Codebase Architecture Report

Automated documentation generated from codebase AST scanning.

File: `tests/src/Unit/Utility/YamlProcessorTest.php`

`testExtractAndParseValidYaml()`

Return Type: `void`

`testExtractAndParseMarkdownYaml()`

Return Type: `void`

`testExtractAndParseInvalidYaml()`

Return Type: `void`

`testNormalizeCleanYaml()`

Return Type: `void`

`testNormalizeInvalidMarkdown()`

Return Type: `void`

File: `tests/src/Unit/Form/SettingsFormTest.php`

`setUp()`

Description: {@inheritdoc}

Return Type: `void`

Property Services Used: `configFactoryMock`, `form`

`testGetFormId()`

Description: Tests the getFormId() method.

Return Type: `void`

Property Services Used: `form`

`testBuildForm()`

Description: Tests the buildForm() method.

Return Type: `void`

Property Services Used: `form`, `configMock`

`testSubmitForm()`

Description: Tests the submitForm() method.

Return Type: `void`

Property Services Used: `form`, `configMock`

File: `tests/src/Unit/Controller/RecipeChatControllerTest.php`

`setUp()`

Description: {@inheritdoc}

Return Type: `void`

`testBuild()`

Return Type: `void`

Property Services Used: `controller`, `storageService`, `formBuilder`

`testCreate()`

Return Type: `void`

File: `tests/src/Unit/Service/YamlServicesTest.php`

`setUp()`

Description: Tests the setup of the YamlServices class.

Return Type: `void`

`testValidateYaml()`

Description: Tests the YAML validation method.

Return Type: `void`

Property Services Used: `yamlServices`

`testValidateSchema()`

Return Type: `void`

Parameters:

Name	Type
`$yaml`	`string`
`$expected_validity`	`bool`
`$expected_errors_contain`	`array`

Property Services Used: `yamlServices`

`schemaValidationDataProvider()`

Description: Provides data for schema validation tests.

Return Type: `array`

File: `tests/src/Unit/Service/RecipeGenerationServiceTest.php`

`isUsable()`

Return Type: `mixed`

Parameters:

Name	Type
`$type`	`mixed`

`getConfiguredModels()`

Return Type: `mixed`

Parameters:

Name	Type
`$type`	`mixed`

`chat()`

Return Type: `mixed`

Parameters:

Name	Type
`$input`	`mixed`
`$model_id`	`mixed`

`getNormalized()`

Return Type: `mixed`

`getText()`

Return Type: `mixed`

`setUp()`

Description: {@inheritdoc}

Return Type: `void`

`testGetProviderOptions()`

Description: Tests retrieving provider options.

Return Type: `void`

Property Services Used: `generationService`, `aiProviderManagerMock`

`testGetModelOptionsForProvider()`

Description: Tests retrieving model options for a specific provider.

Return Type: `void`

Property Services Used: `generationService`, `aiProviderManagerMock`

`testGenerateRecipe()`

Description: Tests generating a recipe YAML.

Return Type: `void`

Property Services Used: `generationService`, `aiProviderManagerMock`

`testExplainRecipe()`

Description: Tests generating a recipe explanation.

Return Type: `void`

Property Services Used: `generationService`, `aiProviderManagerMock`

`testRefineRecipe()`

Description: Tests refining a recipe.

Return Type: `void`

Property Services Used: `generationService`, `aiProviderManagerMock`

`setupConfigAndContextMocks()`

Description: Helper to set up the Config and Context mocks.

Return Type: `void`

Property Services Used: `configFactoryMock`, `contextExtractorMock`

`setupLlmResponseMock()`

Description: Helper to mock the deep chain of LLM Dummy objects dynamically.

Return Type: `MockObject`

Parameters:

Name	Type
`$expectedText`	`string`

File: `tests/src/Unit/Service/ContextExtractorTest.php`

`setUp()`

Description: Testing setup for the ContextExtractorTest.

