Form API now allows for object methods as callbacks

Posted by [plach](https://www.drupal.org/u/plach) on 16 August 2012

**Change record status:**

Published ([View all published change records](https://www.drupal.org/list-changes))

**Project:**

[Drupal core](https://www.drupal.org/project/drupal)

**Introduced in branch:**

8.x

**Issues:**

[#1499596: Introduce a basic entity form controller](https://www.drupal.org/node/1499596)

[#1802072: Ajax callbacks should be callbacks defined as function or class method](https://www.drupal.org/node/1802072)

[#1805688: Support methods as #element\_validate callbacks](https://www.drupal.org/node/1805688)

[#1985838: document Form API #element\_validate callback](https://www.drupal.org/node/1985838)

**Description:**

The form state build info supports a new 'callback' key that can be used to explicitly set a form builder callback. Moreover now any callable can be used as a form handler/callback. This allows the new [Entity form controllers](http://drupal.org/node/1734556) to work, and can also be used to collect all the required functions for a form into a class more easily. The old syntax is still valid.

Before:

class MyForm {

function build($form, &$form\_state) {

// ...

$form['#validate'] = 'my\_oo\_form\_validate';

$form['#submit'] = 'my\_oo\_form\_submit';

return $form;

}

}

function my\_oo\_form\_page(MyForm $object)

return drupal\_get\_form('my\_oo\_form', $object);

}

function my\_oo\_form($form, &$form\_state, $object)

$form\_state['object'] = $object;

return $object->build($form, $form\_state);

}

function my\_oo\_form\_validate($form, &$form\_state) {

$form\_state['object']->validate($form, $form\_state);

}

function my\_oo\_form\_submit($form, &$form\_state) {

$form\_state['object']->submit($form, $form\_state);

}

After:

class MyForm {

function build($form, &$form\_state) {

// The ::methodName syntax is automatically expanded to array($this, 'validate').

// This only works for the form object, not partial forms in widgets, plugins or form alters.

$form['#validate'][] = '::validate';

$form['#submit'][] = '::submit';

}

}

function my\_oo\_form\_page(MyForm $object)

$form\_state = array();

$form\_state['build\_info']['callback'] = array($object, 'build');

return drupal\_build\_form('my\_oo\_form', $form\_state);

}

Additionally, '#element\_validate' and '#ajax' callbacks can now defined as methods, as well.

Before:

class MyField {

function build($form, &$form\_state) {

$form['foo'] = array(

// ...

'#element\_validate' = array('my\_element\_validator'),

);

$form['bar'] = array(

// ...

'#ajax' => array(

'callback' => 'my\_ajax\_callback',

'wrapper' => 'foo',

),

);

return $form;

}

}

function my\_ajax\_callback() {

// Do something fancy.

}

function my\_element\_validator() {

// Validate something.

}

After:

use Drupal\Core\Form\FormStateInterface;

class MyField {

function build($form, &$form\_state) {

$form['foo'] = array(

// ...

'#element\_validate' = array(array($this, 'myElementValidator')),

);

$form['bar'] = array(

// ...

'#ajax' => array(

'callback' => array($this, 'myAjaxCallback'),

'wrapper' => 'foo',

),

);

return $form;

}

function myAjaxCallback() {

// Do something fancy.

}

/\*\*

\* Validates my element.

\*

\* @param array $element

\* The form element to process.

\* @param \Drupal\Core\Form\FormStateInterface $form\_state

\* The form state.

\* @param array $complete\_form

\* The complete form structure.

\*/

public static function myElementValidator(&$element, FormStateInterface $form\_state, &$complete\_form) {

// Validate something.

}

}

See core's [Url::validateUrl()](https://api.drupal.org/api/drupal/core%21lib%21Drupal%21Core%21Render%21Element%21Url.php/function/Url%3A%3AvalidateUrl/8.2.x) for an example.

If operating outside of a class context (say in a form alter hook), a fully-qualified class can be specified instead of $this. In this case, it would be 'Drupal\my\_module\Path\MyField'.

**Impacts:**

Module developers

Updates Done (doc team, etc.)

