

WordPress Core API

1. Устаревшие теги шаблонов не разрешается.
2. Темы не должны содержать WARNING, REQUIRED, RECOMMENDED, и INFO уведомления от плагина 'Theme-Check' насколько возможно. Некоторые допустимые исключения включают следующее:

Warning: Found `base64_encode()`

Warning: Found `base64_decode()`

Warning: Found `fwrite()`

Warning: Found `fopen()`

Warning: Found `fclose()`

Recommended: Found `add_theme_support('custom-header', $args)`

Recommended: Found `add_theme_support('custom-background', $args)`

Info: Found `include_once()`

Info: Found `include()`

Info: Found `require_once()`

Info: Found `require()`

Включая вышеуказанные, редкие исключения могут быть сделаны по усмотрению ревьюера

3. Следующие функции являются обязательными:

`wp_title()`

`wp_head()` - Перед `</head>`

`wp_footer()` - Перед `</body>`

`post_class()`

`body_class()` – (внутри `<body>` или `<html>` тегов)

4. Если в тему включены файлы(части шаблона), они обязаны вызываться с помощью `get_template_part()` или `locate_template()`

5. Именованя файлов и структура должны следовать лучшим практикам WordPress. Например, не используйте `category-header.php`, `custom-header.php` etc. и т.д.

6. Должны быть включены:

Automatic Feed Links

Миниатюры постов (Post Thumbnails)

Комментарии (Comments)

`comments_template()`

`wp_list_comments()` - опционально

`comment_form()`

`wp_enqueue_script('comment-reply')`

7. `$content_width` используется, чтобы назначить максимальную ширину для `media` и является обязательным.

8. Модификация фильтров в `wptexturize()` не допускается.

9. Модификация фильтров в `wpautop` не допускается.

10. По умолчанию CSS классы должны быть в таблицы стилей.

11. Используйте уникальный префикс для всех имен функций, классов, хуков, public/global переменных и записей базы данных, чтобы избежать конфликтов с плагинами и другими темами.
12. Переводимые строки не должны содержать переменные.

Особенности для WordPress

1. Тема должна быть widget ready во всех рекламируемых местах.
2. Использование Timthumb не допускается.
3. Темы обязаны предоставлять поддержку child theme.
4. Функция `wp_nav_menu()` должна быть включена, по крайней мере в одном месте тема для легкого управления меню.
5. Все виджеты WordPress по умолчанию должны быть оформлены/показаны правильно во всех областях виджетов.
6. `index.php` должны быть зарезервированы для стандартного блога "Последние записи".
7. В темы не разрешается добавлять параметры, определяющие количество сообщений для отображения на архивных или страниц категории с помощью глобальной настройки.

WordPress Unit Тест

1. Посты отображаются корректно, без видимых проблем или ошибок.
2. Посты отображаются в правильном порядке.
3. Навигация по странице отображается и работает правильно.
4. Страница результатов поиска отображается верно, с результатами поисковых запросов.
5. Прикрепленные посты являются базовой функцией, тема должна отображать их соответствующим образом.
6. "Read More" ссылка работает должным образом (ссылка на пост вместо "<! - more ->" тега).
7. Если тема поддерживает формат типа записи, отображайте пост в указанном виде.
8. Текст не должен выходить за рамки макета.
9. Тема должна включать "Tag" и "Category" таксономии.
10. Floats очищаются должным образом для перемещаемого элемента (миниатюры изображения) в конце контента поста.
11. Ищите потенциальные проблемы переполнения, длинный неразрывный заголовок для небольшой площади под него.

WordPress ассеты

1. `wp_enqueue_style()` должен быть использован для подключения всех таблиц стилей.
2. `wp_enqueue_script()` должен быть использован для подключения всех JavaScript кодов на страницу.
3. Темы обязаны использовать любую версию JQuery с текущей версией WordPress.
4. Авторы не могут отменить регистрацию версии JQuery по умолчанию и загрузить другую.
5. Если версия CDN библиотеки включена, локальная копия должна быть предоставлена в качестве запасного варианта.

6. Активы должны быть загружены как SSL-friendly где это возможно. Это может быть сделано либо путем проверки `is_ssl()` и установив префикс протокола, или с помощью протокола URIs, через [//example.com/file.js](https://example.com/file.js) вместо <http://example.com/file.js>

Безопасность

Тема WordPress должны использовать следующие встроенные функции для проверки или обработки входящего контента или избежать любого сомнительного контента для вывода:

Для ввода:

`wp_filter_nohtml_kses()`

`wp_filter_kses()`

`wp_kses()`

Для вывода:

`esc_attr()`

`esc_html()`

`esc_js()`

`esc_textarea()`

`esc_url()`

Плагины

Другие плагины должны быть включены с помощью TGM Plugin Activation или аналогичного решения для управления зависимости плагинов. Плагины должны установлены с WordPress.org, частных репозиториях или ZIP файла.

PHP

1. Темы не должны иметь каких-либо PHP уведомлений, предупреждений или ошибок - пожалуйста, разрабатывайте тему с включенным показом ошибок, и включенным `WP_DEBUG`.

2. Не допускаются короткие теги PHP.

3. База данных не должна быть доступна или изменена непосредственно. Должна быть определенная функция, которая может получить необходимые данные. Используйте переменную `$wpdb` и ее методы для взаимодействия с базой данных.

WordPress Core API

1. Устаревшие теги шаблонов не разрешаются.

2. Темы не должны содержать WARNING, REQUIRED, RECOMMENDED, и INFO уведомления от плагина 'Theme-Check' насколько возможно. Некоторые допустимые исключения включают следующее:

Warning: Found `base64_encode()`

Warning: Found `base64_decode()`

Warning: Found `fwrite()`

Warning: Found `fopen()`

Warning: Found `fclose()`

Recommended: Found `add_theme_support('custom-header', $args)`

Recommended: Found add_theme_support('custom-background', \$args)

Info: Found include_once()

Info: Found include()

Info: Found require_once()

Info: Found require()

Включая вышеуказанные, редкие исключения могут быть сделаны по усмотрению ревьюера

3. Следующие функции являются обязательными:

wp_title()

wp_head() - Перед </head>

wp_footer() - Перед </body>

post_class()

body_class() – (внутри <body> или <html> тегов)

4. Если в тему включены файлы(части шаблона), они обязаны вызываться с помощью get_template_part () или locate_template ()

5. Именования файлов и структура должны следовать лучшим практикам WordPress.

Например, не используйте category-header.php, custom-header.php etc. и т.д.

6. Должны быть включены:

Automatic Feed Links

Миниатюры постов (Post Thumbnails)

Комментарии (Comments)

comments_template()

wp_list_comments() - опционально

comment_form()

wp_enqueue_script('comment-reply')

7.\$content_width используется, чтобы назначить максимальную ширину для media и является обязательным.

8. Модификация фильтров в wptexturize () не допускается.

9. Модификация фильтров в wpauto не допускается.

10. По умолчанию CSS классы должны быть в таблицы стилей.

11. Используйте уникальный префикс для всех имен функций, классов, хуков, public/global переменных и записей базы данных, чтобы избежать конфликтов с плагинами и другими темами.

12. Переводимые строки не должны содержать переменные.