



Hello - my name is Andy Tough



Hello - my name is Andy Tough

I provide technical support to people who use Macs

I build websites - usually using MODX



Hello - my name is Andy Tough

I provide technical support to people who use Macs

I build websites - usually using MODX

You can find me at andytough.com

I rarely tweet, but when I do I am @andytough



Hello - my name is Andy Tough

I provide technical support to people who use Macs

I build websites - usually using MODX

You can find me at andytough.com

I rarely tweet, but when I do I am @andytough

I am also a dog training instructor. But that is probably not relevant right now

How MODX and four Amazing MODX Extras
are enabling me to build a website which centres
around people and their professional profiles

Recently I have been building a website for dog trainers that...

- Allows members to create a front end account on the site
- Allow members to edit a professional profile on the site, again from the front end
- Displays members professional profiles
- Allows some people to display information about one or more businesses that they run

Some requirements I needed to allow for

- Collect information during the registration process that needed to be stored in non-standard user fields
- Limiting registration to people who have been invited to join, and have been provided with an ID number and registration verification code
- Automatically displaying peoples professional profiles without having to create a resource for each person
- Allow specific users to create and edit resources

When talking about the MODX Extras that have enabled me to meet the sites requirements I will not cover things which are well documented.

Instead I am going to focus on the problems I have encountered and some of the solutions or workarounds I have come up with.

Collect information during the registration process that needed to be stored in non-standard user fields

- MODX Extra used: Login
- Written by Shaun McCormick
- <https://docs.modx.com/extras/revo/login>

Login.Register sample

Here is the code that allows for creating an account and for the email address, username and password to be stored.

```
[[!Register?
&submitVar=`registerbtn`
&activationResourceId=`12`
&activationEmailTpl=`myActivationEmailTpl`
&activationEmailSubject=`Thanks for Registering!`
&submittedResourceId=`45`
&usergroups=`Marketing,Research`
&validate=`nospam:blank,
username:required:minLength=^8^,
password:required:minLength=^8^,
password_confirm:password_confirm=^password^,
fullname:required,
email:required:email`
&placeholderPrefix=`reg.`
]]
```

<https://docs.modx.com/extras/revo/login/login.register/>

Login.Register username duplicates: Issues: part 1

If someone tried to register with a user name that was already in use, registration would fail, without displaying an error

Login.Register username duplicates: Issues: part 1 - Solution

If someone tried to register with a user name that was already in use, registration would fail, without displaying an error

Adding **checkExists** to the user name entry solved that problem

```
&validate=`username:required`
```

```
&validate=`username:required:checkExists`
```

Login.Register username duplicates: Issues part 2

If someone tried to register with a user name that had already been requested, **but the first person to request the user name had yet to activate their account** then adding **checkExists did not work**. Instead the second persons account would delete the first persons account.

Login.Register username duplicates: Issues part 2 - Solution

If someone tried to register with a user name that had already been requested, **but the first person to request the user name had yet to activate their account** then adding **checkExists did not work**. Instead the second persons account would delete the first persons account.

Adding **&removeExpiredRegistrations=`0`** solved that problem

`&removeExpiredRegistrations=`0``

Login.Register: An observation

You can set a minimum length for user passwords in Login.Register.

Login.Register: An observation

You can set a minimum length for user passwords in your Login.Register form.

Make sure that the length you set in your form is not shorter than the length you set in your MODX configuration.

System Settings & Events / Authentication and Security /
Minimum Password Length

Additional user fields

I needed to store a lot of information about each member.

I wanted to do this using user fields associated with the members account

Some information needed to be captured during registration, the rest of the information could be added or edited later.

MODX has an **MODX User Extended Fields** area. This stores any additional user data as a JSON string. I didn't think this field was not going to work for the large amount of user information I needed to store.