Return Type: `void`

`testGetSiteContextMap()`

Description: Tests the extraction of site configuration into a JSON map.

Return Type: `void`

Property Services Used: `configFactory`, `entityTypeBundleInfo`, `entityTypeManager`, `contextExtractor`

`testGetModuleDependenciesStatusInstalled()`

Description: Tests tracking module status when dependencies are already installed.

Return Type: `void`

Property Services Used: `configFactory`, `contextExtractor`

`testGetModuleDependenciesStatusMissingButValid()`

Description: Tests tracking module status for an uninstalled module exists on drupal.

Return Type: `void`

Property Services Used: `configFactory`, `contextExtractor`, `httpClient`

`testGetModuleDependenciesStatusInvalidWithSuggestion()`

Description: Tests tracking module status when a module does not exist on drupal.org.

Return Type: `void`

Property Services Used: `configFactory`, `contextExtractor`, `httpClient`

File: `tests/src/Functional/Form/ArenaFormTest.php`

`setUp()`

Description: {@inheritdoc}

Return Type: `void`

`testArenaFormRenders()`

Description: Tests that the form loads correctly and core elements exist.

Return Type: `void`

`testAddColumnFallback()`

Description: Tests the 'Add another model' button functionality. In a BrowserTestBase (which lacks JavaScript), clicking an #ajax button falls back to a standard form POST, allowing us to verify the form state updates correctly and rebuilds the new column.

Return Type: `void`

File: `tests/src/Functional/Form/RecipeChatFormTest.php`

`setUp()`

Description: {@inheritdoc}

Return Type: `void`

`testChatDashboardLayoutRenders()`

Description: Tests the core layout and rendering of the chat dashboard.

Return Type: `void`

`testEmptyPromptTriggersInlineError()`

Description: Tests that submitting an empty prompt triggers the inline error toast.

Return Type: `void`

File: `tests/src/Functional/Form/BenchmarkDashboardFormTest.php`

`setUp()`

Description: {@inheritdoc}

Return Type: `void`

`testDashboardRendersCorrectly()`

Description: Tests that the dashboard renders all primary components.

Return Type: `void`

`testDashboardFiltersSubmit()`

Description: Tests submitting the filter form updates the URL.

Return Type: `void`

File: `tests/src/FunctionalJavascript/Form/ArenaFormTest.php`

`setUp()`

Description: {@inheritdoc}

Return Type: `void`

`testAjaxAddAndRemoveModelColumn()`

Description: Tests the AJAX "Add another model" and "Remove" column functionality.

Return Type: `void`

`testAjaxProviderDropdownUpdatesModel()`

Description: Tests the AJAX provider dropdown triggering the model dropdown update.

Return Type: `void`

File: `tests/src/FunctionalJavascript/Form/RecipeChatFormTest.php`

`setUp()`

Description: {@inheritdoc}

Return Type: `void`

`testAjaxModelDropdown()`

Description: Tests the Model dropdown populates based on Provider selection via AJAX.

Return Type: `void`

File: `tests/src/Kernel/Form/ArenaFormTest.php`

`setUp()`

Description: {@inheritdoc}

Return Type: `void`

Property Services Used: `recipeGenerationServiceMock`, `container`

`testAddModelSubmit()`

Description: Tests adding a model column utilizing FormState and TempStore.

Return Type: `void`

Property Services Used: `container`, `form`

`testSubmitFormSetsBatch()`

Description: Tests the submission logic and verifies the Batch API is triggered.

Return Type: `void`

Property Services Used: `form`

`testFinishBatchSuccess()`

Description: Tests the batch completion callback interacts with Messenger and TempStore.

Return Type: `void`

Property Services Used: `container`

File: `tests/src/Kernel/Form/RecipeChatFormTest.php`

`setUp()`

Description: {@inheritdoc}

Return Type: `void`

Property Services Used: `container`, `storageServiceMock`, `fileSystemMock`, `generationServiceMock`

`testExecuteGenerationSetsFormState()`

Description: Tests the primary prompt submission process.

