FormAPI, AJAX and Node.js

Overview session for people who are new to coding in Drupal.

Ryan Weal
Kafei Interactive Inc.
http://kafei.ca

These slides posted to: http://verbosity.ca

Why?

- New developers bring their knowledge from other web apps, but in many cases Drupal is different
- Learning these « Drupal-isms » will help make your life easier

What makes Drupal different?

- <form> tags are not allowed!
- JavaScript is done on the field level in many cases
- Adding JS libraries is common
- Sometimes DrupalJS has limits.

Where is JavaScript used in Drupal?

- In forms where users input their data → FormAPI
- Packaged in themes and modules which provide front-end functionality → DrupalAPI
- When we communicate with remote systems → AJAX / JSON

FormAPI – What is it?

- Library of re-usable form elements
  - https://api.drupal.org/api/drupal/developer!topics!forms_
- Easy way to add JS to forms
- Customizable – override any Drupal form
- Flexible : store values between pages, etc.

Form Controls

Legend:
X = attribute can be used with this type
- = attribute is not applicable to this type

#type	checkbox	checkboxes	date	fieldset	file	machine_name	managed_file	password	password_confirm	radio	radios	select	tableselect	text_format	textarea	textfield	vertical_tabs	weight	
#access	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	#access
#after_build	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	#after_build
#ajax	X	X	-	-	-	X	-	-	-	X	X	X	X	X	X	X	-	-	#ajax
#array_parents	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	#array_parents
#attached	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	#attached
#attributes	X	X	X	X	X	X	X	X	-	X	X	X	X	X	X	X	-	X	#attributes
#autocomplete_path	-	-	-	-	-	X	-	-	-	-	-	-	-	-	-	X	-	-	#autocomplete_path
#collapsed	-	-	-	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	#collapsed
#collapsible	-	-	-	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	#collapsible
#cols	-	-	-	-	-	-	-	-	-	-	-	-	-	X	X	-	-	-	#cols
#default_tab	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	X	-	#default_tab
#default_value	X	X	X	-	-	X	-	-	-	X	X	X	X	X	X	X	-	X	#default_value
#delta	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	X	#delta
#description	X	X	X	X	X	X	X	X	X	X	X	X	-	X	X	X	-	X	#description
#disabled	X	X	X	-	X	X	X	X	X	X	X	X	-	X	X	X	-	X	#disabled
#element_validate	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	#element_validate
#empty	-	-	-	-	-	-	-	-	-	-	-	-	X	-	-	-	-	-	#empty
#empty_option	-	-	-	-	-	-	-	-	-	-	-	X	-	-	-	-	-	-	#empty_option
#empty_value	-	-	-	-	-	-	-	-	-	-	-	X	-	-	-	-	-	-	#empty_value
#field_prefix	X	-	-	-	-	X	-	X	X	X	-	X	-	-	X	X	-	-	#field_prefix
#field_suffix	X	-	-	-	-	X	-	X	X	X	-	X	-	-	X	X	-	-	#field_suffix
#group	-	-	-	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	#group
#header	-	-	-	-	-	-	-	-	-	-	-	-	X	-	-	-	-	-	#header
#type	checkbox	checkboxes	date	fieldset	file	machine_name	managed_file	password	password_confirm	radio	radios	select	tableselect	text_format	textarea	textfield	vertical_tabs	weight	#type
#js_select	-	-	-	-	-	-	-	-	-	-	-	-	X	-	-	-	-	-	#js_select
#maxlength	-	-	-	-	-	X	-	X	-	-	-	-	-	-	-	X	-	-	#maxlength
#multiple	-	-	-	-	-	-	-	-	-	-	-	X	X	-	-	-	-	-	#multiple
#options	-	X	-	-	-	-	-	-	-	-	X	X	X	-	-	-	-	-	#options
#parents	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	#parents
#post_render	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	#post_render
#prefix	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	#prefix
#pre_render	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	#pre_render
#process	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	#process
#required	X	X	X	-	X	X	-	X	X	X	X	X	-	X	X	X	-	X	#required
#resizable	-	-	-	-	-	-	-	-	-	-	-	-	-	X	X	-	-	-	#resizable
#return_value	X	-	-	-	-	-	-	-	-	X	-	-	-	-	-	-	-	-	#return_value
#rows	-	-	-	-	-	-	-	-	-	-	-	-	-	X	X	-	-	-	#rows
#size	-	-	-	-	X	X	-	X	X	-	-	X	-	-	-	X	-	-	#size
#states	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	#states
#suffix	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	#suffix
#theme	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	#theme
#theme_wrappers	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	#theme_wrappers
#title	X	X	X	X	X	X	X	X	X	X	X	X	-	X	X	X	-	X	#title
#title_display	X	X	X	X	X	X	X	X	X	X	X	X	-	X	X	X	-	X	#title_display
#tree	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	#tree
#value_callback	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	#value_callback
#weight	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	#weight

