

WikiWorks

# GRC-ATF Parsoid config

2024-09-26

---

Thursday 26 September 2024 / 4.75 hrs

Monday 30 September 2024 / 1 hr

Tuesday 01 October 2024 /

## ATTENDEES

Gregory Rundlett

## AGENDA

### Top Issues

- Parsoid configuration
- Visual Editor / Rich Text support for Structured Discussions

## NOTES

The problem is two-fold:

- The configuration of Parsoid has not been adequately defined for various versions of MediaWiki (especially MW1.39) throughout the transition of JavaScript Parsoid, REST API, Parsoid-PHP
- The unmaintained Flow extension may not be compatible with certain point-in-time releases of MediaWiki and/or Visual Editor

## ACTION ITEMS

1. **Reproduce Flow (Structured Discussions) edit problem**
2. Review configuration of Parsoid

## THIS WEEK'S AGENDA

Create Talk page edits

Look at documentation and configuration of Flow (Structured Discussions)

### Error Details

It's important to note that once you modify anything, you are going to see multiple other errors; each triggered for a particular code execution path. IOW, you can only test one change in isolation and probably need a much more rigorous debugging approach to inspect the details of the environment during execution.

**flow-error-parsoid-failure** is a message key that gets referenced in `/opt/htdocs/mediawiki/extensions/Flow/extension.json`

The only file that uses it (aside from the translation files) is `extensions/Flow/modules/editor/editors/visualeditor/mw.flow.ve.Target.js`

```
return new mw.Api().post({
    action: 'flow-parsoid-utils',
    from: fromFormat,
    to: toFormat,
    content: content,
    title: mw.config.get('wgPageName')
})
```

This code makes an API call to convert content from one format to another and handles the response by either returning the converted content or an error message.

Our tests show that we can provoke this error by trying to save rich content, or even by switching the editor from Rich to wikitext.

You might see the message: Unable to transfer content: Error contacting the server for conversion between wikitext and HTML. Please check your Internet connection or try again later if the problem persists. If you still get this error please file a bug

```
{  
  
  "error": {  
  
    "code": "flow-error-process-wikitext",  
  
    "info": "An error has occurred while processing HTML/wikitext conversion.",  
  
    "detail": "Conversion from 'html' to 'wikitext' was requested, but core's Parser only supports  
'wikitext' to 'html' conversion",  
  
    "**": "See https://meza.freephile.org/demo/api.php for API usage. Subscribe to the  
mediawiki-api-announce mailing list at  
&lt;https://lists.wikimedia.org/postorius/lists/mediawiki-api-announce.lists.wikimedia.org/&gt; for  
notice of API deprecations and breaking changes."  
  
  }  
  
}
```

## Test Environment

In our test environment (meza.freephile.org/demo) the versions are as follows:

Component	Version	Git hash	Date
Parsoid	2.0.0		
Flow	1.2.0	a843aa5	02:09, 23 September 2024
Visual Editor	0.1.2	c40e7f8	02:11, 23 September 2024

## Docs

[Docs for Flow](#) (Structured Discussions) say to ensure that wgVirtualRestConfig is set

## Configuration

```
// Configure Flow with VE on MW 1.39
wfLoadExtension( "Flow" );

// Permit at least one "group" to create Flow boards
$wgGroupPermissions['user']['flow-create-board'] = true;

// Set the default content format for Flow
$wgFlowContentFormat = 'html';

// Configure Content Model for namespaces using Flow
$wgNamespaceContentModels[NS_TALK] = 'flow-board';
$wgNamespaceContentModels[NS_USER_TALK] = 'flow-board';

// Configure "subpages" feature for the same namespaces
$wgNamespacesWithSubpages[NS_TALK] = true;
$wgNamespacesWithSubpages[NS_USER_TALK] = true;
```

```

// Flow needs Parsoid if using VE, and with MW 1.39 Flow still ignores
zero-config Parsoid

wfLoadExtension( 'Parsoid', "$IP/vendor/wikimedia/parsoid/extension.json"
);

// Flow ignores RESTBase, but we did not test or document this setting
// $wmgUseRestbaseVRS= false;

$wgVirtualRestConfig['modules']['parsoid'] = [
    'url' => "http://localhost:8080/\$wikiId/rest.php", //
http://localhost/demo/rest.php

// server and port variables would be better here, but other Meza roles
are not synchronized

// 'url' => "$wgServer/$wikiId/rest.php", //
https://meza.freephile.org/demo/rest.php

// 'domain' => $wikiId,
    'prefix' => $wikiId,

// Forward cookies is dangerous for non-private wikis
// 'forwardCookies' => true
];