Additional user fields

For this requirement I turned to Bob Ray's:

Class Extender - Extend modUser

<https://modx.com/extras/package/classextender>

<https://bobsguides.com/classextender-class.html>

Additional user fields

Class Extender - Extend modUser:

Step 1

Populate two chunks containing XML data, with the names and details of the fields you need...

Additional user fields:

Chunk: MyExtUserSchema

This chunk is used when creating an extra table in your MODX database and creating the fields within it.

```
<?xml version="1.0" encoding="UTF-8"?>
<model package="extendeduser" baseClass="xPDOObject" platform="mysql"
defaultEngine="MyISAM" tablePrefix="ext_" version="1.0.0">
  <!-- extend the modUser class -->
  <object class="extUser" extends="modUser">
    <composite alias="Data" local="id" class="userData"
foreign="userdata_id" cardinality="one" owner="local"/>
  </object>
  <object class="userData" table="user_data" extends="xPDOSimpleObject">
    <field key="userdata_id" dbtype="int" precision="11" phptype="integer"
null="false" attributes="unsigned"/>
    <!--////////////////////// ONE SAMPLE FIELD BELOW \\\\\\\\\\\\\\\\\\\-->
    <field key="idCode" dbtype="varchar" precision="50" phptype="string"
null="true"/>
    <!--////////////////////// END OF SAMPLE FIELDS \\\\\\\\\\\\\\\\\\\-->
    <aggregate alias="User" class="modUser" local="userdata_id"
foreign="id" cardinality="one" owner="foreign"/>
    <aggregate alias="Profile" class="modUserProfile" local="userdata_id"
foreign="internalKey" cardinality="one" owner="foreign"/>
  </object>
</model>
```

Additional user fields:

Chunk: MyExtraUserFields

This chunk is used to create a view of these fields in your MODX Manager / Manage / Users area.

```
<div class="x-form-item x-tab-item">
  <label class="x-form-item-label" style="width: auto; font-weight:bold;
float:none; font-size:12px">ID Code</label>
  <div class="x-form-item x-tab-item">
    <input type="text" name="idCode" value="[[+idCode]]" style="width:
597px;" class="x-form-text x-form-field"
    onClick="Ext.getCmp('modx-panel-user').markDirty();" />
  </div>
</div>
```

Additional user fields

Class Extender - Extend modUser:

Step 2

Run ClassExtender by loading a page that ClassExtender installs on your site:

<https://yoursite.com/class-extender/extend-moduser>

Additional user fields

So I had the user fields I wanted. Allowing logged in users to edit these fields is quite straight forward and well documented.

A very minimal example of code that would allow a logged in user to edit their **extended user fields** is shown on the next slide.

Updating additional user fields

```
[[!ExtUserUpdateProfile]]  
[[!UpdateProfile?  
&useExtended=`0`]]
```

Line one ensures Extend modUser fields are included in your profile update

Line two is the normal UpdateProfile call

Line three ensures that data intended for your Extend modUser fields does not end up in the default MODX User Extended Fields

Login.Register - Capturing additional user field data during registration

Remember the original Login.Register code...

```
[[!Register?  
&submitVar=`registerbtn`  
&activationResourceId=`12`  
&activationEmailTpl=`myActivationEmailTpl`  
&activationEmailSubject=`Thanks for Registering!`  
&submittedResourceId=`45`  
&usergroups=`Marketing,Research`  
&validate=`nospam:blank,  
username:required:minLength=^6^:checkExists,  
password:required:minLength=^6^,  
password_confirm:password_confirm=^password^,  
fullname:required,  
email:required:email`  
&placeholderPrefix=`reg.`  
&removeExpiredRegistrations=`0`  
]]
```