FormAPI – Making forms

- Just a PHP function in a custom module
- You can call the code with drupal_get_form() or by created a hook_menu()
- Use « devel » module for debugging forms by wrapping the variable in dpm($form);

```php
$form = drupal_get_form('my_module_example_form');
...
function my_module_example_form($form, &$form_state) {
  $form['submit'] = array(
    '#type' => 'submit',
    '#value' => t('Submit'),
  );
  return $form;
}
function my_module_example_form_validate($form, &$form_state) {
  // Validation logic.
}
function my_module_example_form_submit($form, &$form_state) {
  // Submission logic.
}
```

FormAPI – Overrides make life easy

- Use standard Drupal structures like node and user, and simply override the things you need
- Change form widgets, add CSS, JS, redirects, etc.
- Put descriptive text into a form

```
function hook_form_alter(&$form, &$form_state, $form_id) {
  if (isset($form['type']) && $form['type']['#value'] . '_node_settings' == $form_id)
{
    $form['workflow']['upload_' . $form['type']['#value']] = array(
      '#type' => 'radios',
      '#title' => t('Attachments'),
      '#default_value' => variable_get('upload_' . $form['type']['#value'], 1),
      '#options' => array(t('Disabled'), t('Enabled')),
    );
  }
}
```

FormAPI – Better than <form>

- No « submit » URL : all Drupal forms typically submit back to the same page (self)
- All forms on a page get submitted if ANY submit button is clicked
- Easy to override redirect destinations
- Plays nice with caching

FormAPI – Meet AJAX

- Just add a component to your field with a PHP function name
- Your PHP function can do things like replace #id and .class
- RELOAD the same form as part of the replacement → dynamic forms!

```
function main_page() {
  return drupal_get_form('ajax_example_simplest');
}

function ajax_example_simplest($form, &$form_state) {
  $form = array();
  $form['changethis'] = array(
    '#type' => 'select',
    '#options' => array(
      'one' => 'one',
      'two' => 'two',
      'three' => 'three',
    ),
    '#ajax' => array(
      'callback' => 'ajax_example_simplest_callback',
      'wrapper' => 'replace_textfield_div',
     ),
  );

  // This entire form element will be replaced with an updated value.
  $form['replace_textfield'] = array(
    '#type' => 'textfield',
    '#title' => t("The default value will be changed"),
    '#description' => t("Say something about why you chose") . "'" .
      (!empty($form_state['values']['changethis'])
      ? $form_state['values']['changethis'] : t("Not changed yet")) . "'",
    '#prefix' => '<div id="replace_textfield_div">',
    '#suffix' => '</div>',
  );
  return $form;
}

function ajax_example_simplest_callback($form, $form_state) {
  // The form has already been submitted and updated. We can return the replaced
  // item as it is.
  return $form['replace_textfield'];
}
```

AJAX Callbacks – What are they?

- Just a regular Drupal PHP function that returns a JSON object
- Needs a special hook_menu entry so that the system does not try to render pages at same time
- The system still loads a ton of Drupal code to do the reply

