

توسعه وردپرس (WP Developers)

قالب‌ها (Themes):

قالب وردپرس به مجموعه‌ای از صفحات و استایل‌هایی گفته می‌شود که برای تعریف ظاهر و نحوه نمایش سایت ساخته شده توسط وردپرس، مورد استفاده قرار می‌گیرند.

• پایه:

- فایل استایل اصلی با شناسه یا کلید.
- پوشه بندی های قالب که شامل root و assets و include و template-parts و languages و غیره می باشند.
- شامل فایل های اصلی شامل سربرگ، پاورقی، آرشیو، سینگل، کلمنت، تمپلیت های اختصاصی، فانکشن و ... که با توابع Template Tags قابل فراخوانی می باشند. (header و footer و sidebar و comments و search و template part و بخش هستند و مابقی صفحات، همچنین می توان بخش های مختلف را به دلفواه ساخت و برای فراخوانی از تابع get_template_part استفاده نمود بقیه توابع پیشفرض خود را دارند مانند get_search_form و comments_template و get_header و get_sidebar و ...)
- پست تایپ ها که 5 مورد پست تایپ پیشفرض Post و Page و Attachment و Revision و Navigation menu را شامل می شوند.
- هایلرکی یا سلسله ساختار که می توان با Conditional Tags و هوک {type}_template یا هوک template_include (با استفاده از توابع get_query_template و get_queried_object و locate_template و *_template و load_template) گسترش داد.
- حلقه های وردپرس به دو صورت ساده و آپبکت هستند که اگر در صفحات هایلرکی وردپرس باشیم می توان از حلقه ساده و بدون نیاز به آپبکت (کوئری) استفاده کرد (همچنین در صورت استفاده از حلقه ساده نیاز به استفاده از تابع the_post نیست) و در صورت لزوم باید از حلقه های آپبکت استفاده نمود. همچنین برای نمایش اطلاعات داخل حلقه باید از توابع Loop Tags و Template Tags و Conditional Tags استفاده کرد. و در آخر از تابع wp_reset_userdata یا wp_reset_query برای پاکسازی کوئری می توان بهره برد. (اگر حلقه تکرار شود یا اصطلاحاً حلقه در حلقه استفاده کرده باشیم باید به جای ریست کوئری به روش های قبل از تابع rewind_posts استفاده کنیم)
- برای لینک دهی به فایل های css و js و url و path می توان از توابع get_template_directory و get_template_directory_uri و get_stylesheet_directory و get_stylesheet_directory_uri و get_theme_file_path و get_theme_file_uri و get_theme_root و get_theme_roots و get_theme_root_uri و get_files و get_parent_theme_file_path و get_parent_theme_file_uri استفاده کرد.

• ساختار:

- شامل صفحات و تمپلیت فایل هایی که می توان طبق ساختار هایلرکی ایجاد کرد و از یکسری توابع مانند get_post_format و get_search_query و get_page_template و wp_get_theme()->get_page_templates و is_page_template و get_attachment_template و غیره بهره برد.
- یکسری توابع خاص توی ساختار html های رعایت شود مثل post_class و class_body و the_ID برای آیدی پست ها و content-{ \$ } برای کلاس های محتوای اصلی.

• توابع:

- توابع اولیه مورد نیاز برای قالب ها شامل: add_menu_page و remove_menu_page و add_submenu_page و remove_submenu_page و add_comments_page و add_pages_page و add_links_page و add_media_page و add_posts_page و add_dashboard_page و add_theme_page و add_plugins_page و add_users_page و add_management_page و add_options_page که همگی درون هوک admin_menu قرار دارند.
- همچنین می توان از هوک after_setup_theme و توابع add_theme_support و remove_theme_support و add_post_type_support و set_post_format و get_post_format و has_post_format و get_post_format_string و غیره بهره برد.
- برای ایجاد بخش نظرات می توان از توابع wp_comment_query (کوئری) و wp_list_comment (تابع) و comments_template (فراخوانی فایل) استفاده کرد.
- برای سایدبار ها می توان از توابع register_sidebar و register_sidebars با استفاده از هوک widgets_init و برای فراخوانی فایل سایدبار از تابع get_sidebar و برای فراخوانی ابزارک سایدبار از تابع dynamic_sidebar با تعیین شرط is_sidebar_active می توان استفاده کرد.
- برای تعریف ویجت باید از هوک widgets_init با تابع register_widget و ساخت کلاس ویجت مورد نظر استفاده کرد. و برای فراخوانی داخل صفحات قالب از تابع the_widget باید استفاده کرد.
- برای ایجاد منو ها باید از هوک init و تابع register_nav_menus و برای فراخوانی از تابع wp_nav_menu استفاده کرد.
- برای صفحه بندی حلقه ها باید از توابع next_posts_link و previous_posts_link و posts_nav_link و wp_link_pages یا متغیر سراسری \$wp_query با تابع paginate_links و the_posts_pagination استفاده کرد. همچنین توابع previous_post_link و next_post_link و wp_link_pages برای صفحه بندی محتوای یک فایل سینگل (<!--nextpage-->) استفاده کرد.
- برای بدست آوردن تصاویر می توان از توابع wp_get_attachment_image و wp_get_attachment_image_src و wp_get_attachment_image_url و get_the_post_thumbnail و the_post_thumbnail استفاده کرد. و همچنین این توابع می توانند مفید واقع شوند که عبارتند از