Return Type: `void`

Property Services Used: `storageServiceMock`, `form`, `yamlServicesMock`, `generationServiceMock`

`testSubmitSaveRecipe()`

Description: Tests the file system saving logic for generated recipes.

Return Type: `void`

Property Services Used: `form`

File: `tests/src/Kernel/Form/BenchmarkDashboardFormTest.php`

`setUp()`

Description: {@inheritdoc}

Return Type: `void`

Property Services Used: `container`, `dashboardServiceMock`

`testSubmitFormRedirectsWithFilters()`

Description: Tests that submitting the form builds.

Return Type: `void`

Property Services Used: `form`

File: `tests/src/Kernel/Service/ArenaServiceTest.php`

`setUp()`

Description: {@inheritdoc}

Return Type: `void`

`testRunPromptWithNoProvider()`

Description: Tests running a prompt when no valid AI Provider is chosen.

Return Type: `void`

Property Services Used: `container`, `yamlServicesMock`

`testRunPromptSuccess()`

Description: Tests a successful pipeline cycle including validation formats.

Return Type: `void`

Property Services Used: `recipeGenerationServiceMock`, `yamlServicesMock`, `container`

`testRunPromptExceptionHandling()`

Description: Tests how the arena handles unexpected provider API runtime exceptions.

Return Type: `void`

Property Services Used: `recipeGenerationServiceMock`, `yamlServicesMock`, `container`

`testSaveToDashboard()`

Description: Tests dashboard storage dispatch with generated parameters.

Return Type: `void`

Property Services Used: `container`, `storageServiceMock`

File: `tests/src/Kernel/Service/RecipeStorageServiceTest.php`

`setUp()`

Description: {@inheritdoc}

Return Type: `void`

Property Services Used: `container`

`testLogGeneration()`

Description: Tests logging a base recipe generation.

Return Type: `void`

Property Services Used: `container`, `storageService`

`testLogVariant()`

Description: Tests logging an edited variant of an existing recipe.

Return Type: `void`

Property Services Used: `container`, `storageService`

`testGetRecentThreads()`

Description: Tests recent threads tracking along with truncation and pagination limits.

Return Type: `void`

Property Services Used: `container`, `storageService`

`testGetThreadMessages()`

Description: Tests separation of user and system thread interactions.

Return Type: `void`

Property Services Used: `container`, `storageService`

`testGetLatestGeneration()`

Description: Tests fetching the latest generation block within an evaluation thread.

Return Type: `void`

Property Services Used: `container`, `storageService`

File: `tests/src/Kernel/Service/BenchmarkDashboardServiceTest.php`

`setUp()`

Description: {@inheritdoc}

Return Type: `void`

Property Services Used: `container`

`testRecordAndLoadBenchmarks()`

Description: Tests recording and loading benchmarks from the database.

Return Type: `void`

Property Services Used: `benchmarkService`

`testGetSummaryMetrics()`

Description: Tests the calculation of summary metrics.

Return Type: `void`

Property Services Used: `benchmarkService`

`testGetOptions()`

Description: Tests extracting and filtering provider and model options.

Return Type: `void`

Property Services Used: `benchmarkService`

`testAssessRecipeYaml()`

Description: Tests YAML validation and schema checking logic.

Return Type: `void`

Property Services Used: `benchmarkService`

File: `src/Utility/YamlProcessor.php`

`normalize()`

Description: Cleans a response down to YAML text when possible.

Return Type: `string`

Parameters:

Name	Type
`$response`	`string`

`extractAndParse()`

Description: Attempts to extract and validate YAML from a given string.

