `Functions[]` with `RequirePolicy=enabled` in the block, so a user without the `ezjscore/call` policy for your entry is refused before your code runs.
2. A check inside the method for the module policy that the data belongs to (`hasAccessTo()`, `canRead()` on nodes).

Writes

A method that changes data must refuse anything but POST and rely on the form token:

PHP

```php
if ( $_SERVER['REQUEST_METHOD'] !== 'POST' )
    throw new Exception( 'POST required' );
$http = eZHTTPTool::instance();
$title = trim( (string)$http->postVariable( 'title', '' ) );
```

For signed-in users, `ezformtoken` checks the token on every POST before the router runs: the JavaScript sends it as the field `ezxform_token` or the header `X-CSRF-Token` (CSRF protection and ezformtoken).

Several calls in one request

A POST of `ezjscServer_function_arguments` with several calls joined by `@SEPARATOR$` runs them in one request. Each call is checked like a single one.

expservices: the structured form

For services that scripts, apps and other front ends call (not only your own pages), the `expservices` extension builds on the same router with one class per domain extending `expServiceBase`, a declared `public static $services` list, discovery (`expservices::catalog`), a fixed envelope (`{"ok": true, "data": ...}`), paging, error codes, API tokens and an audit event for every write (Backend services over ezjscore):

PHP

```php
class expFooServices extends expServiceBase
{
    public static $services = array(
        'bar' => array( 'summary' => '...', 'access' => array( 'content', 'read' ),
                        'write' => false, 'args' => array( 'node_id' => 'int' ), 'returns' => '..
.' ) );

    public static function bar( $args )
    {
        self::guard( __FUNCTION__ );
        $node = self::node( self::arg( $args, 0, 'int' ) );
        return self::ok( array( 'name' => $node->attribute( 'name' ) ) );
    }
}
```

Declare `$services` as the property's value in the class body, never by an assignment after the class: a persistent Velocity worker puts statics back to their declared values between requests,

and an assignment outside the class does not run again (Static state traps).

Testing

Shell

```
curl -s 'https://www.example.com/ezjscore/call/acmereport::count::open?ContentType=json'
php vendor/bin/phpunit tests/tests/extension/expservices/
```

Test the method itself in a DB-less test by calling it with an arguments array.

Related pages

- ezjscore server functions reference
- Ajax with ezjscore server functions
- REST and remote services from PHP
- Free documentation: this page is free to copy, translate and improve under the GNU FDL 1.3.

> **Become an Exponential Partner**
>
> Partners share their information, provide funding, contribute back and develop the products with us, and that is how Exponential grows. A partner website and dashboard are planned for mid-2027: please become a partner today to speed up our growth rate and land new features faster.
>
> Partners: start here | How to apply

Exponential Copyright © 1998-2026 Exponential Foundation and 7x
Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".
This documentation is free documentation, licensed under the GNU Free Documentation License 1.3 or later:
https://www.gnu.org/licenses/fdl-1.3.html.