- the_post_thumbnail و set_post_thumbnail_size و has_post_thumbnail و get_post_thumbnail_id و set_post_thumbnail_size و add_image_size تابع باید استفاده کرد.
 - get_permalink و برای alt از تابع the_title_attribute و برای تعریف ساینز هرید از تابع add_image_size باید استفاده کرد.
 - برای چنانی سازی می توان از هوک after_setup_theme با توابع i18n و load_theme_textdomain و load_child_theme_textdomain به همراه text domain و توابع ترجمه استفاده کرد.
 - برای مملی سازی می توان از توابع i18n و توابع ترجمه استفاده کرد.

• سفارشی سازی:

- برای ایبار سفارشی سازی قالب می توان از کلاس های wp_customize_manager و wp_customize_control و wp_customize_section و wp_customize_panel و wp_customize_setting و از هوک customize_update_nav و customize_preview_nav_mwnu و customize_register و مترهای add_panel و get_panel و remove_panel و add_section و get_section و remove_section و add_setting و get_setting و remove_setting و add_control و get_control و remove_control استفاده کرد.

• امنیت:

- برای رعایت امنیت قالب باید از توابع sanitize_* (ضد عفونی کننده داده ها) و esc_* یا esc_html_* (نمایش داده ها) و wp_kses یا wp_kses_post (برای فیلتر تگ های html) و مترهای برقراری ارتباط با دیتابیس با کلاس wpdb و برای اعتبارسنجی داده ها باید از توابع empty یا isset (از توابع php و برای متغیرها) preg_match یا strlen یا mb_strlen (از توابع php و برای رشته ها) strops یا count یا in_array (از توابع php و برای آرایه ها) is_email و term_exists و username_exists و validate_file و conditional tags و is_* و validate_* و exists_* و eval (از توابع WP) استفاده کرد.
 - همچنین می توان از CSRF نیز بهره برد که از توابع wp_nonce_url و wp_nonce_field و wp_create_nonce و check_admin_referer و check_ajax_referer و wp_verify_nonce پشتیبانی می کند بهره برد.

• خصوصیتی:

- برای بفش قوانین و مقررات می توان از توابع get_privacy_policy_url و the_privacy_policy_link و get_the_privacy_policy_link و wp_add_privacy_policy_content بهره برد.

• نکات:

- بیشترین API های قالب که افزونه ها از آن استفاده می کنند wp_head و wp_body_open و wp_footer و wp_meta و comment_form هستند.
 - برای اعتبارسنجی قالب باید بفش های html و css و js و fedd و rss و xml و seo و speed و optimize استاندارد سازی شوند. همچنین امنیت php و wp هم باید لحاظ گردد.
 - برای ایبار یک صفحه آرشیو برای page ها می توان یک فایل با نام paged.php و همانند سایر صفحات آرشیو طراحی نمود.
 - برای تغییر مقادیر کوثری پیشفرض می توان از تابع get_query_var استفاده کرد.
 - ساینز بندی تصاویر عبارتند از: thumbnail (150*150) و medium (300*300) و medium_large (max-height*768) و large (1024*1024) و full (اصلی) یا یک آرایه array(n*n)
 - برای ایبار پوسته فرزند باید فایل style.css را ایبار و درون شناسه اش از کلمه template که نام پوسته پدر هست را استفاده کنیم و در آفر فایل CSS پوسته پدر را import یا wp_enqueue_style کنیم.
 - از لحاظ امنیتی به جای echo از تابع __() یا __e() استفاده شود.
 - برای بدست آوردن نتایج کوثری جستجو ی توان از تابع get_search_query استفاده کرد.