Return Type: `mixed`

Parameters:

Name	Type
`$response`	`string`

File: `src/Form/SettingsForm.php`

`getFormId()`

Description: {@inheritdoc}

Return Type: `mixed`

`getEditableConfigNames()`

Description: {@inheritdoc}

Return Type: `mixed`

`buildForm()`

Description: {@inheritdoc}

Return Type: `mixed`

Parameters:

Name	Type
`$form`	`array`
`$form_state`	`FormStateInterface`

`submitForm()`

Description: {@inheritdoc}

Return Type: `mixed`

Parameters:

Name	Type
`$form`	`array`
`$form_state`	`FormStateInterface`

File: `src/Form/ArenaForm.php`

`__construct()`

Return Type: `mixed`

Parameters:

Name	Type
`$arena_service`	`ArenaService`
`$recipe_generation_service`	`RecipeGenerationService`
`$temp_store_factory`	`PrivateTempStoreFactory`

`create()`

Description: {@inheritdoc}

Return Type: `mixed`

Parameters:

Name	Type
`$container`	`ContainerInterface`

Container Services Injected: `ai_recipe_generator.arena_service`, `ai_recipe_generator.recipe_generator`, `tempstore.private`

`getFormId()`

Description: {@inheritdoc}

Return Type: `mixed`

`buildForm()`

Description: {@inheritdoc}

Return Type: `mixed`

Parameters:

Name	Type
`$form`	`array`
`$form_state`	`FormStateInterface`

Property Services Used: `tempStoreFactory`, `recipeGenerationService`

`updateModelDropdownCallback()`

Description: AJAX callback to update the model dropdown based on the selected provider. Uses regex to identify which provider triggered the callback and updates the corresponding model dropdown.

Return Type: `mixed`

Parameters:

Name	Type
`$form`	`array`
`$form_state`	`FormStateInterface`

`addModelSubmit()`

Description: Adds a new column ID to the active array.

Return Type: `mixed`

Parameters:

Name	Type
`$form`	`array`
`$form_state`	`FormStateInterface`

Property Services Used: `tempStoreFactory`

`removeModelSubmit()`

Description: Removes a specific column ID from the active array.

Return Type: `mixed`

Parameters:

Name	Type
`$form`	`array`
`$form_state`	`FormStateInterface`

Property Services Used: `tempStoreFactory`

`updateModelsWrapperAjaxCallback()`

Description: Universal AJAX callback for adding or removing columns.

Return Type: `mixed`

Parameters:

Name	Type
`$form`	`array`
`$form_state`	`FormStateInterface`

`submitForm()`

Description: Runs the user prompt against a specific provider and model. Returning structured data about the response and its validity.

Return Type: `mixed`

Parameters:

Name	Type
`$form`	`array`
`$form_state`	`FormStateInterface`

Property Services Used: `tempStoreFactory`

`processBatch()`

Description: Processes a single model evaluation in the batch.

Return Type: `mixed`

Parameters:

Name	Type
`$prompt`	`mixed`
`$provider`	`mixed`
`$model`	`mixed`
`$context`	`mixed`

`finishBatch()`

Description: Completion callback for the batch process that saves results to tempstore.

Return Type: `mixed`

Parameters:

Name	Type
`$success`	`mixed`
`$results`	`mixed`
`$operations`	`mixed`

`saveBenchmarkSubmit()`

Description: Saves a benchmark result to the dashboard via Arena Service.

Return Type: `mixed`

Parameters:

Name	Type
`$form`	`array`
`$form_state`	`FormStateInterface`

Property Services Used: `arenaService`

File: `src/Form/RecipeChatForm.php`

`__construct()`

Return Type: `mixed`

Parameters:

Name	Type
`$generation_service`	`RecipeGenerationService`
`$storage_service`	`RecipeStorageService`
`$yaml_services`	`YamlServices`
`$file_system`	`FileSystemInterface`
`$root_path`	`string`

`create()`

Description: {@inheritdoc}

Return Type: `mixed`

Parameters:

Name	Type
`$container`	`ContainerInterface`

Container Services Injected: `ai_recipe_generator.yaml_services`, `ai_recipe_generator.storage_service`, `file_system`, `ai_recipe_generator.recipe_generator`

`getFormId()`

Description: {@inheritdoc}

Return Type: `string`

`buildForm()`

Description: {@inheritdoc}

Return Type: `array`

Parameters:

Name	Type
`$form`	`array`
`$form_state`	`FormStateInterface`

Property Services Used: `generationService`, `storageService`

`submitForm()`

Description: Submit handler for the main recipe generation form.

