WEB PAGE ARCHIVE

Austin Drupal Users Group – November 2017

David Stinemetze
Twitter: @davidstinemetze
Drupal.org/GitHub: @WidgetsBurritos

OUTLINE

- Web Page Archive Overview
- Capture Utilities and Responses
- Project Roadmap
- How You Can Help
- Demonstration
- Q&A

OUR PROBLEMS

- What did that page look like on October 17?
- What was the content of that page on June 19?
- When did that bug get introduced?
- When did that setting change?
- How will these new HTML/CSS/Javascript changes affect other parts of our website?

OUR SOLUTION

- **Web Page Archive Module**
  - https://www.drupal.org/project/issues/web_page_archive
  - https://github.com/WidgetsBurritos/web_page_archive
- **Project Maintainers**
  - David Stinemetze - @WidgetsBurritos
  - David Porter - @bighappyface
  - Paul Maddern - @pobster

MODULE OVERVIEW

- The Web Page Archive module allows you to use Drupal to perform *periodic* snapshots on *local* and *remote* websites based on a list of *URLs* or *XML sitemaps*.

- This means snapshots can be run on any site.
  - Not just your current Drupal instance.
  - Not just Drupal-based websites.
  - Not just sites you host.
  - Not just sites you own (but be nice!)

- **Requirements**
  - Drupal 8.3+
  - PHP 7.0+
  - Must be installed via composer

WEB PAGE ARCHIVE ENTITIES

- **WebPageArchive** configuration entities contain settings for distinct jobs.
  - Entity type: *web_page_archive*
- **WebPageArchiveRun** content entities contain the actual capture results.
  - Entity type: *web_page_archive_run*
  - Each configuration entity is mapped to a content entity.
  - Each job run is stored as a separate revision.

CAPTURE UTILITIES

- **Capture Utility** plugins are responsible for performing different types of captures.
  - Implement *CaptureUtilityInterface*
  - Extend *CaptureUtilityBase*
  - Use annotation-based plugin discovery.

- **Configurable Captured Utilities** provide configuration forms for adding additional settings.
  - Implement Configurable*CaptureUtilityInterface*
  - Extend *ConfigurableCaptureUtilityBase*

CAPTURE UTILITY STRUCTURE

```
/**
 * Skeleton capture utility, useful for creating new plugins.
 *
 * @CaptureUtility(
 *   id = "wpa_skeleton_capture",
 *   label = @Translation("Skeleton capture utility", context = "Web Page Archive"),
 *   description = @Translation("Illustrates how capture utilities work", context = "Web Page Archive")
 * )
 */
class SkeletonCaptureUtility extends ConfigurableCaptureUtilityBase {
  public function capture(array $data = []) {}
  public function getResponse() {}
  public function defaultConfiguration() {}
  public function buildConfigurationForm(array &$form, FormStateInterface $form_state) {}
  public function submitConfigurationForm(array &$form, FormStateInterface $form_state) {}
  public function cleanupRevision($revision_id) {}
}
```

CAPTURE RESPONSES

- Captured data is stored and rendered using **Capture Response** objects.
  - Implement *CaptureResponseInterface*
  - Extend *CaptureResponseBase*

- ***UriCaptureResponse*** stores path to file on server.
  - Response is rendered as path to file on system.
  - This should typically be extended for particular use cases.

CAPTURE RESPONSES

- ***HtmlCaptureResponse***
  - extends *UriCaptureResponse*.
  - Preview mode shows captured URL.
  - When preview link is clicked, a modal opens containing the captured HTML formatted with Fast Syntax Highlighter.

- ***ScreenshotCaptureResponse***
  - extends *UriCaptureResponse*.
  - Preview mode shows small thumbnail of captured URL.
  - When preview link is clicked, a modal opens displaying a larger screenshot in a modal.

CAPTURE RESPONSE STRUCTURE

```
/**
 * URI capture response.
 */
class UriCaptureResponse extends CaptureResponseBase {
  public function __construct($content, $capture_url) {}
  public function getCaptureSize() {}
  public function renderable(array $options = []) {}
  public function cleanupRevision($revision_id) {}
}
```

SCREENSHOT CAPTURE UTILITY

- **What it does:**
  - Captures Screenshots of URLs.

- **How it works:**
  - Currently uses PhantomJS to facilitate screenshot capturing.
  - Currently working on a new version based on headless chrome. (more on this later)
  - Responses stored as *ScreenshotCaptureResponse* object.

- **Where it lives:**
  - *wpa_screenshot_capture* submodule

HTML CAPTURE UTILITY

- **What it does:**
  - Captures raw HTML of URLs.
- **How it works:**
  - Uses Guzzle, which is part of Drupal Core, to capture raw HTML.
  - Responses stored as *HtmlCaptureResponse* object.
- **Where it lives:**
  - *wpa_html_capture* submodule

SKELETON CAPTURE UTILITY

- **What it does:**
  - Absolutely nothing.
- **Why does it exist:**
  - Prototype for how to build other capture utilities.
  - Can be used to test surrounding behavior without having to perform actual captures.
- **Where it lives:**
  - *wpa_skeleton_capture* submodule

WEB PAGE TEST CAPTURE UTILITY

- **What it does:**
  - Captures web performance test results.
- **How it works:**
  - Starts and captures performance test results from webpagetest.org.
  - While this module can presently capture data, we don't have a way of actually showing this data yet.
  - Responses presently stored as *UriCaptureResponse* object.
- **Where it lives:**
  - (Experimental) Performance Budget module: https://www.drupal.org/project/performance_budget

CONFIGURATION CAPTURE UTILITY

- **What it does:**
  - Captures Drupal's configuration settings.
- **How it works:**
  - Uses Drupal's configuration management services for monitoring changes to config settings.
  - Responses stored as *UriCaptureResponse* object.
- **Where it lives:**
  - (Experimental) Configuration Archive module: https://www.drupal.org/project/configuration_archive

OTHER PROPOSED CAPTURE UTILITIES

- Rendered DOM Capture Utility
- Google Analytics Threshold Monitor
- Composer Package Update Monitor
- Security Scan Utility

CURRENT PROJECT STATUS: BETA 1

- Reliably captures screenshots and HTML.
- Captures both manually and automatically.
- Crontab-based scheduling.
- Parses XML sitemaps.
- Honors robots.txt restrictions.
- Test coverage for most functionality.
- Basic garbage collection.
- Views-driven user interface.

1.0 RELEASE PLAN

- 1.0 Release Plan:
  - https://www.drupal.org/node/2913852
- Some Major Issues:
  - #2908430: Global Configuration Settings
  - #2901556: Headless Chrome Screenshot Capture Utility
  - #2903384: Per-plugin permissions

FUTURE ROADMAP

- Roadmap:
  - https://www.drupal.org/node/2916976
- Some Upcoming Issues:
  - #2894273: Snapshot diffs
  - #2908636: Inject javascript into session
  - #2922937: Don't save duplicate captures
  - #2896502: Run entity performance on large sitemaps

HOW YOU CAN HELP

- Project issue queue
- Submit patches or pull requests
- Review code
- Report bugs
- Submit feature requests
- Write tests
- Help improve documentation
- Help improve UX

DEMONSTRATION

rackspace®

WEB PAGE ARCHIVE

- https://www.drupal.org/project/issues/web_page_archive
- https://github.com/WidgetsBurritos/web_page_archive

David Stinemetze
Twitter: @davidstinemetze
Drupal.org/GitHub: @WidgetsBurritos