افزونه ها (Plugins):

افزونه ها بسته های کدی هستند که عملکرد اصلی وردپرس را گسترش می دهند.

• پایه:

- فایل اصلی با شناسه یا کلید همراه با توضیحات GPL لایسنس.
 - پوشه بندی های افزونه که شامل root و assets و include و languages و plugin-name و uninstall و admin و public و غیره می باشند.
 - می تتوان از هوک های اصلی register_activation_hook و register_deactivation_hook و register_uninstall_hook برای فعال فعالسازی، غیر فعالسازی و حذف افزونه و همچنین ایبار فایل uninstall.php با ثابت WP_UNINSTALL_PLUGIN برای عملیات حذف اتوماتیک استفاده کرد.
 - می توان از هوک های پراستفاده add_filters و apply_filter و apply_filters_ref_array و has_filter و current_filter و doing_filter و remove_filter و remove_all_filters و add_action و do_action و do_action_ref_array و has_action و did_action و doing_action و remove_action و remove_all_actions نیز بهره برد
 - برای لینک دهی به فایل های css و js و url و path می توان از توابع include و include_once و require و require_once و dirname(__FILE__) و plugin_basename و plugin_url و plugins_url و plugin_dir_path(__FILE__) و plugin_dir_url(__FILE__) استفاده کرد.

- می توان از توابع isset (برای متغیر ها) و function_exists (برای توابع) و class_exists (برای کلاس ها) و defined (برای ثابت ها) برای چک کردن همنام نبودن یا موجود نبودن جهت جلوگیری از تداخل استفاده کرد.

• امنیت:

- برای رعایت امنیت افزونه باید از توابع current_user_can و is_role (چک کردن نقش و دسترسی) و sanitize_* (فیلتر کردن داده ها) و esc_* یا esc_html_* (نمایش داده ها) و wp_kses_post یا wp_kses (برای فیلتر تگ های html) و متدهای برقراری ارتباط با دیتابیس با کلاس wpdb و برای اعتبارسنجی داده ها باید از توابع isset یا empty (از توابع php و برای متغیرها) strlen یا mb_strlen (از توابع php و برای رشته ها) strpos یا count یا in_array (از توابع php و برای آرایه ها) is_email و term_exists و username_exists و validate_file و conditional tags و is_* و validate_* و exists_* و eval (از توابع wp) استفاده کرد. همچنین می توان از CSRF نیز بهره برد که از توابع wp_nonce_url و wp_nonce_field و wp_create_nonce و check_admin_referer و check_ajax_referer و wp_verify_nonce پشتیبانی می کند بهره برد.

• هوک:

- به سرفصل هوک ها مراجعه شود بصورت کامل توضیح داده شده است.

• خصوصی:

- می توان از توابع wp_page_for_privacy_policy و erase_others_personal_data و export_others_personal_data و manage_privacy_options و هوک های user_request_action_confirmed و wp_privacy_delete_old_export_files و wp_privacy_personal_data_erased و wp_privacy_personal_data_export_file_created و wp_privacy_personal_data_export_file و wp_get_default_privacy_policy_content و the_privacy_policy_link و privacy_policy_url و user_request_action_email_content و user_request_action_description و user_request_action_confirmed_message و user_request_confirmed_email_content و user_request_action_email_subject و user_request_action_email_headers و user_request_confirmed_email_to و user_request_confirmed_email_subject و user_request_confirmed_email_headers و wp_privacy_export_expiration و wp_privacy_additional_user_profile_data و user_request_key_expiration و wp_privacy_personal_data_email_headers و wp_privacy_personal_data_email_content و wp_privacy_personal_data_email_to و wp_privacy_personal_data_email_to و wp_privacy_personal_data_email_subject و wp_privacy_personal_data_exporters و wp_privacy_personal_data_erasure_page و wp_privacy_personal_data_erasers و wp_privacy_exports_url و wp_privacy_exports_dir و wp_privacy_anonymize_data و wp_privacy_personal_data_export_page و user_erasure_complete_email_subject و user_erasure_fulfillment_email_to و user_confirmed_action_email_content و user_confirmed_action_email_headers و user_erasure_complete_email_headers استفاده کرد.

