Parser Filters

- Server-side parser
- Client-side parser
- Filters

When the editor is interacting with blocks, these are stored in memory as data structures comprising a few basic properties and attributes. Upon saving a working post we serialize these data structures into a specific HTML structure and save the resultant string into the `post_content` property of the post in the WordPress database. When we load that post back into the editor we have to make the reverse transformation to build those data structures from the serialized format in HTML.

The process of loading the serialized HTML into the editor is performed by the *block parser*. The formal specification for this transformation is encoded in the parsing expression grammar (PEG) inside the `@wordpress/block-serialization-spec-parser` package. The editor provides a default parser implementation of this grammar but there may be various reasons for replacing that implementation with a custom implementation. We can inject our own custom parser implementation through the appropriate filter.

Server-side parser

Plugins have access to the parser if they want to process posts in their structured form instead of a plain HTML-as-string representation.

Client-side parser

The editor uses the client-side parser while interactively working in a post. The plain HTML-as-string representation is sent to the browser by the backend and then the editor performs the first parse to initialize itself.

Filters

To replace the server-side parser, use the `block_parser_class` filter. The filter transforms the string class name of a parser class. This class is expected to expose a `parse` method.

Example:

```
class EmptyParser {
  public function parse( $post_content ) {
    // return an empty document
    return array();
  }
}

function wpdocs_select_empty_parser( $prev_parser_class ) {
    return 'EmptyParser';
}

add_filter( 'block_parser_class', 'wpdocs_select_empty_parser', 10, 1 );
```

> **Note**: At the present time it’s not possible to replace the client-side parser.

□□□ #2
□ □□□ □□□ 2026-10-31 18:11:38 UTC
□ □□□ □□□ 2026-10-31 18:11:39 UTC