Return Type: `mixed`

Parameters:

Name	Type
`$form`	`array`
`$form_state`	`FormStateInterface`

`submitRetryPrompt()`

Description: Submit handler for retrying the last prompt.

Return Type: `mixed`

Parameters:

Name	Type
`$form`	`array`
`$form_state`	`FormStateInterface`

`executeGeneration()`

Description: Executes the recipe generation process.

Return Type: `mixed`

Parameters:

Name	Type
`$form_state`	`FormStateInterface`
`$prompt`	`string`

Property Services Used: `generationService`

`submitExplainRecipe()`

Description: Submit handler for explaining the generated recipe.

Return Type: `mixed`

Parameters:

Name	Type
`$form`	`array`
`$form_state`	`FormStateInterface`

Property Services Used: `generationService`

`submitApplyEdits()`

Description: Submit handler for applying edits to the generated recipe.

Return Type: `mixed`

Parameters:

Name	Type
`$form`	`array`
`$form_state`	`FormStateInterface`

Property Services Used: `generationService`

`submitVerify()`

Description: Submit handler for verifying the edited YAML.

Return Type: `mixed`

Parameters:

Name	Type
`$form`	`array`
`$form_state`	`FormStateInterface`

Property Services Used: `yamlServices`

`submitSaveRecipe()`

Description: Saves the generated YAML to a physical folder on the server.

Return Type: `mixed`

Parameters:

Name	Type
`$form`	`array`
`$form_state`	`FormStateInterface`

Property Services Used: `fileSystem`

`submitExportZip()`

Description: Generates a ZIP file on the fly and forces a browser download.

Return Type: `mixed`

Parameters:

Name	Type
`$form`	`array`
`$form_state`	`FormStateInterface`

Property Services Used: `fileSystem`

`submitLoadMoreThreads()`

Description: Submit handler for the "Load More" button in the sidebar.

Return Type: `mixed`

Parameters:

Name	Type
`$form`	`array`
`$form_state`	`FormStateInterface`

`submitLoadThread()`

Description: Submit handler for loading a specific historical thread.

Return Type: `mixed`

Parameters:

Name	Type
`$form`	`array`
`$form_state`	`FormStateInterface`

Property Services Used: `storageService`, `yamlServices`

`submitNewChat()`

Description: Submit handler for starting a new chat.

Return Type: `mixed`

Parameters:

Name	Type
`$form`	`array`
`$form_state`	`FormStateInterface`

`processAndLogState()`

Description: Centralizes Storage to Database.

Return Type: `void`

Parameters:

Name	Type
`$form_state`	`FormStateInterface`
`$yaml_result`	`string`
`$prompt`	`string`
`$action_type`	`string`
`$provider`	`string`
`$model`	`string`
`$latency`	`int`

Property Services Used: `storageService`, `yamlServices`

`clearUserInputs()`

Description: Clears specific user inputs and rebuilds the form.

Return Type: `void`

Parameters:

Name	Type
`$form_state`	`FormStateInterface`

`ajaxRefreshMain()`

Description: Ajax Callback for refreshing the main chat dashboard.

Return Type: `mixed`

Parameters:

Name	Type
`$form`	`array`
`$form_state`	`FormStateInterface`

`ajaxRefreshPreviewPane()`

Description: Ajax Callback for refreshing the preview pane only.

Return Type: `mixed`

Parameters:

Name	Type
`$form`	`array`
`$form_state`	`FormStateInterface`

`ajaxRefreshModel()`

Description: Ajax Callback for refreshing the model dropdown based on selected provider.

Return Type: `mixed`

Parameters:

Name	Type
`$form`	`array`
`$form_state`	`FormStateInterface`

`ajaxRefreshSidebar()`

Description: Ajax Callback for refreshing the sidebar pane only.