Login.Register - Capturing additional user field data during registration

Just add **&postHooks=`ExtUserRegisterPosthook`** to capture your mod extendedUser fields

```
[[!Register?  
&submitVar=`registerbtn`  
&activationResourceId=`12`  
&activationEmailTpl=`myActivationEmailTpl`  
&activationEmailSubject=`Thanks for Registering!`  
&submittedResourceId=`45`  
&usergroups=`Marketing,Research`  
&validate=`nospam:blank,  
username:required:minLength=^6^:checkExists,  
password:required:minLength=^6^,  
password_confirm:password_confirm=^password^,  
fullname:required,  
email:required:email`  
&placeholderPrefix=`reg.`  
&removeExpiredRegistrations=`0`  
&postHooks=`ExtUserRegisterPosthook`  
]]
```

Login.Register - Capturing fullname whilst collecting first and last name

Using First name and Last Name fields. But also populating MODX's default fullname field automatically using a snippet called by a preHook and a hidden field in the html form

```
&preHooks=`fullname`
```

fullname snippet code

```
<?php
    $fullname = $hook->getValue('firstName').' '.$hook->getValue('lastName');
    $hook->setValue('fullname',$fullname);
    return true;
?>
```

In the html

```
<input type="hidden" name="fullname" id="fullname" value="[[!+firstName]] [[!+lastName]]" />
```

Limiting registration

Two required fields held data that was supplied to prospective members and had to validate

Membership number - this follows a pattern

Registration code - this is a pre-agreed string of text that can be changed a little by the site owner

Limiting registration

Login.Register supports Regular Expressions!

```
&validate=`registerCode:required:true:regexp=^/andysignup(\d\d\d\d)$/^,  
idCode:required:true:regexp=^/[R](\d\d)[A-Z][\-](\d\d)[A-Z]$/^`
```

registerCode example: **andysignup1234** - last four digits could be any four digits

idCode example: **R07D-07Z**

Service Providers code: **S07D-07Z**

```
[S](\d\d)[A-Z][\-](\d\d)[A-Z]
```

I urge you to explore Regular Expressions at <http://regexpr.com>

Displaying member profiles

Automatically displaying peoples professional profiles using a template resource rather than having to create a resource for each person

Displaying member profiles using getUrlParam

getUrlParam: <https://modx.com/extras/package/geturlparam>

by Paul Merchant

Displaying member profiles using getUrlParam

Send members idCode to the page that will display their profile

```
<a href="[[~106? &val=`[[+idCode]]`]]">
```

ie: <https://modx.andytough.com/profileview?val=R07D-07Z>

Display their details on that page using GetExtUsers having captured their idCode value using getUrlParam

```
[[!GetExtUsers?  
  &prefix=`extUser.`  
  &where={`{"idCode:=": "[[!getUrlParam? &name=`val` &string=`1`]]"}`}  
  &extUserOuterTpl=`ExtUserOuterViewTpl`  
  &extUserRowTpl=`extUserRowViewTpl`  
  &extUserInnerTpl=`ExtUserInnerViewTpl`  
]]
```

<https://modx.andytough.com/list-of-signatories>

Display information about one or more businesses

Some members will run businesses that are Service Providers. They need to display the services their business provides. I can't store their service provider data in their Extend modUser fields because...

- They may pass their business on to someone else
- They may run more than one business

So they need a Resource to store the data for each business they run

Display information about one or more businesses using NewsPublisher

News Publisher: <https://modx.com/extras/package/newspublisher>

by Bob Ray

Documentation: <https://bobsguides.com/newspublisher-tutorial.html>

News Publisher allows the creation and editing of resources from the front end.

Display information about one or more businesses using NewsPublisher

Add an existing member to the **service providers** group as an **editor** by visiting a page containing this code whilst logged in.