Name	Location	Description
ajax_command_after	includes/ajax.inc	Creates a Drupal Ajax 'insert/after' command.
ajax_command_alert	includes/ajax.inc	Creates a Drupal Ajax 'alert' command.
ajax_command_append	includes/ajax.inc	Creates a Drupal Ajax 'insert/append' command.
ajax_command_before	includes/ajax.inc	Creates a Drupal Ajax 'insert/before' command.
ajax_command_changed	includes/ajax.inc	Creates a Drupal Ajax 'changed' command.
ajax_command_css	includes/ajax.inc	Creates a Drupal Ajax 'css' command.
ajax_command_data	includes/ajax.inc	Creates a Drupal Ajax 'data' command.
ajax_command_html	includes/ajax.inc	Creates a Drupal Ajax 'insert/html' command.
ajax_command_insert	includes/ajax.inc	Creates a Drupal Ajax 'insert' command using the method in #ajax['method'].
ajax_command_invoke	includes/ajax.inc	Creates a Drupal Ajax 'invoke' command.
ajax_command_prepend	includes/ajax.inc	Creates a Drupal Ajax 'insert/prepend' command.
ajax_command_remove	includes/ajax.inc	Creates a Drupal Ajax 'remove' command.
ajax_command_replace	includes/ajax.inc	Creates a Drupal Ajax 'insert/replaceWith' command.
ajax_command_restripe	includes/ajax.inc	Creates a Drupal Ajax 'restripe' command.
ajax_command_settings	includes/ajax.inc	Creates a Drupal Ajax 'settings' command.
ajax_command_update_build_id	includes/ajax.inc	Creates a Drupal Ajax 'update_build_id' command.

```php
$commands = array();
// Replace the content of '#object-1' on the page with 'some html here'.
$commands[] = ajax_command_replace('#object-1', 'some html here');
// Add a visual "changed" marker to the '#object-1' element.
$commands[] = ajax_command_changed('#object-1');
// Menu 'page callback' and #ajax['callback'] functions are supposed to
// return render arrays. If returning an Ajax commands array, it must be
// encapsulated in a render array structure.
return array('#type' => 'ajax', '#commands' => $commands);
```

```php
$commands = array();
$commands[] = ajax_command_replace(NULL, $output);
$commands[] = ajax_command_prepend(NULL, theme('status_messages'));
return array('#type' => 'ajax', '#commands' => $commands);
```

Front-end JavaScript

- Use an exsiting module?
- Custom → in the theme?
- Custom → in the module?

Front-end JavaScript : Adding custom code

- Load the code using drupal_add_js in your module or theme

```
<?php
drupal_add_js(drupal_get_path('library', 'modernizr') . '/js/modernizr-
1.6.min.js',
                              array('group' => JS_THEME, 'every_page' =>
TRUE));
?>
```

Front-end JavaScript : communicating with PHP

- Mixing PHP with client JS : Settings array!

```php
<?php
$my_settings = array(
  'basePath' => $base_path,
  'animationEffect' => variable_get('effect', 'none')
 );
?>
```

```php
<?php
drupal_add_js(array('myModule' => $my_settings), 'setting');
?>
```

Front-end JavaScript : behaviors

- Respond to Drupal events

```
Drupal.behaviors.myModuleBehavior = {
  attach: function (context, settings) {
    // This jQuery code ensures that this element
    // is only processed once.  It is basically saying:
    // 1) Find all elements with this class, that do not
    // have the processed class on it
    // 2) Iterate through them
    // 3) Add the myCustomBehavior-processed class (so that it will not
    // be processed again).
    $('input.myCustomBehavior', context).once('myCustomBehavior', function () {
      // Apply the myCustomBehaviour effect to the elements only once.
    });
  }
};
```

Front-end JavaScript: Mixing in libraries

- Find an existing module?
- Roll it yourself : wrap it up!
  - Run multiple versions of jQuery!
  - Prevent your code from conflicting with others

```
(function ($) {
  // Original JavaScript code.
})(jQuery);
```

Front-end JavaScript: limitations

- Want to build a chat system in Drupal?
- Need asyncronous data?
- Displaying fresh data in real-time

Back-end JavaScript: what is it?