Return Type: `mixed`

Parameters:

Name	Type
`$form`	`array`
`$form_state`	`FormStateInterface`

File: `src/Form/BenchmarkDashboardForm.php`

`__construct()`

Description: Constructs a new benchmark dashboard form.

Return Type: `mixed`

Parameters:

Name	Type
`$benchmark_dashboard_service`	`BenchmarkDashboardService`
`$request_stack`	`RequestStack`

`create()`

Description: {@inheritdoc}

Return Type: `mixed`

Parameters:

Name	Type
`$container`	`ContainerInterface`

Container Services Injected: `ai_recipe_generator.benchmark_dashboard`, `request_stack`

`getFormId()`

Description: {@inheritdoc}

Return Type: `mixed`

`buildForm()`

Description: {@inheritdoc}

Return Type: `mixed`

Parameters:

Name	Type
`$form`	`array`
`$form_state`	`FormStateInterface`

Property Services Used: `benchmarkDashboardService`

`submitForm()`

Description: {@inheritdoc}

Return Type: `mixed`

Parameters:

Name	Type
`$form`	`array`
`$form_state`	`FormStateInterface`

`getFilters()`

Description: Collects request query values.

Return Type: `array`

Property Services Used: `currentRequestStack`

`applyFilters()`

Description: Applies filters and sorting to the loaded benchmark rows.

Return Type: `array`

Parameters:

Name	Type
`$rows`	`array`
`$filters`	`array`

`buildMetricCard()`

Description: Builds the dashboard metric card markup.

Return Type: `array`

Parameters:

Name	Type
`$icon_class`	`string`
`$label`	`string`
`$value`	`string`
`$note`	`string`

`formatMilliseconds()`

Description: Formats a latency value.

Return Type: `string`

Parameters:

Name	Type
`$value`	`float`

`formatPercent()`

Description: Formats a percentage value.

Return Type: `string`

Parameters:

Name	Type
`$value`	`float`

`buildTableHeader()`

Description: Builds the sortable table header.

Return Type: `array`

Parameters:

Name	Type
`$filters`	`array`

`buildSortLink()`

Description: Creates a sortable header link.

Return Type: `array`

Parameters:

Name	Type
`$label`	`string`
`$field`	`string`
`$current_sort`	`string`
`$current_order`	`string`
`$filters`	`array`

`buildTableRows()`

Description: Builds the benchmark table rows.

Return Type: `array`

Parameters:

Name	Type
`$rows`	`array`

`buildValueChip()`

Description: Returns a chip markup string for a value.

Return Type: `array`

Parameters:

Name	Type
`$markup`	`string`
`$class`	`string`

`bandClass()`

Description: Returns a banded class name based on the current value.

Return Type: `string`

Parameters:

Name	Type
`$value`	`float`
`$average`	`float`
`$lower_is_better`	`bool`

`accuracyBandClass()`

Description: Returns an accuracy class based on the score.

Return Type: `string`

Parameters:

Name	Type
`$score`	`float`

File: `src/Controller/RecipeChatController.php`

`__construct()`

Return Type: `mixed`

Parameters:

Name	Type
`$storage_service`	`RecipeStorageService`
`$form_builder`	`FormBuilderInterface`

`create()`

Description: {@inheritdoc}

Return Type: `mixed`

Parameters:

Name	Type
`$container`	`ContainerInterface`

Container Services Injected: `ai_recipe_generator.storage_service`, `form_builder`

`build()`

Description: Builds the AI Recipe Dashboard page.

Return Type: `array`

Property Services Used: `storageService`, `formBuilder`

File: `src/Service/ArenaService.php`

`__construct()`

Return Type: `mixed`

Parameters:

Name	Type
`$database`	`Connection`
`$time`	`TimeInterface`
`$ai_provider_manager`	`mixed`
`$recipe_generation_service`	`RecipeGenerationService`
`$storage_service`	`RecipeStorageService`
`$yaml_services`	`YamlServices`

`runPrompt()`

Description: Runs multiple models/providers against the same prompt.