**Online documentation:**

Not done

**Theming guide:**

Not done

**Module developer documentation:**

Not done

**Examples project:**

Not done

**Coder Review:**

Not done

**Coder Upgrade:**

Not done

**Other:**

Other updates done

* [Log in](https://www.drupal.org/user/login?destination=comment/reply/1734540%23comment-form) or [register](https://www.drupal.org/user/register?destination=comment/reply/1734540%23comment-form) to post comments

Comments

[](https://www.drupal.org/u/stalski)

[**create valid callback**](https://www.drupal.org/node/1734540#comment-6498124)

[Stalski](https://www.drupal.org/u/stalski)*commented 5 years ago*

It needs one altering though to be a valid callback.

class MyForm {

function build($form, &$form\_state) {

// ...

$form['#validate'] = array(array($this, 'validate'));

$form['#submit'] = array(array($this, 'submit'));

}

}

* [Log in](https://www.drupal.org/user/login?destination=comment/reply/1734540%23comment-form) or [register](https://www.drupal.org/user/register?destination=comment/reply/1734540%23comment-form) to post comments

[](https://www.drupal.org/u/plach)

[**Updated**](https://www.drupal.org/node/1734540#comment-6498210)

[plach](https://www.drupal.org/u/plach)*commented 5 years ago*

I updated the example, thanks.

* [Log in](https://www.drupal.org/user/login?destination=comment/reply/1734540%23comment-form) or [register](https://www.drupal.org/user/register?destination=comment/reply/1734540%23comment-form) to post comments

[](https://www.drupal.org/u/xano)

[**Note that #value\_callback**](https://www.drupal.org/node/1734540#comment-7645125)

[Xano](https://www.drupal.org/u/xano)*commented 4 years ago*

Note that #value\_callback does not yet support methods. See [#2040559: Replace function\_exists() with is\_callable() in FormBuilder](https://www.drupal.org/node/2040559).

[My name is Bart - web & events](http://mynameisbart.com/)

* [Log in](https://www.drupal.org/user/login?destination=comment/reply/1734540%23comment-form) or [register](https://www.drupal.org/user/register?destination=comment/reply/1734540%23comment-form) to post comments

[](https://www.drupal.org/u/sprocketman)

[**AJAX callbacks**](https://www.drupal.org/node/1734540#comment-7933831)

[sprocketman](https://www.drupal.org/u/sprocketman)*commented 4 years ago*

Is there any similar method to this for form elements using AJAX? For example:

$form['my\_element'] = array(

#type => 'radios',

#title => t('My AJAX Element'),

...

#ajax => array(

'callback' => array(array($this, 'my\_method')),

'wrapper' => 'my-wrapper',

),

);

* [Log in](https://www.drupal.org/user/login?destination=comment/reply/1734540%23comment-form) or [register](https://www.drupal.org/user/register?destination=comment/reply/1734540%23comment-form) to post comments

[](https://www.drupal.org/u/sprocketman)

[**Let me rephrase my**](https://www.drupal.org/node/1734540#comment-7933875)

[sprocketman](https://www.drupal.org/u/sprocketman)*commented 4 years ago*

Let me rephrase my question...I'm actually curious about using callbacks that are not located in the current class (so not using $this, but some other class object). Is that possible? If so, how might that look?

* [Log in](https://www.drupal.org/user/login?destination=comment/reply/1734540%23comment-form) or [register](https://www.drupal.org/user/register?destination=comment/reply/1734540%23comment-form) to post comments

[](https://www.drupal.org/u/colan)

[**Example updated**](https://www.drupal.org/node/1734540#comment-11590289)

[colan](https://www.drupal.org/u/colan)*commented about a year ago*

I just updated the page to include information on this.

* [Log in](https://www.drupal.org/user/login?destination=comment/reply/1734540%23comment-form) or [register](https://www.drupal.org/user/register?destination=comment/reply/1734540%23comment-form) to post comments

[](https://www.drupal.org/u/aaronbauman)

[**2 questions**](https://www.drupal.org/node/1734540#comment-8134605)

[aaronbauman](https://www.drupal.org/u/aaronbauman)*commented 4 years ago*

1. Should the callback methods be public, protected, or private?  
2. Must the callback methods be connected to a route in .routing.yml?

* [Log in](https://www.drupal.org/user/login?destination=comment/reply/1734540%23comment-form) or [register](https://www.drupal.org/user/register?destination=comment/reply/1734540%23comment-form) to post comments

[](https://www.drupal.org/u/colan)

[**Example updated. No.**](https://www.drupal.org/node/1734540#comment-11590293)

[colan](https://www.drupal.org/u/colan)*commented about a year ago*

1. Should the callback methods be public, protected, or private?

I just updated the page to include information on this.

2. Must the callback methods be connected to a route in .routing.yml?

No.