• مدیریت:

- می توان از توابع add_submenu_page و add_menu_page با هوک admin_meu برای ایجاد منوهای مدیریت استفاده کرد. همچنین می توان از توابع remove_menu_page و menu_page_url بهره برد.

• شورتکد:

- می توان از توابع add_shortcode و do_shortcode با هوک init استفاده کرد. همچنین می توان از توابع remove_shortcode و shortcode_exists و has_shortcode و array_change_key_case و shortcode_atts نیز بهره برد.
- شورتکد های پیشفرض وردپرس [caption] و [gallery] و [audio] و [video] و [playlist] و [embed] می باشد.

• تنظیمات:

- می توان از توابع register_setting و unregister_setting و add_settings_section و add_settings_field و settings_fields و do_settings_sections و do_settings_fields و add_settings_error و get_settings_errors و settings_errors با هوک admin_init استفاده کرد.
- همچنین می توان از توابع add_option و get_option و update_option و delete_option و add_site_option و get_site_option و update_site_option و delete_site_option بهره برد.

• متادیتا:

- می توان از توابع add_post_meta و update_post_meta و delete_post_meta و get_post_meta یا get_post_custom و the_meta و get_post_custom_values برای ایجاد متادیتا استفاده کرد.
- می توان از توابع add_meta_box و do_meta_boxes و remove_meta_box با هوک add_meta_boxes برای ایجاد متاباکس استفاده کرد.
- همچنین می توان از توابع wp_slash و stripslashes برای ایجاد امنیت مقادیر متا بهره برد.

• پست تایپ:

- می توان از توابع register_post_type با هوک init استفاده کرد.
- همچنین می توان از توابع is_post_type_archive و post_type_archive_title بهره برد.
- **تاکسونومی:**
 - می توان از توابع register_taxonomy با هوک init استفاده کرد.
 - همچنین می توان از توابع the_terms و wp_tag_cloud و is_taxonomy و has_term و wp_get_split_terms بهره برد.
- **کاربران:**
 - می توان از توابع wp_create_user و wp_insert_user و wp_update_user و wp_delete_user با هوک init استفاده کرد.
 - می توان از توابع add_user_meta و update_user_meta و delete_user_meta و get_user_meta با هوک های show_user_profile و edit_user_profile و personal_options_update و edit_user_profile استفاده کرد.
 - می توان از توابع add_role و remove_role و get_role و add_cap و remove_cap و wp_get_current_user و user_can و current_user_can یا current_user_can_for_blog با هوک init استفاده کرد.
 - همچنین می توان از توابع wp_slash و username_exists و email_exists و wp_generate_password بهره برد.
- **HTTP API**
 - می توان از توابع wp_remote_get و wp_remote_retrieve_body و wp_remote_retrieve_response_code و wp_remote_retrieve_header و wp_remote_retrieve_headers و wp_remote_retrieve_response_message و wp_remote_post و wp_remote_request و wp_remote_head با هوک init استفاده کرد.
 - همچنین می توان از توابع set_transient و delete_transient و get_transient بهره برد.
- **آپدیت:**
 - می توان از توابع wp_localize_script و پارامترهای action و title و callback و nonce و WP_Ajax_Response و wp_send_json و wp_send_json_success و wp_send_json_error استفاده کرد.
- **کرون:**
 - می توان از توابع wp_schedule_event و wp_next_scheduled و wp_unschedule_event و _get_cron_array با هوک cron_schedules یا bl_cron_hook استفاده کرد.
 - همچنین می توان از ثابت DISABLE_WP_CRON بهره برد.
- **بین المللی سازی:**
 - می توان از توابع __ و _e و printf و sprintf و _n و _x و _ex و _n_noop و _nx_noop و number_format_i18n و date_i18n استفاده کرد.
 - همچنین می توان از ثابت WPLANG بهره برد.
- **دایرکتوری:**
 - می توان از توابع home_url و get_home_path و admin_url و site_url و content_url و includes_url و wp_upload_dir و plugins_url و get_admin_url و get_admin_url و get_site_url و network_admin_url و network_site_url و network_home_url ثابت های WP_PLUGIN_URL و WP_PLUGIN_DIR و WP_CONTENT_URL و WP_CONTENT_DIR استفاده کرد.

