

AUTOMATION

BOOTCAMP!

cPanel Conference '10

OCTOBER 4TH-6TH, 2010

Modern cPanel

The API Unification Project

Phil King

version 09-23

AUTOMATION

BOOTCAMP!

cPanel Conference '10

Simplify cPanel code

- support many different ways of adding functionality
- currently two approaches for cPanel methods
- defines Unified API, and migrate current codebase
- currently three tag syntaxes
- standardize on Template Toolkit
- 50 modules / 750 API methods
- game changing initiative

Terminology

OCTOBER 4TH-6TH, 2010



cPanel

API layer

x3 theme

XML

Desktop

PHP

AUTOMATION

BOOTCAMP!

cPanel Conference '10

```
<cpanel Users="add_user($FORM{name}, $FORM{pass})">
```

- Applying a template to the API call not possible
- Data printed directly to screen
- Errors and messages output from method
- HTML frontend is assumed

AUTOMATION

BOOTCAMP!

cPanel Conference '10

```
<?cp Users::list_users(  
  [tr class="row-%"][td]%/td]  
  [td]%/td[/tr],  
  *num, ^url, &html)  
  searchFilter=$FORM{'filter'} ?>
```

- Implicit loop over an ArrayOfHashes
- Syntax heavy
- Execute with parameters, iterate over data, applying template and macros (HTML and URI encoding)

API2 tags (cont'd)

OCTOBER 4TH-6TH, 2010

AUTOMATION

BOOTCAMP!

cPanel Conference '10

Assume Success!

```
<?cp Users::add_user(...) name=$FORM{'name'} ||  
  '[tr][td]$LANG{'ignoresuc'}[/td][/tr]' ?>
```

C PANEL 11

User Accounts

New User Successfully Created

Complete failure to create new user **SuccessMan**. Ignore any messages of success. This can only result in failure!

[← Go Back](#)

API2 tags (cont'd)

OCTOBER 4TH-6TH, 2010

AUTOMATION

BOOTCAMP!

cPanel Conference '10

No syntax for variable capture

```
<?cp Users::list_users(...) ?>
```

Complicated templates use multiple API calls

- cherry-pick pieces of data
- often recompute same data



From editfilter.html

OCTOBER 4TH-6TH, 2010

`api2_loadfilter` assigns to `$currentfilter`

Filter Name: `api2_filtername`

The filter name must be unique. If you give the filter the same name as another filter, the previous filter will be overwritten.

Rules `api2_filtrerrules`

▲	<input type="text" value="From"/>	◆	<input type="text" value="contains"/>	◆	or ◆	-	+
▼	<input type="text" value="phil@cpanel.net"/>						
▲	<input type="text" value="Reply Address"/>	◆	<input type="text" value="contains"/>	◆		-	+
▼	<input type="text" value="phil@cpanel.net"/>						

Actions `api2_filteractions`

<input type="text" value="Deliver to folder"/>	◆	<input type="text" value="/Junk"/>	<input type="button" value="Change"/>	-	+
--	---	------------------------------------	---------------------------------------	---	---

AUTOMATION

BOOTCAMP!

cPanel Conference '10

Template Toolkit

OCTOBER 4TH-6TH, 2010

From x3/mime/list_hotlinks.tmpl:

```
[% result = execute('Mime', 'list_hotlinks') %]  
[% hotlinks = result.data %]
```

Former API2 calls	Template Toolkit
hotlinkingenabled	hotlinks.status
gethotlinkurls	hotlinks.urls
gethotlinkext	hotlinks.extensions
linkallownull	hotlinks.allow_null
gethotlinkrurl	hotlinks.redirect_url

AUTOMATION

BOOTCAMP!

cPanel Conference '10

Template Toolkit (cont'd)

OCTOBER 4TH-6TH, 2010

```
[% result = execute('Users', 'list_users',
  { searchFilter=FORM.filter }) %]
[% IF result.status %]
  [% FOREACH q = result.data %]
    <tr><td>[% q.url FILTER uri %]</td>
      <td>[% q.user_text FILTER html %]</td></tr>
  [% END %]
[% ELSE %]
  [% cptext('LocaleErrorKey') %]
  [% FOREACH error = result.errors %]
    <font color="red">[% error %]</font>
  [% END %]
[% END %]
```

API2 tag recap

OCTOBER 4TH-6TH, 2010

AUTOMATION

BOOTCAMP!