Return Type: `array`

Parameters:

Name	Type
`$prompt`	`string`
`$provider_id`	`string`
`$model_id`	`string`

Property Services Used: `recipeGenerationService`

`formatResult()`

Description: Formats the result of a benchmark run.

Return Type: `array`

Parameters:

Name	Type
`$provider_id`	`string`
`$model_id`	`string`
`$prompt`	`string`
`$yaml_output`	`string`
`$start_time`	`float`

Property Services Used: `yamlServices`

`saveToDashboard()`

Description: Saves the benchmark run data to the dashboard using RecipeStorageService.

Return Type: `int`

Parameters:

Name	Type
`$run_data`	`array`

Property Services Used: `storageService`

File: `src/Service/YamlServices.php`

`validateYaml()`

Description: Validates a given YAML string syntax.

Return Type: `array`

Parameters:

Name	Type
`$yaml_string`	`string`

`validateSchema()`

Description: Validates a recipe's YAML structural schema.

Return Type: `array`

Parameters:

Name	Type
`$yaml_string`	`string`

File: `src/Service/RecipeStorageService.php`

`__construct()`

Description: Constructs a new RecipeStorageService.

Return Type: `mixed`

Parameters:

Name	Type
`$database`	`Connection`
`$time`	`TimeInterface`

`logGeneration()`

Description: Logs a generation interaction to the history table. Returns the newly inserted ID.

Return Type: `int`

Parameters:

Name	Type
`$thread_id`	`string`
`$interaction_type`	`string`
`$provider`	`string`
`$model`	`string`
`$prompt`	`string`
`$response`	`string`
`$latency`	`int`
`$yaml_valid`	`int`
`$schema_compliant`	`int`

Property Services Used: `time`, `database`

`logVariant()`

Description: Logs a user-edited variant of a generated recipe.

Return Type: `int`

Parameters:

Name	Type
`$history_id`	`int`
`$edited_yaml`	`string`
`$yaml_valid`	`int`
`$schema_compliant`	`int`

Property Services Used: `time`, `database`

`getRecentThreads()`

Description: Retrieves a list of recent chat threads with pagination.

Return Type: `array`

Parameters:

Name	Type
`$limit`	`int`
`$offset`	`int`

Property Services Used: `database`

`getThreadMessages()`

Description: Loads chat history for a specific thread to populate the middle pane.

Return Type: `array`

Parameters:

Name	Type
`$thread_id`	`string`

Property Services Used: `database`

`getLatestGeneration()`

Description: Gets the latest generation record for a thread to populate the right pane.

Return Type: `mixed`

Parameters:

Name	Type
`$thread_id`	`string`

Property Services Used: `database`

File: `src/Service/RecipeGenerationService.php`

`__construct()`

Description: Constructs a new RecipeGenerationService object.

Return Type: `mixed`

Parameters:

Name	Type
`$ai_provider_manager`	`PluginManagerInterface`
`$context_extractor`	`ContextExtractor`
`$config_factory`	`ConfigFactoryInterface`

`getProviderOptions()`

Description: Retrieves a list of all providers that support chat operations.

Return Type: `array`

Property Services Used: `aiProviderManager`

`getModelOptionsForProvider()`

Description: Retrieves the models for a specific AI provider.

Return Type: `array`

Parameters:

Name	Type
`$provider_id`	`string`

Property Services Used: `aiProviderManager`

`buildChatInputRecipeYamlGeneration()`

Description: Construction of the ChatInput (System Prompt + Context + User Prompt).

Return Type: `ChatInput`

Parameters:

Name	Type
`$user_prompt`	`string`

Property Services Used: `contextExtractor`, `configFactory`

`buildChatInputRecipeExplanationGeneration()`

Description: Construction of the ChatInput for recipe explanation generation.

