ezuser datatype (User account)

The ezuser datatype (User account): what it stores and how templates use it.

Source file: doc/eng-us/examples/datatypes/ezuser.md Last changed 2026-11-04
https://doc.exponential.earth/learninminutes/exphelp/viewer/examples/datatypes/ezuser.md

The ezuser datatype (User account)

Class `eZUserType` in `kernel/classes/datatypes/ezuser/ezusertype.php`; identifier `ezuser`, listed in `settings/content.ini [DataTypeSettings] AvailableDataTypes[]`.

What it stores

The columns of the attribute tables its class reads or writes: `data_text` (`data_text`, `data_int`, `data_float` and `sort_key_*` are the object attribute's; `data_text1` ... `data_int4` the class attribute's settings). Found by reading the class: a heuristic. It is indexed by the search.

In templates

`my_attribute` stands for the identifier of an attribute of this type in your class:

Template

```
{def $attribute = $node.data_map.my_attribute}
{if $attribute.has_content}
    {attribute_view_gui attribute=$attribute}
{/if}
{* the value as the datatype returns it (objectAttributeContent()) *}
{$attribute.content|attribute( show, 1 )}
```

The shipped template that shows it (`attribute_view_gui`), `design/standard/templates/content/datatype/view/ezuser.tpl` (first 25 lines):

Template

```
{* DO NOT EDIT THIS FILE! Use an override template instead. *}
<div class="block">

<div class="element">
<label>{'User ID'|i18n( 'design/standard/content/datatype' )}:</label>
{$attribute.content.contentobject_id}
</div>

<div class="element">
<label>{'Username'|i18n( 'design/standard/content/datatype' )}:</label>
{$attribute.content.login|wash( xhtml )}
</div>

<div class="element">
<label>{'Email'|i18n( 'design/standard/content/datatype' )}:</label>
<a href="mailto:{$attribute.content.email|wash}">{$attribute.content.email|wash}</a>
</div>

<div class="element">
<label>{'Account status'|i18n( 'design/admin/content/datatype/ezuser' )}:</label>
    {if $attribute.content.is_enabled}
```

```
    <span class="userstatus-enabled">{'enabled'|i18n( 'design/standard/content/datatype' )}</span
>
    {else}
    <span class="userstatus-disabled"> {'disabled'|i18n( 'design/standard/content/datatype' )}</s
pan>
    {/if}
```

The shipped template that edits it (`attribute_edit_gui`), `design/standard/templates/content/datatype/edit/ezuser.tpl` (first 25 lines):

Template

```
{* DO NOT EDIT THIS FILE! Use an override template instead. *}
{default attribute_base=ContentObjectAttribute}

{* User ID. *}
<div class="block">
    <label>{'User ID'|i18n( 'design/standard/content/datatype' )}:</label>
    <p>{$attribute.content.contentobject_id}</p>
</div>

{if ne( $attribute_base, 'ContentObjectAttribute' )}
    {def $id_base = concat( 'ezcoa-', $attribute_base, '-', $attribute.contentclassattribute_id,
'_', $attribute.contentclass_attribute_identifier )}
{else}
    {def $id_base = concat( 'ezcoa-', $attribute.contentclassattribute_id, '_', $attribute.conten
tclass_attribute_identifier )}
{/if}

<div class="block">

{* Username. *}
<div class="element">
    <label for="{$id_base}_login">{'Username'|i18n( 'design/standard/content/datatype' )}:</label
>
    {if $attribute.content.has_stored_login}
        <input id="{$id_base}_login" type="text" name="{$attribute_base}_data_user_login_{$attrib
ute.id}_stored_login" size="16" value="{$attribute.content.login|wash()}" disabled="disabled" />
        <input id="{$id_base}_login_hidden" type="hidden" name="{$attribute_base}_data_user_login
_{$attribute.id}" value="{$attribute.content.login|wash()}" />
    {else}
        <input id="{$id_base}_login" class="ezcc-{$attribute.object.content_class.identifier} ezc
ca-{$attribute.object.content_class.identifier}_{$attribute.contentclass_attribute_identifier}" t
ype="text" name="{$attribute_base}_data_user_login_{$attribute.id}" size="16" value="{$attribute.
content.login|wash()}" />
```

Override them for one class attribute in an `override.ini.append.php` with `Source=content/datatype/view/ezuser.tpl` and a match on `attribute_identifier` (see A template override).

Related

- All datatypes
- A custom datatype
- A content class

Generated by `./console exp:help:examples` from `kernel/classes/datatypes/ezuser/ezusertype.php`, `settings/content.ini`,

`design/standard/templates/content/datatype/view/ezuser.tpl`, `design/standard/templates/content/datatype/edit/ezuser.tpl`. Do not edit this page: change the source and generate again (how). Free documentation: GFDL 1.3 or later.

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.