Using News Publisher's Policy Template this allows the user to create and edit their own resources using News Publisher

```
[[!addUserToServiceproviders]]
```

```
<p>Welcome: [[+personalize.name]] | <a href="[[~145]]">Success please create your page</a></p>
```

<https://modx.andytough.com/signatories-area/add-a-signatory-to-the-service-providers-group>

Display information about one or more businesses using NewsPublisher

addUserToServiceproviders snippet

```
<?php
    $groupname = 'serviceproviders';
    if(!$modx->user->isMember($groupname)) {
        $modx->user->joinGroup($groupname, 'Editor');
        $userId = $modx->user->get('id');
        // fix for a bug in joinGroup()
        unset($_SESSION["modx.user.{$userId}.userGroups"]);
        $context = 'web';
        // get the attributes to reload
        $targets = explode(',', $modx->getOption('principal_targets',
            null,
            'modAccessContext,modAccessResourceGroup,modAccessCategory,sources.modAccessMediaSource'));
        array_walk($targets, 'trim');
        // reload the attributes
        $modx->user->loadAttributes($targets, $context, true);
    }
?>
```

Display information about one or more businesses using NewsPublisher

The page which allows the user to create their own resource

```
[ [!NewsPublisher@rc?  
&show=`pagetitle,alias`  
&initrte=`0`  
&initdatepicker=`1`  
&parentid=`35`  
&published=`1`  
&hidemenu=`1`  
&templateid=`12`  
&ErrorTpl=`rcErrorTpl`  
&ownPagesOnly=`1`  
&captions=`alias:Your Service Providers ID code`  
&outertpl=`rcOuterTpl`  
]]
```

The alias for the page is created automatically. I have captioned the alias field as 'Your Service Providers ID code'. So the service providers url will be their Service Providers code number

<https://modx.andytough.com/signatories-area/add-a-service-provider-page>

Display information about one or more businesses using NewsPublisher

Limit the service provider to adding only one page with the onePageOnly snippet...

```
[[!onePageOnly]]
[[!NewsPublisher@rc?
&show=`pagetitle,alias`
&initrte=`0`
&initdatepicker=`1`
&parentid=`35`
&published=`1`
&hidemenu=`1`
&templateid=`12`
&ErrorTpl=`rcErrorTpl`
&ownPagesOnly=`1`
&captions=`alias:Your Service Providers ID code`
&outertpl=`rcOuterTpl`
]]
```

Display information about one or more businesses using NewsPublisher

Limit the service provider to adding only one page with the **onePageOnly** snippet...

```
<?php
    /* Keep non-admins from creating more than one Doc */
    $sorryPageId = 160; // change to the ID of your "sorry" resource which
    should contain a chunk tag for the "sorry" chunk.
    $sorryChunk = 'NpSorryOverLimit'; // Name of chunk telling them they're
    out of luck.
    $pageLimit = 1;
    if (! $modx->user->isMember('Administrator')) {
        /* Check the user's page count */
        $userId = $modx->user->get('id');
        $count = (int) $modx->getCount('modResource', array('createdby' =>
$userId));
        if ($count >= $pageLimit) { /* They've already created at least one
page */
            $msg = $modx->getChunk($sorryChunk);
            $url = $modx->makeUrl($sorryPageId, "", "", "full");
            $modx->sendRedirect($url);
        }
        /*}*/
    }
?>
```

Four MODX Extras

Login: <https://docs.modx.com/extras/revo/login>

Class Extender: <https://modx.com/extras/package/classextender>

getUrlParam: <https://docs.modx.com/extras/revo/geturlparam>

News Publisher: <https://modx.com/extras/package/newspublisher>

Several people

Shaun McCormick: <http://www.splittingred.com/>

Bob Ray: <https://bobsguides.com/modx.html>

Paul Merchant: <https://forums.modx.com/u/paulmerchant/>

And a great website:

Regexr: <http://regexr.com>

The MODX Community

Forums: <https://forums.modx.com/>

Documentation - become a contributor: <https://modxos.wufoo.com/forms/become-a-modx-documentation-editor/>

Documentation: <https://docs.modx.com/>

I would like to rename my talk...

How MODX, four MODX Extras
and The Amazing MODX Community
are enabling me to build a website which centres
around people and their professional profiles

Resources from this talk

Slides and links from this talk will be available on my website shortly:

andytough.com/articles/modx-meetup

Sample site: <http://modx.andytough.com>