Return Type: `ChatInput`

Parameters:

Name	Type
`$yaml_content`	`string`

`buildChatInputRecipeRefineGeneration()`

Description: Construction of the ChatInput for recipe explanation generation.

Return Type: `ChatInput`

Parameters:

Name	Type
`$yaml_content`	`string`
`$user_feedback`	`string`

Property Services Used: `contextExtractor`, `configFactory`

`generateRecipe()`

Description: Generates a Drupal recipe YAML based on a user prompt, provider, and model.

Return Type: `string`

Parameters:

Name	Type
`$user_prompt`	`string`
`$provider_id`	`string`
`$model_id`	`string`

`queryLlm()`

Description: Queries the LLM.

Return Type: `string`

Parameters:

Name	Type
`$provider_id`	`string`
`$model_id`	`string`
`$input`	`ChatInput`

Property Services Used: `aiProviderManager`

`explainRecipe()`

Description: Generates a plain-text explanation of the generated YAML.

Return Type: `string`

Parameters:

Name	Type
`$yaml_content`	`string`
`$provider_id`	`string`
`$model_id`	`string`

`refineRecipe()`

Description: Refines the generated YAML.

Return Type: `string`

Parameters:

Name	Type
`$yaml_content`	`string`
`$user_feedback`	`string`
`$provider_id`	`string`
`$model_id`	`string`

File: `src/Service/BenchmarkDashboardService.php`

`__construct()`

Description: Constructs a new benchmark dashboard service.

Return Type: `mixed`

Parameters:

Name	Type
`$database`	`Connection`

`loadBenchmarks()`

Description: Returns all benchmark rows with derived comparison fields.

Return Type: `array`

Property Services Used: `database`

`normalizeRecipeYaml()`

Description: Normalizes raw model output into clean YAML text when possible.

Return Type: `string`

Parameters:

Name	Type
`$yaml`	`string`

`getSummaryMetrics()`

Description: Builds dashboard summary metrics.

Return Type: `array`

Parameters:

Name	Type
`$rows`	`array`

`getProviderOptions()`

Description: Returns provider options from benchmark history.

Return Type: `array`

Parameters:

Name	Type
`$rows`	`array`

`getModelOptions()`

Description: Returns model options filtered to a provider, when given.

Return Type: `array`

Parameters:

Name	Type
`$rows`	`array`
`$provider`	`?string`

`recordBenchmark()`

Description: Stores a benchmark row for later comparison.

Return Type: `void`

Parameters:

Name	Type
`$provider`	`string`
`$model`	`string`
`$prompt`	`string`
`$response`	`string`
`$latency`	`int`
`$yaml_valid`	`int`
`$schema_compliant`	`int`
`$thread_id`	`string`
`$interaction_type`	`string`

Property Services Used: `database`

`assessRecipeYaml()`

Description: Assesses whether generated YAML is valid and schema compliant.

Return Type: `array`

Parameters:

Name	Type
`$yaml`	`string`

`hasRequiredRecipeKeys()`

Description: Checks whether the parsed YAML looks like a Drupal recipe.

Return Type: `bool`

Parameters:

Name	Type
`$parsed`	`array`

File: `src/Service/ContextExtractor.php`

`__construct()`

Description: Constructs a new ContextExtractor object.

Return Type: `mixed`

Parameters:

Name	Type
`$config_factory`	`ConfigFactoryInterface`
`$http_client`	`ClientInterface`
`$entity_type_manager`	`EntityTypeManagerInterface`
`$entity_type_bundle_info`	`EntityTypeBundleInfoInterface`

`getSiteContextMap()`

Description: Retrieves existing content types and fields as a JSON map.

Return Type: `string`

Property Services Used: `configFactory`, `entityTypeBundleInfo`, `entityTypeManager`

`getModuleDependenciesStatus()`

Description: Parses YAML to extract required modules and checks installation status.

Return Type: `array`

Parameters:

Name	Type
`$yaml_string`	`string`

Property Services Used: `configFactory`, `httpClient`