cPanel Conference '10

Equivalent API2 tag

```
<?cp Users::list_users(  
  [tr class="row-%"][td]%/td]  
  [td]%/td[/tr],  
  *num, ^url, &html)  
  searchFilter=$FORM{'filter'} ||  
  '[tr][td]$LANG{'ErrorKey'}[/td][/tr]' ?>
```

Template pros/cons

OCTOBER 4TH-6TH, 2010

AUTOMATION

BOOTCAMP!

cPanel Conference '10

Template Toolkit analysis

- More verbose, but clearer
- The “Perl standard”
- One tag syntax rather than three
- Eliminate code in cpanel binary
- Speed increase through caching
- Well-tested, well documented



Cpanel::Result object

OCTOBER 4TH-6TH, 2010

Stores data

- `$result->data(\@arrayOfHash)`
- `[% data = result.data %]`

Handles messaging and locale

- `$result->error("Error: [_1]", $user);`
- `$str = $result->errors_as_string()`
- `[% FOR err = result.errors %]`
- Analogous methods for "messages"

Manages Status

- "return 1" means `$result->status(1)`
- `[% IF !result.status %] Error [% END %]`

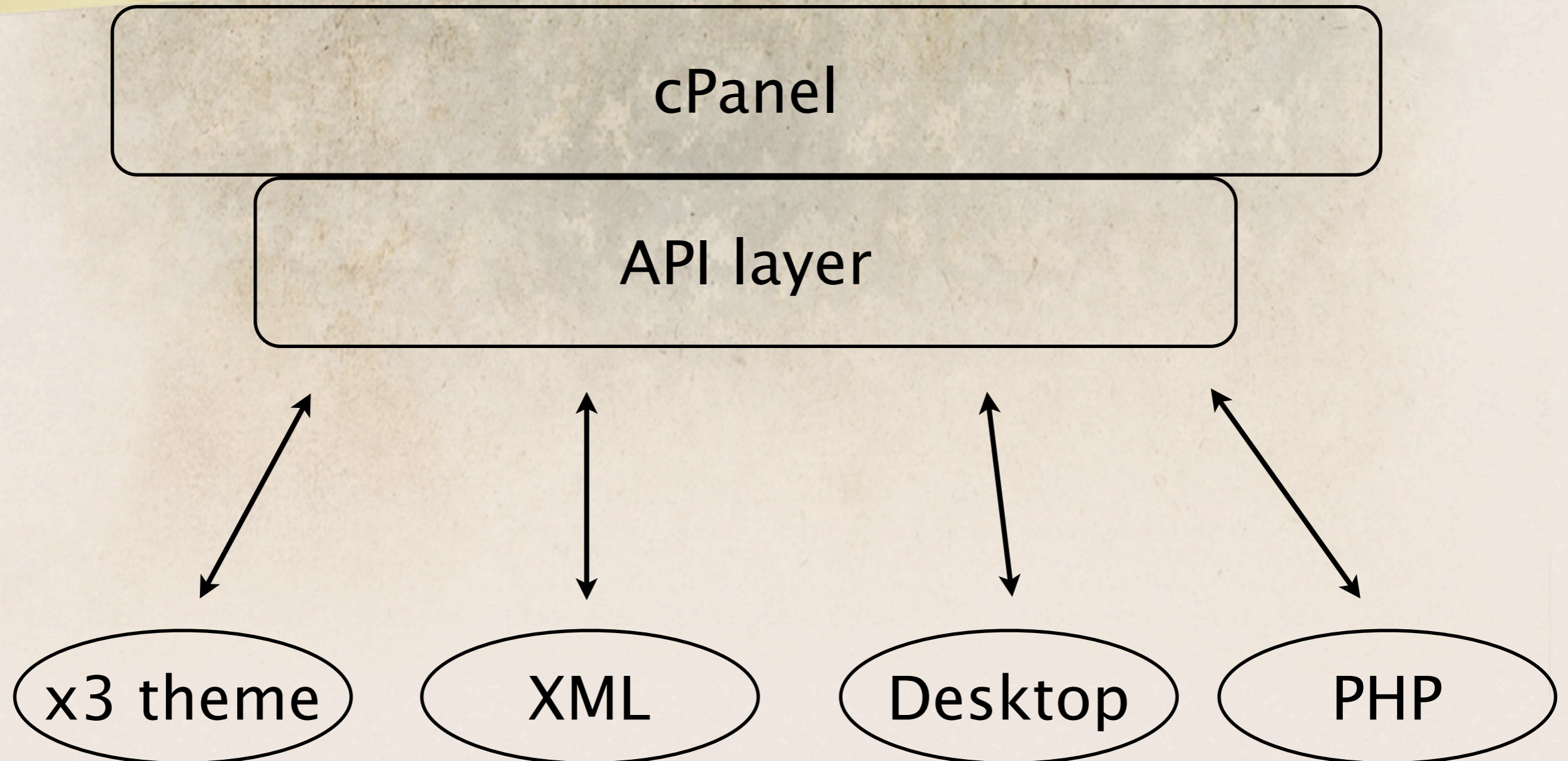
Cpanel::Result

OCTOBER 4TH-6TH, 2010

AUTOMATION

BOOTCAMP!

cPanel Conference '10



API method internals

OCTOBER 4TH-6TH, 2010

AUTOMATION

BOOTCAMP!

cPanel Conference '10

- Methods receive an argument hash and the result object

```
my ( $args, $result ) = @_;
```

```
my ( $user, $pass ) = @{$args}{qw(user pass)};
```

- Methods return either true or false (undefined)

```
if ( $some_error ) {
```

```
    $result->error( 'User "[_1]" failed', $user );
```

```
    return;
```

```
}
```

```
$result->data( \@arrayOfHash )
```

```
return 1;
```

- And store computed data in the result object.

AUTOMATION

BOOTCAMP!

cPanel Conference '10

Complete API example

OCTOBER 4TH-6TH, 2010

In Cpanel::API::Users package:

```
sub change_password {
    my ( $args, $result ) = @_;
    my ( $user, $pass )    = @{$args}{qw(user pass)};

    my ( $status, $arrayOfHash ) = _util_function( $user, $pass );
    if ( !$status ) {
        $result->error( 'Problem with user "[_1]"', $user );
        return;
    }
    $result->message( 'Password changed: "[_1]"', $pass );