توابع مشترک (Common APIs):

- توابع مشترک مجموعه ای از توابع، متدها و کلاس های ضروری و مشترک برای توسعه قالب ها و افزونه ها و تغییرات در هسته وردپرس می باشد.
- **تصاویر:**
 - اندازه های پیشفرض تصاویر شامل Thumbnail (150*150) و Medium (300*300) و Medium_large (768*auto) و Large (1024*1024) و Full size (auto*auto) می باشد.
 - برای ایجاد تصاویر و آکشنرها با اندازه های خودکار می توان از توابع wp_get_attachment_image_srcset و wp_calculate_image_srcset و wp_get_attachment_image_sizes و wp_calculate_image_sizes و wp_image_add_srcset_and_sizes و max_srcset_image_width استفاده کرد.
 - روال نگ تصاویر:
 - 1) از 4 ویژگی alt و loading و sizes و srcset باید استفاده کرد.
 - 2) از 2 پاکسازی esc_attr و esc_url باید استفاده کرد.
 - 3)

مروگر بر حسب ساینزهای تعریف شده در ویژگی sizes (wp_get_attachment_image_sizes) برای عرض های مختلف از لیست srcset تصویر را بارگیری می کند.
(4) برای استفاده در بوت استرپ در ساینزهای xs (0 تا 575/98 ساینز چیدر 576px) و sm (576 تا 767/98 ساینز medium_large) و md (768 تا 991/98 ساینز large) و lg (992 تا 1199/98 ساینز full size) و xl (1200 تا آخر ساینز full size) می توان بهره برد.

• پیشفوان:

- می توان از تابع wp_add_dashboard_widget و هوک wp_dashboard_setup یا wp_network_dashboard_setup استفاده کرد.
- برای مرتب سازی بکس های داشبورده می توان از آرایه سراسری wp_meta_boxes بهره برد و همچنین برای حذف بکس ها می توان از تابع remove_meta_box استفاده کرد.
- برای بدست آوردن همه ابزارک های داشبورده می توان از توابع ('dashboard_widget_options') get_option و یا یک ابزارک از wporg_get_dashboard_widget_options استفاده کرد.

• دیتابیس:

- می توان از Options API و Transients API و Metadata API استفاده کرد برای دریافت اطلاعات بیشتر به سرفصل پایگاه داده مراجعه شود بصورت کامل توضیح داده شده است.

• بین المللی سازی:

- می توان از توابع ترجمه i18n (بین المللی سازی) و l10n (معلی سازی) استفاده کرد.
 - پایه (برای متغیرها و رشته ها):
__ و _e (به جای echo)
_x و _ex (یک کلمه با چند معنی مثل سیر که هم صفت است و هم یک واژه)
_n , _nx و _n_noop و _nx_noop (برای جمع و مفرد مثل یک دیرگه و چند دیرگه)
printf و sprintf (اگر متغیری داخل یک رشته باشد):
 - تک ها:
__esc_html , __esc_html_e , __esc_html_x , __esc_attr , __esc_attr_e , __esc_attr_x
 - تاریخ:
number , _format_i18n و date_i18n
 - جاوااسکریپت:
__ , _x و _n و _nx و sprintf
 - و در آخر می توان از ترجمه های mo و po استفاده کرد. و در صورت عدم تطبیق یا اعمال ترجمه ها می توان داخل فایل wp-config.php ثابت WPLANG را با مقدار fa_IR تعیین کرد.

• فایل سیستم:

- می توان از کلاس های WP_Filesystem و WP_Filesystem_Base و WP_Filesystem_Direct و WP_Filesystem_FTPext و WP_Filesystem_SSH2 و WP_Filesystem_ftpsocket و request_filesystem_credentials استفاده کرد.

