Task: build a small CRUD module step by step

A hands-on task for framework developers - an extension with its own table, a persistent object model, a list view and an edit form with actions, validation, the CSRF token, a policy function, templates and a deploy…

Source file: `doc/eng-us/help/web-framework-developers/your-first-module-step-by-step.md` Last changed 2026-11-04
https://doc.exponential.earth/learninminutes/exphelp/viewer/help/web-framework-developers/your-first-module-step-by-step.md

This page is for developers who want to see every piece once, the way `make:crud`, `rails generate scaffold` or a Django admin model would produce them. You build an extension `expdemo` with a table of items, a list at `/expdemo/list` and an edit form at `/expdemo/edit/<id>`. **Setup > RAD** can write much of this boilerplate for you afterwards; doing it by hand first shows what the wizards write (RAD tools).

Time: about an hour on a development installation (`./console exp:install --apply` gives one on SQLite in minutes: Installing in one command).

Contents

1. The files
2. Register the module
3. The table and the model
4. The list view
5. The edit view
6. The templates
7. Activate, deploy and grant
8. Check it

The files

```
extension/expdemo/
├── classes/expdemoitem.php
├── modules/expdemo/module.php
├── modules/expdemo/list.php
├── modules/expdemo/edit.php
├── design/standard/templates/expdemo/list.tpl
├── design/standard/templates/expdemo/edit.tpl
├── settings/module.ini.append.php
├── settings/design.ini.append.php
└── share/db_schema.dba
```

Register the module

INI

```
# settings/module.ini.append.php
<?php /* #?ini charset="utf-8"?
[ModuleSettings]
ExtensionRepositories[]=expdemo
ModuleList[]=expdemo
*/ ?>
```

INI

```
# settings/design.ini.append.php
<?php /* #?ini charset="utf-8"?
[ExtensionSettings]
DesignExtensions[]=expdemo
*/ ?>
```

PHP

```
<?php
// modules/expdemo/module.php
$Module = array( 'name' => 'Demo' );

$ViewList = array();
$ViewList['list'] = array(
    'script' => 'list.php',
    'functions' => array( 'read' ),
    'unordered_params' => array( 'offset' => 'Offset' ) );
$ViewList['edit'] = array(
    'script' => 'edit.php',
    'functions' => array( 'manage' ),
    'params' => array( 'ItemID' ),
    'single_post_actions' => array( 'StoreButton' => 'Store', 'CancelButton' => 'Cancel' ),
    'post_action_parameters' => array( 'Store' => array( 'Title' => 'ItemTitle' ) ) );

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

The table and the model

Create the table on your development database, for example on SQLite:

SQL

```
CREATE TABLE expdemo_item ( id INTEGER PRIMARY KEY AUTOINCREMENT, user_id INTEGER NOT NULL DEFAUL
T 0,
                            title VARCHAR(200) NOT NULL DEFAULT '', created INTEGER NOT NULL DEFA
ULT 0 );
```

Then dump it into a database-neutral schema file for every other installation (Migrations and seed data):

Shell

```
php bin/php/ezsqldumpschema.php --type=sqlite --table-include=expdemo_item --output-array \
  var/storage/sqlite3/exponential.db extension/expdemo/share/db_schema.dba --allow-root-user
```

The model is the `expDemoItem` class of ORM: eZPersistentObject, saved as

`classes/expdemoitem.php`, plus one helper:

PHP

```
    public static function fetch( $id )
    {
        return eZPersistentObject::fetchObject( self::definition(), null, array( 'id' => (int)$id
 ) );
    }
```

The list view

PHP

```
<?php
// modules/expdemo/list.php
$offset = isset( $Params['Offset'] ) ? max( 0, (int)$Params['Offset'] ) : 0;
$items = eZPersistentObject::fetchObjectList( expDemoItem::definition(), null, null,
                                              array( 'created' => 'desc' ),
                                              array( 'offset' => $offset, 'length' => 20 ) );

$tpl = eZTemplate::factory();
$tpl->setVariable( 'items', $items ?: array() );
$tpl->setVariable( 'offset', $offset );

$Result = array();
$Result['content'] = $tpl->fetch( 'design:expdemo/list.tpl' );
$Result['path'] = array( array( 'text' => 'Items', 'url' => false ) );
```

The edit view

PHP

