ChatGPT

# Async Crawler Function Canvas

This canvas summarizes the main methods of `AdvancedWebCrawler` and links to the relevant sections in the repository and documentation.

- `crawl_with_memory_adaptive_dispatcher` – Configures a memory-adaptive dispatcher with a `RateLimiter` and `CrawlerMonitor`, then calls `crawler.arun_many` to crawl a list of URLs; returns a list of dictionaries containing URL, title, content length, markdown and status [1] .

- `crawl_with_semaphore_dispatcher` – Uses `SemaphoreDispatcher` for simpler concurrency. It also calls `crawler.arun_many` and only collects URL, markdown content, title and status for successful results [2] .

- `crawl_with_llm_analysis` – Checks for OpenAI credentials and defines an `LLMExtractionStrategy` with the `PageSummary` schema and instruction “Provide a brief summary of the page content and its title” [3] . It runs the crawl using this strategy and parses the LLM output into dictionaries containing the URL, raw markdown and analysis data [4] .

- `crawl_and_store_in_supabase` – Calls `crawl_with_llm_analysis` and then invokes `SupabaseHandler.store_crawl_results` to insert each record into the Supabase `pages` table [5] .

---

[1] [2] [3] [4] [5] async_crawler.py
https://github.com/Gaya56/supa-crawl/blob/main/src/crawlers/async_crawler.py