• متغیرهای سراسری:

- متغیرهایی که خود وردپرس بر می گرداند و در همه جا قابل دسترس هستند و اگر داخل حلقه فراخوانی شوند حاوی اطلاعات پست فعلی که در حال پردازش هست می باشند.
 - مفصوصی حلقه:
post , posts , authordata , currentday , currentmonth , page , pages , multipage , more و numpages
 - مفصوص مروگر:
is_iphone , is_chrome , is_safari , is_NS4 , is_opera , is_macIE , is_winIE , is_gecko , is_lynx , is_IE و is_edge
 - مفصوص وب سرور:
is_apache , is_IIS و is_iis7 و is_nginx
 - مفصوص ورژن:
wp_version , wp_db_version , tinymce_version , manifest_version , required_php_version , required_mysql_version
 - مفصوص مدیریت:
pagenow (همراه با تابع get_current_screen) و post_type , allowedposttags , allowedtags و menu
 - متفرقه:
super_admins , wp_query , wp_rewrite , wp و wpdb و wp_locale و wp_admin_bar و wp_roles و wp_meta_boxes و wp_registered_sidebars , wp_registered_widgets , wp_registered_widget_controls و wp_registered_widget_updates

• متریتا:

- برای بدست آوردن متا دیتا های پست، کامنت، ترم و کاربر می توان از توابع add_metadata و delete_metadata و update_metadata و get_metadata استفاده کرد.

• option:

- می توان از توابع add_option و delete_option و update_option و get_option و add_site_option و delete_site_option و update_site_option و get_site_option استفاده کرد.

• افزونه ها:

- به سرفصل افزونه ها مراجعه شود بصورت کامل توضیح داده شده است.

• HTTP:

- می توان از توابع wp_remote_get و wp_remote_retrieve_body و wp_remote_retrieve_response_code و wp_remote_retrieve_header و wp_remote_retrieve_headers و wp_remote_retrieve_message و wp_remote_post و wp_remote_request و wp_remote_head با هوک init استفاده کرد.

• دکمه های ادیتور:

- می توان از تابع جاوا اسکریپتی QTags.addButton با هوک admin_print_footer_scripts استفاده کرد.

• REST:

- به سرفصل وب اپلیکیشن ها مراجعه شود بصورت کامل توضیح داده شده است.

• بازنویسی:

- می توان از کلاس WP_Rewrite و هوک های root_rewrite_rules و post_rewrite_rules و page_rewrite_rules و date_rewrite_rules و search_rewrite_rules و comments_rewrite_rules و author_rewrite_rules و rewrite_rules_array و generate_rewrite_rules و {\$permastructname}_rewrite_rules همچنین می توان از توابع add_rewrite_tag و add_rewrite_rule و add_rewrite_endpoint و flush_rules و flush_rewrite_rules و generate_rewrite_rules و add_permastruct و add_feed بهره برد.
- در ضمن تماماً قبل و بعد ایجاد تنظیمات بازنویسی از تابع flush_rules استفاده شود.

• تنظیمات:

- می توان از توابع register_setting و unregister_setting و add_settings_section و add_settings_field استفاده کرد.
- همچنین می توان از توابع بهره settings_fields و do_settings_sections و do_settings_fields و add_settings_error و get_settings_errors و submit_button و settings_errors.

• شورنگد:

- می توان از توابع add_shortcode و do_shortcode با هوک init استفاده کرد. همچنین می توان از توابع remove_shortcode و shortcode_exists و has_shortcode و array_change_key_case و shortcode_atts نیز بهره برد.
- شورنگد های پیشفرض وردپرس [caption] و [gallery] و [audio] و [video] و [playlist] و [embed] می باشد.

• سلامت سایت:

- می توان از هوک های site_health_navigation_tabs و site_health_tab_content استفاده کرد.

• قالب ها:

- به سرفصل قالب ها مراجعه شود بصورت کامل توضیح داده شده است.

• Transients:

- می توان از توابع set_transient و set_site_transient و delete_transient و get_transient و get_site_transient و delete_site_transient بهره برد.

• XML-RPC:

- این بخش مربوط به API های قدیمی سایت های بلاگ دهی می شود که در حال حاضر منسوخ شده است لذا برای کسب اطلاعات بیشتر می توان به سایت خود وردپرس مراجعه شود بصورت کامل توضیح داده شده است.

وب اپلیکیشن (Rest API):

- رابط کاربری برنامه ها برای تعامل با سایت وردپرسی، با ارسال و دریافت داده ها به عنوان اشیاء (JavaScript Object Notation) JSON را فراهم می کند. که از دو بخش route (همان روت های http یا آدرس url) و endpoint (همان توابعی که با هر روت اجرا می شوند)
- از وظایف مهم rest می توان به دو عمل ایجاد و حذف روت، چهار عمل اصلی crud فیلد یا متا برای هر روت، اعتبارسنجی به ازای هر روت اشاره کرد.
 - برای غیرفعالسازی rest می توان از هوک های json-enabled و json-jsonp-enabled استفاده کرد.