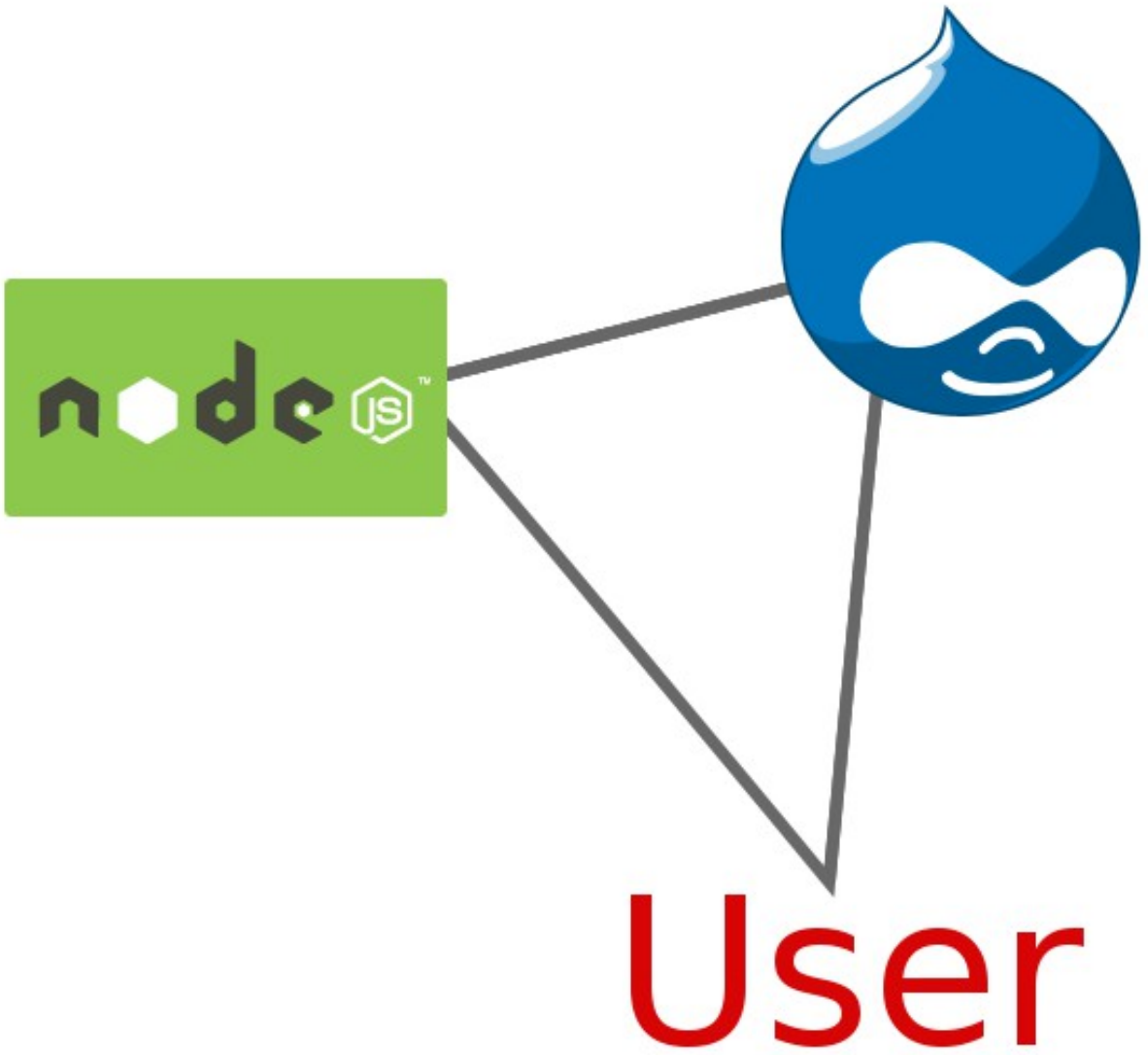
- Chrome's JavaScript engine is used to compile JS
- Most of the projects have nothing to do with front-end JS
- The best thing it does is « real-time » applications which are asynchronous

Back-end JavaScript: the techology stack

Node.js	Drupal
Firewall	Firewall
Node.js + NoSQL (JS databases)	Apache / Nginx
	PHP
	Drupal
	JavaScript

Back-end JavaScript: why not both?

- Keep a « live » connection between the server and your users
- Stop polling the server! Use socket.io
- Drupal node.js module maintains relationship with user account


node
JS
User

Node.js : how it works (with Drupal)

- Adds a socket.io JavaScript embedded in the webpage
- Connects to a node.js server that is connected to Drupal
- Spools information to and from Drupal and the user
- Define #id and .class regions that will receive data from node.js

Node.js : how to integrate

- Install node.js on your server/system
- Add « nodejs » Drupal contrib module to your site
- Edit the server.js file
- Run node from sites/all/modules/nodejs

When to use each technology :

- Only need to update fields → FormAPI
- Need to do general JavaScript on a page → include a .js file in your code base, wrapped
- Asyncronous / real-time / not-cachable data → node.js

Resources

- Form API Reference : https://api.drupal.org/api/drupal/developer!topics!fo
- The Drupal JavaScript API : https://www.drupal.org/node/304258
- Node.js contrib module : https://www.drupal.org/project/nodejs

FormAPI, AJAX and Node.js

Overview session for people who are new to coding in Drupal.

Ryan Weal
Kafei Interactive Inc.
http://kafei.ca

These slides posted to: http://verbosity.ca