```
<?php
// modules/expdemo/edit.php
$Module = $Params['Module'];
$id = (int)$Params['ItemID'];
$item = $id > 0 ? expDemoItem::fetch( $id ) : null;
if ( $id > 0 && !$item )
    return $Module->handleError( eZError::KERNEL_NOT_FOUND, 'kernel' );

if ( $Module->isCurrentAction( 'Cancel' ) )
    return $Module->redirectTo( '/expdemo/list' );

$errors = array();
$title = $item ? $item->attribute( 'title' ) : '';
if ( $Module->isCurrentAction( 'Store' ) )
{
    $title = $Module->hasActionParameter( 'Title' ) ? trim( (string)$Module->actionParameter( 'Ti
tle' ) ) : '';
    if ( $title === '' || mb_strlen( $title ) > 200 )
        $errors['title'] = ezpI18n::tr( 'extension/expdemo', 'Enter a title of at most 200 charac
ters.' );

    if ( $errors === array() )
    {
        if ( !$item )
            $item = new expDemoItem( array( 'user_id' => eZUser::currentUserID(), 'created' => ti
me() ) );
        $item->setAttribute( 'title', $title );
        $item->store();
```

```
        return $Module->redirectTo( '/expdemo/list' );   // POST, redirect, GET
    }
}

$tpl = eZTemplate::factory();
$tpl->setVariable( 'item', $item );
$tpl->setVariable( 'title', $title );
$tpl->setVariable( 'errors', $errors );

$Result = array();
$Result['content'] = $tpl->fetch( 'design:expdemo/edit.tpl' );
$Result['path'] = array( array( 'text' => 'Items', 'url' => 'expdemo/list' ),
                         array( 'text' => 'Edit', 'url' => false ) );
```

`/expdemo/edit/0` creates a new item.

The templates

```
{* design/standard/templates/expdemo/list.tpl *}
<h1>{'Items'|i18n( 'extension/expdemo' )}</h1>
<p><a href={'expdemo/edit/0'|ezurl}>{'New item'|i18n( 'extension/expdemo' )}</a></p>
<ul>
{foreach $items as $item}
  <li><a href={concat( 'expdemo/edit/', $item.id )|ezurl}>{$item.title|wash}</a></li>
{/foreach}
</ul>
```

```
{* design/standard/templates/expdemo/edit.tpl *}
<form method="post" action={concat( 'expdemo/edit/', first_set( $item.id, 0 ) )|ezurl}>
  <label for="expdemo-title">{'Title'|i18n( 'extension/expdemo' )}</label>
  <input id="expdemo-title" type="text" name="ItemTitle" value="{$title|wash}" maxlength="200" />
  {if is_set( $errors.title )}<p class="error">{$errors.title|wash}</p>{/if}
  <button type="submit" name="StoreButton">{'Save'|i18n( 'extension/expdemo' )}</button>
  <button type="submit" name="CancelButton">{'Cancel'|i18n( 'extension/expdemo' )}</button>
</form>
```

No CSRF code: for signed-in users `ezformtoken` adds the token to the form and checks it on the POST (Forms and CSRF). Every printed value goes through `|wash`.

Activate, deploy and grant

INI

```
# settings/override/site.ini.append.php
[ExtensionSettings]
ActiveExtensions[]=expdemo
```

Shell

```
php bin/php/ezpgenerateautoloads.php -e
./console exp:velocity deploy --allow-root-user     # or clear the ini, design_base and template-override caches
```

In **Users > Roles**, give the editors' role the policies `expdemo / read` and `expdemo / manage`.

Check it

1. Signed in with the role: `/expdemo/list` shows an empty list; **New item** opens the form.
2. Save an empty title: the error appears next to the field and the form keeps what you typed.
3. Save a title: you are redirected to the list, which shows it.
4. Signed in without `expdemo` / `manage`: `/expdemo/edit/0` answers "access denied". Logged out: the login page.
5. `/expdemo/edit/abc` is read as `0` through the cast and shows the form for a new item, never an error.

Next steps: a fetch function so other templates can list items (Fetch functions), the class form of the views (Controllers), an install command for the table, and tests for a validator class (Testing).

Related pages

- Your first extension
- Custom modules and views
- Extension development, part 3
- 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.