With RAD: a module with views

The Module wizard writes a module with two views and policies; read the output, correct the template folder by hand, and know when to use Module over existing tables.

Source file:
doc/eng-us/help/extension-development/part-3-extension-developer/tutorials/rad-module.md Last changed 2026-11-04
https://doc.exponential.earth/learninminutes/exphelp/viewer/help/extension-development/part-3-extension-developer/tutorials/rad-module.md

Tutorial 4. Two wizards write modules. The **Module wizard** (`/setup/modulewizard`) writes a module with views and policy functions; it is also the tool for "View for an existing module" and "Policy function and limitation". **Module over existing tables** (`/setup/moduleextension`) writes a module, persistent object classes and pages for tables that already exist. This tutorial uses the first. The output shown was produced by the wizard's generator (`expModuleWizard::files()`) with the same input; every PHP file passed `php -l`.

1. The form

Field	Value
Extension name	`mysite_hello`
Module	`hello`
Title	`Hello module`
Views	`world: read Name?` and, on the next line, `admin: administrate`
Policies	`read` and `administrate`

The views field takes one view per line: a name, a colon, then words. The wizard's rule (`expModuleWizard::viewList()`): "A word in capitals is an ordered parameter, a word in lower case is a policy", and a trailing `?` marks a parameter that may be left out.

2. The files

```
.gitignore
LICENSE
README.md
composer.json
design/mysite_hello/templates/hello/admin.tpl
design/mysite_hello/templates/hello/world.tpl
doc/examples.php
extension.xml
ezinfo.php
modules/hello/admin.php
modules/hello/module.php
modules/hello/world.php
settings/design.ini.append.php
settings/menu.ini.append.php
settings/module.ini.append.php
```

The generated `modules/hello/module.php`:

PHP

```php
<?php
$Module = array( 'name' => 'Hello module' );

$ViewList = array();

$ViewList['world'] = array(
    'script' => 'world.php',
    'functions' => array( 'read' ),
    'default_navigation_part' => 'ezsetupnavigationpart',
    'unordered_params' => array( 'name' => 'Name' ),
);

$ViewList['admin'] = array(
    'script' => 'admin.php',
    'functions' => array( 'administrate' ),
    'default_navigation_part' => 'ezsetupnavigationpart',
);

$FunctionList = array();

$FunctionList['read'] = array();

$FunctionList['administrate'] = array();
```

Every view requires its policy, as it should. The optional parameter `Name?` became an unordered parameter, so the address is `/hello/world/(name)/Ada` (the generated README's table says so).

The view script starts with the right warning and a deliberately safe default:

PHP

```php
// Everything in $Params came out of the address, which means it came from
// whoever typed it. The declared type is nothing: cast, and check.
$name = isset( $Params['Name'] )
          ? (int) $Params['Name'] : 0;
```

The `(int)` assumes an ID. For a name, replace it with a string check, as in View parameters.

3. One correction before you use it

The wizard writes the templates to `design/mysite_hello/templates/`, a design named after the extension. The design chain only searches designs that a siteaccess uses (`SiteDesign`, `AdditionalSiteDesignList[]`, `StandardDesign`), so those templates are not found and the views answer with an empty page. Move them to `standard` (or to your admin design):

Shell

```shell
mv extension/mysite_hello/design/mysite_hello extension/mysite_hello/design/standard
php bin/php/ezcache.php --clear-id=template-override,design_base
```

The generated templates use the classic admin's box markup; for admin4 pages, rewrite them with the shared components (Admin pages in the admin4 style).

4. The menu

`settings/menu.ini.append.php` adds the module to a navigation part's menu (`[NavigationPartMenu_ezsetupnavigationpart]` with `Menu[]=hello`, `MenuTitle[hello]` and `MenuURL[hello]=/hello/world`). Check the result in your administration design and adjust; the admin4 left menu groups links as described in Admin pages in the admin4 style.

5. Switch it on, grant, open

Shell

```
./console exp:ini add 'site.ini/ExtensionSettings/ActiveExtensions[]' mysite_hello global
php bin/php/ezpgenerateautoloads.php -e
php bin/php/ezcache.php --clear-all
```

Grant `hello/read` to a role (**Users > Roles and policies**), then open `/hello/world/(name)/Ada`.

Module over existing tables

`/setup/moduleextension` is for data that already has tables. It reads their structure and writes, among others, one persistent object class per table (`classes/<class>.php`), a connection and a registry class, the module with its `function_definition.php` and a function collection, a remove view and template, a settings file named after the extension, a menu entry, and `share/db_schema.dba`. Use it when you build administration pages for data another system writes; for new data, design the table and class yourself (Persistent objects and your own tables).

Related pages

- Previous: With RAD: a datatype
- Next: With RAD: a workflow event
- Modules and views, By hand: a module with a view
- 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.