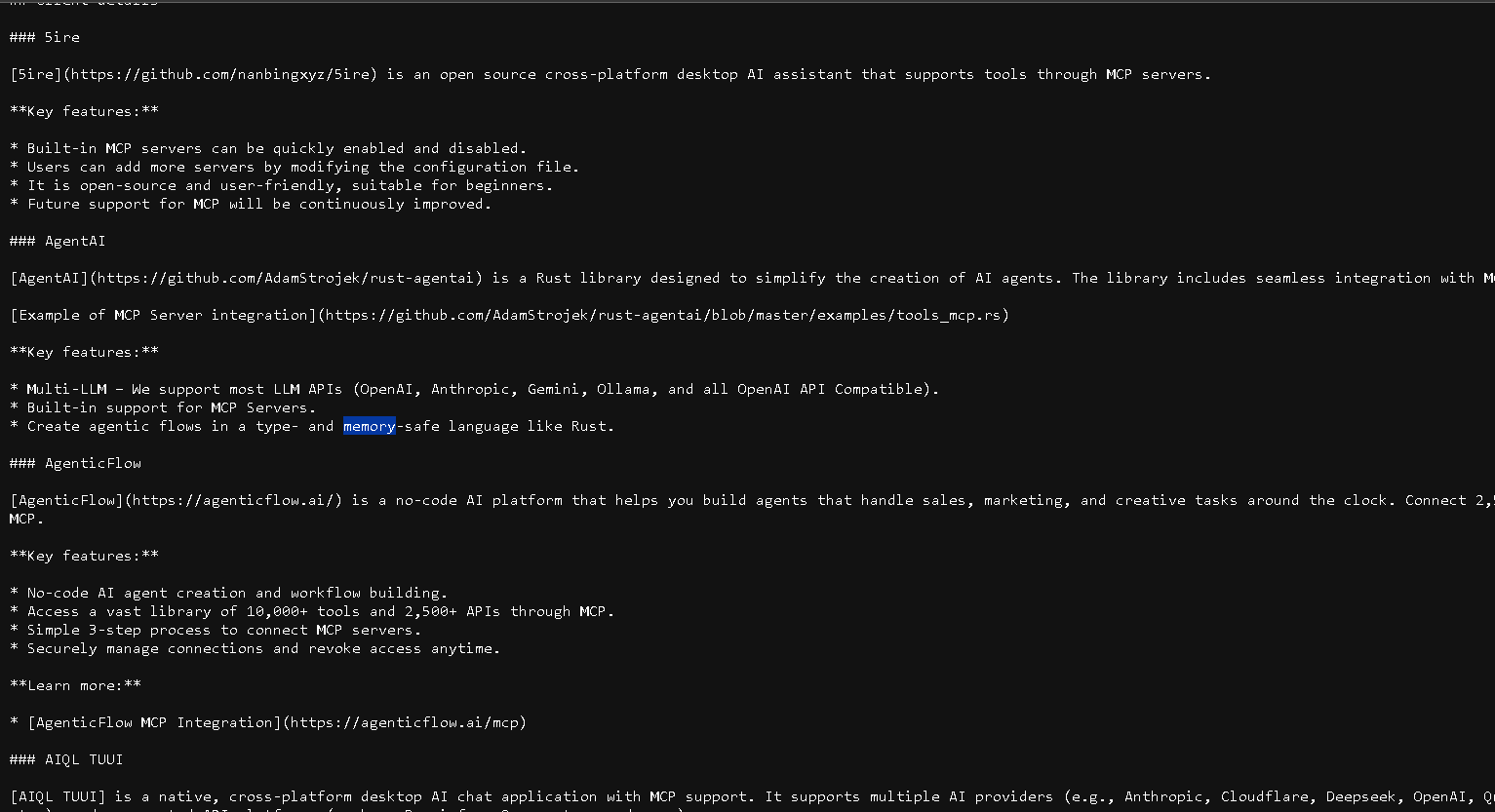
1. Llms.txt is a standard file, which is designed to help LLMs and AI agents to better understand and process website content.
2. So, this file typically is placed in the root directory of a given website and provides a concise summary of the site's most important content and structure in a machine-readable Markdown format.



1. The purpose of this is to give those AI systems, like ChatGPT, LLMs, like Google Gemini, or your application-based AI agents to process the web content more accurately and more efficiently.
2. The format is usually a Markdown file, and the content will include the URLs to the important pages of the website with a brief description of each page content and the purpose of that page. And it may have an optional additional information.
3. The benefits of using llms.txt is it will improve the AI accuracy when extracting the information from the website, So, to downstream that information makes it very, very easy. It also enhances the content discoverability of LLMs with the website content. It also provides obviously better context for understanding the website structure and can even improve the SEO by making the content more accessible for AI-driven search engines.
4. Each website would have their own llms.txt.
5. By MCP: *https://modelcontextprotocol.io/tutorials/building-mcp-with-llms*
6. For langchain, There are two types of LLM.txt: llms.txt and llms-full.txt. The difference is that the llms.txt is only going to include the URLs and a short description while the llms-full.txt is going to be all the information and all the texts of those pages. So, it's going to be a huge file.
7. I'll use the llms.txt, when, for example, I have an AI agent or an MCP server, which has the web scraping tool, like Firecrawl, which enables us to download the content of a website. So, we have the context or all the mapping of the website, and we know exactly which one to download. So, in case, for example, we need to download the content about “LangChain memory”, then we can simply go and give it in the context, and the LLM would be able to choose the tool with the correct URL to download only the memory. So, we'll download, and we'll retrieve only the information that we'll need. It's going to fetch that information in real time, so this is the key point here.
8. Llms-full.txt: we'll chunk it ourself and index it into our vector store, we can get very similar functionality. If we're using LLMs with large context, we can even send the entire thing. Or if those LLMs have context cache, we can cache this information.
9. So, if we want to use the the llms.txt which has only the brief information and the links, then the usage pattern of this is usually by using an agent with a scraping tool and a search tool. So once we do that, we are going to get real time information dynamically, so that's one advantage. However, it's going to take us a bit more time (latency) because we first need to download this page and to fetch it with the URL scraper. We then need to process it with an LLM, then the LLM will tell us what we need to download. So then we need to make another fetch and to download another file and another web page content and to process that, and only then we'll get a result.