    $result->data($arrayOfHash);
    return 1;
}
```

AUTOMATION

BOOTCAMP!

cPanel Conference '10

Support for Legacy APIs

OCTOBER 4TH-6TH, 2010

In Cpanel::Users package:

```
sub legacy_passwd {  
    my ($FORM) = @_;  
    my $result = execute( 'Users', 'change_password',  
        { user => $FORM{'user'}, pass => $FORM{'pass'} } );  
    if ( $result->status() ) {  
        return ( 1, $result->data() );  
    }  
    return ( 0, $result->errors_as_string() );  
}
```

- Legacy API call utilizes the Unified API method
- Uses the result object to return the expected “quirks”

AUTOMATION

BOOTCAMP!

cPanel Conference '10

Points for Style

OCTOBER 4TH-6TH, 2010

Consistent syntax at every level

```
my $result = execute('Module', 'method', { user=>$FORM{'user'} });  
[% result = execute('Module', 'method', { user=FORM.user }) %]
```

Method name conventions within subsystems

- list_accounts
- get_account
- add_account
- delete_account

Legible method names and arguments

- theLegibilityOfPerlIsBadEnough
- lgblyprlbadenough
- the_legibility_of_perl_is_bad_enough
- thelegibilityofperlisbadenough

What does this mean?

OCTOBER 4TH-6TH, 2010

AUTOMATION

BOOTCAMP!

cPanel Conference '10

cPanel development much simpler

- custom templates (themes, branding)
- Unified API
- Cpanel::Result object
- easy to learn, easy to remember
- enables market for 3rd party development
- consistency

AUTOMATION

BOOTCAMP!

cPanel Conference '10

“When can I have this?” and/or “I don’t want this yet.”

- phased delivery
- x3 theme (API1 and API2 tags) continue to work
- Mime, Ftp, SSL, Email
- `<?cptt filename.tmpl ?>` and `[% result = execute(...) %]`
- More modules to follow

The Big Picture

OCTOBER 4TH-6TH, 2010

AUTOMATION

BOOTCAMP!

cPanel Conference '10

Big Next Steps

- Unified API for both WHM and cPanel
- Plugin system
- Extend Cpanel::Result for long running processes
- cPanel as application server

Send feedback

OCTOBER 4TH-6TH, 2010

AUTOMATION

BOOTCAMP!

cPanel Conference '10

Phil King

phil@cpanel.net