```

FlowContentFormat

### Code Changes

I did try the master branch of Structured Discussions, but that is not compatible (throws an error) with MediaWiki 1.39.x

I tried the code patch that has been cited in various discussions to modify extensions/Flow/includes/Conversion/Utils.php

```
git apply includes/Conversion/patch.diff
```

```
diff --git a/includes/Conversion/Utils.php b/includes/Conversion/Utils.php
```

index 5b41921..29772d2 100644

--- a/includes/Conversion/Utils.php

+++ b/includes/Conversion/Utils.php

@@ -18,7 +18,9 @@

use MediaWiki\MediaWikiServices;

use OutputPage;

use ParserOptions;

+use ParsoidVirtualRESTService;

use RequestContext;

+use RestbaseVirtualRESTService;

use Sanitizer;

use Title;

use VirtualRESTServiceClient;

@@ -63,10 +65,13 @@

```
        if ( $from === 'wikitext' || $from === 'html' ) {
            if ( $to === 'wikitext' || $to === 'html' ) {
                if ( self::isParsoidConfigured() ) {
-                    return self::parsoid( $from, $to, $content, $title );
-                } else {
-                    return self::parser( $from, $to, $content, $title );
+                    try {
+                        return self::parsoid( $from, $to, $content,
+ $title );
+                    } catch ( NoParserException $e ) {
+                        /* fall through, try legacy parser */
+                    }
                }
+                return self::parser( $from, $to, $content, $title );
            } else {
                throw new WikitextException( "Conversion from '$from' to '$to'
was requested, " .
                    "but this is not supported." );
            }
        }
    }
}
```

@@ -294,37 +299,37 @@

\* For backwards compatibility, \$wgFlowParsoid\* variables are used

```

* to specify a Parsoid configuration as a fall back.
*
+ * See ApiParsoidTrait::getVRSObject() in VisualEditor (which should
+ * eventually be upstreamed to core, T261310).
+ *
* @return \VirtualRESTService the VirtualRESTService object to use
* @throws NoParserException When Parsoid/RESTBase is not configured
*/
private static function makeVRSObject() {
-     global $wgVirtualRestConfig, $wgFlowParsoidURL, $wgFlowParsoidPrefix,
+     global $wgFlowParsoidURL, $wgFlowParsoidPrefix,
+         $wgFlowParsoidTimeout, $wgFlowParsoidForwardCookies,
-         $wgFlowParsoidHTTPProxy;
+         $wgFlowParsoidHTTPProxy, $wgVisualEditorParsoidAutoConfig;
+     $context = RequestContext::getMain();

    // the params array to create the service object with
    $params = [];
-     // the VRS class to use; defaults to Parsoid
-     $class = 'ParsoidVirtualRESTService';
-     // the global virtual rest service config object, if any
-     $vrs = $wgVirtualRestConfig;
+     // the VRS class to use, defaults to Parsoid
+     $class = ParsoidVirtualRESTService::class;
+     // The global virtual rest service config object, if any
+     $vrs = $context->getConfig()->get( 'VirtualRestConfig' );
    // HACK: don't use RESTbase because it'll drop data-parsoid, see T115236
-     /*if ( isset( $vrs['modules'] ) && isset( $vrs['modules']['restbase'] ) ) {
+     // @phan-suppress-next-line PhanImpossibleCondition
+     if ( false && isset( $vrs['modules'] ) && isset( $vrs['modules']['restbase'] ) )
    ) {

        // if restbase is available, use it
        $params = $vrs['modules']['restbase'];
-         $params['parsoidCompat'] = false; // backward compatibility
-         $class = 'RestbaseVirtualRESTService';