- روت های پیشفرض وردپرس با آدرس {route name}/wp-json/wp/v2/domain name/ می باشند.
- روت های پیشفرض ووکامرس با آدرس {route name}/wp-json/wc/v3/domain name/ می باشند.
- برای تغییر مسیر پیشفرض rest می توان از تابع get_rest_url با هوک rest_url استفاده کرد.
- برای تغییر مسیر پیشفرض wp-json می توان از هوک های rest_url_prefix و json_url_prefix استفاده کرد.
- برای تغییر مسیر روت های پیشفرض می توان از هوک rest_route_for_{type} (مثلا به جای \$type می توان از post یا term یا غیره) استفاده کرد.
- برای ساخت روت جدید می توان از توابع register_rest_route و callback همراه با پارامترهای آن (\$data) با استفاده از هوک rest_api_init استفاده کرد.
- به جای تعیین نوع متر register_rest_route می توان از کلاس با ثبات های (GET) READBLE و (POST) CREATABLE و (POST,PUT,PATCH) EDITABLE و (DELETE) DELETABLE و (DELETE) ALLMETHODS (GET,POST,PUT,PATCH,DELETE) و به شکل WP_REST_SERVER::READABLE استفاده کرد.
- برای ساخت فیلدهای جدید می توان از توابع register_rest_field و register_meta و show_in_rest و fields_parameter استفاده کرد.
- برای افزودن لینک می توان از توابع WP_REST_Response::get_data() و WP_REST_Response::add_data() استفاده کرد.
- برای فیلتر cct های اختصاصی می توان از هوک های register_post_type_args یا register_taxonomy_args یا غیره استفاده کرد.
- برای اعتبارسنجی روت ها می توان از current_user_can و کلاس WP_Error استفاده کرد. همچنین می توان افزونه jwt را نصب و با ابزار postman به آدرس domain name/wp-json/json/jwt-auth/v1/token رفته و از بخش body یوزر،نیم و پسورد را ست کرده و از بخش header و کلمه کلیدی Bearer-فاصله-<توکن،Token مورد نظر را ارسال کرد.
- در ضمن می توان از هوک rest_authentication_errors (دارای سه نوع پاسخ null یا Boolean یا WP_Error) بهره برد.
- برای ایجاد درخواست های داخلی همراه با آبجکت می توان از توابع rest_do_request و WO_REST_Request بهره برد.
- برای ایجاد محدودیت دسترسی می توان از هوک های rest_pre_serve_request و rest_send_cors_headers بهره برد.
- برای موارد دیگر می توان از توابع WP_REST_Server::response_to_data یا rest_get_server()->response_to_data بهره برد.

کدهای استاندارد (Coding Standards):

کافیست از ابزارهای استانداردسازی آنلاین برای بخش های html و css و sass و javascript و jquery و xml و json و php و seo و performance استفاده کرد.

نکات:

- قبل از شروع توسعه وردپرس ثابت های WP_DEBUG و WP_DEBUG_LOG (داخل فولدر content/debug.log ذخیره می شود) و WP_DEBUG_DISPLAY (برای نمایش خطاها داخل تگ های html) و SCRIPT_DEBUG (برای اشکال زدایی css و js) و SAVEQUERES (ذخیره دیباگ در دیتابیس) فعال شوند.
- (' ', 'var_dump(current_filter());', 'all', create_function(' ', ' ')) add_action جهت خالص سازی دیباگ برای همه اکشن ها و فیلتر ها.
- همچنین افزونه های تست و دیباگین شامل: Debug Bar و Debug Bar Console و Debug Bar Monitor و Debug Bar Shortcode و Debug Bar Constants و Log و Debug Bar Remote و Debug Bar Transients و Debug Bar Action and Filter و Debug Bar Cron و Debug Bar Posttype و Deprecated Notices و Debug Bar Script Style و Monster Widgets و Developer و Theme-Check و WordPress Beta Tester و Core Control و Theme Unit و Requests و Test Data نصب گردند.
- برای ایجاد بخش mu-plugins می توان از توابع get_mu_plugins و wp_get_mu_plugins و ثابت های WPMU_PLUGINS_URL و WPMU_PLUGINS_DIR و require ,
- استفاده از تابع version_compare برای ایجاد نسخه بندی قالب و افزونه.