```

```

-         } else
-         */
-         if ( isset( $vrs['modules'] ) && isset( $vrs['modules']['parsoid'] ) ) {
+             // backward compatibility
+             $params['parsoidCompat'] = false;
+             $class = RestbaseVirtualREStService::class;
+         } elseif ( isset( $vrs['modules'] ) && isset( $vrs['modules']['parsoid'] ) ) {
+             // there's a global parsoid config, use it next
+             $params = $vrs['modules']['parsoid'];
+             $params['restbaseCompat'] = true;
-         } else {
-             // no global modules defined, fall back to old defaults
-             if ( !$wgFlowParsoidURL ) {
-                 throw new NoParserException( 'Flow Parsoid configuration is
unavailable', 'process-wikitext' );
-             }
+         } elseif ( $wgFlowParsoidURL ) {
+             $params = [
+                 'URL' => $wgFlowParsoidURL,
+                 'prefix' => $wgFlowParsoidPrefix,
@@ -332,6 +337,14 @@
+                 'HTTPProxy' => $wgFlowParsoidHTTPProxy,
+                 'forwardCookies' => $wgFlowParsoidForwardCookies
+             ];
+         } elseif ( isset( $wgVisualEditorParsoidAutoConfig ) ) {
+             $params = $vrs['modules']['parsoid'] ?? [];
+             $params['restbaseCompat'] = true;
+             // forward cookies on private wikis
+             $params['forwardCookies'] = !MediaWikiServices::getInstance()
+                 ->getPermissionManager()->isEveryoneAllowed( 'read' );
+         } else {
+             throw new NoParserException( 'Flow Parsoid configuration is
unavailable', 'process-wikitext' );
+         }
+         // merge the global and service-specific params
+         if ( isset( $vrs['global'] ) ) {

```



```

@@ -339,14 +352,12 @@
        }

        // set up cookie forwarding
        // @phan-suppress-next-line PhanTypePossiblyInvalidDimOffset
-        if ( $params['forwardCookies'] &&
-
!MediaWikiServices::getInstance()->getPermissionManager()->isEveryoneAllowed( 'read' )
-
) {
+        if ( $params['forwardCookies'] ) {
            if ( PHP_SAPI === 'cli' ) {
                // From the command line we need to generate a cookie
                $params['forwardCookies'] =
self::generateForwardedCookieForCli();
            } else {
-
                $params['forwardCookies'] =
RequestContext::getMain()->getRequest()->getHeader( 'Cookie' );
+
                $params['forwardCookies'] = $context->getRequest()->getHeader(
'Cookie' );
            }
        } else {
            $params['forwardCookies'] = false;

```

## Basic Handling

- [Flow] Flow\Hooks::initFlowExtension: Warning: \$wgFlowContentFormat was set to 'html', but you do not have Parsoid enabled. Changing \$wgFlowContentFormat to 'wikitext'

This comes from `Utils::isParsoidConfigured()` which calls `self::getVRSObject()` that calls `self::makeVRSObject()`

```

$wgVirtualRestConfig['modules']['parsoid'] = [
//   'url' => "http://localhost/$wikiId/rest.php",
    'URL' => "$wgServer/$wikiId/rest.php",
//   'domain' => $wikiId,
    'prefix' => $wikiId,

```

```
// Forward cookies is necessary for private wikis
    'forwardCookies' => true
];
```

Using all-caps URL parameter to \$wgVirtualRestConfig was deprecated in MediaWiki 1.35 [Called from Flow\Conversion\Utils::makeVRSObject in /opt/htdocs/mediawiki/extensions/Flow/includes/Conversion/Utils.php at line 354]