LARAVEL AND LLMS

PRACTICAL PHP INTEGRATIONS

LLM hype may be fading, but these tools empower us to solve real cstomer needs more efficiently.

by Alfred Nutile

.001

Laravel and LLMs

Practical PHP Integrations

Alfred Nutile

This book is available at http://leanpub.com/laravel_and_llms

This version was published on 2024-08-18

The author generated this text in part with GPT-3, OpenAI's large-scale language-generation model. Upon generating draft language, the author reviewed, edited, and revised the language to their own liking and takes ultimate responsibility for the content of this publication.

Tweet This Book!

Please help Alfred Nutile by spreading the word about this book on Twitter!

The suggested tweet for this book is:

Excited to learn practical applications of LLMs in my Laravel applications.

The suggested hashtag for this book is #laravelandllms.

Find out what other people are saying about the book by clicking on this link to search for this hashtag on Twitter:

#laravelandllms

Contents

[COMPLETED] Introduction - Embracing the Reality of LLMs in Development

Yes, the hype around LLMs might be fading, but the reality is that these tools offer transformative possibilities for developers. My goal is to provide you with the knowledge and confidence to integrate LLMs into your projects, work effectively with them locally, and add a powerful new tool to your development arsenal. And we won't just skim the surface with "Hello World" examples. We're going to dig deep into the real-world applications that can genuinely enhance your day-to-day work as a developer.

> PHP has a place in this seemingly Python + Ai era.

By the end of this book, I'll equip you with the skills to:

- Work Locally with Ollama: Learn how to save costs and iterate on your ideas effectively by working locally with Ollama. This approach will allow you to refine your concepts without the overhead of cloud services, giving you the flexibility to develop and test in your own environment.
- Test and Mock with Confidence: We'll explore how to set up robust testing and mocking frameworks, so you can deploy your code with the assurance that it will perform reliably in production. This chapter will help you build a foundation of confidence, knowing that your integrations are solid and ready for real-world use.
- Integrate Multiple LLMs Seamlessly: Discover how to write code that's flexible enough to work with any LLM service—whether it's Claude, Ollama, OpenAI, Groq, or another. We'll cover strategies for creating a unified interface, allowing you to switch between services or use multiple providers simultaneously, all within the same codebase.
- Apply Real-World Solutions: I'll share numerous examples of real-world solutions I've implemented (or plan to implement) that demonstrate the practical power of LLMs. These examples will show you how to save time, write efficient code, and tackle complex situations that might have seemed daunting before.
- Master Tools for LLMs: One of the most critical skills you'll learn is how to empower your prompts with tools. We'll dive into how tools can drive a wide range of tasks and automations, making your LLMs not just smart, but practically useful in day-to-day development.
- Make Tools LLM-Agnostic: Finally, we'll cover how to create tools that are independent of any specific LLM. This means you can write your prompts and logic once and use them across multiple LLMs, ensuring that your work remains flexible and future-proof.

By the end of this book, you'll not only have the skills and confidence to move forward with LLM technology but also a clear understanding of how to apply it practically in your work. This journey is about more than just learning a new tool; it's about expanding your capabilities as a developer and seeing the real potential that LLMs bring to solving complex problems efficiently and effectively.

Local LLMs Streamlining Your Development Workflow

Summary: We explore the benefits of working with local LLMs like Ollama, which allow you to develop and test your ideas without incurring cloud costs. This chapter includes practical tips for setting up local environments and integrating them into your workflow. This will include tips and tricks on how to test locally and in CI.

Building the LLM Agnostic Driver

Summary: This chapter covers how to integrate multiple LLMs into your projects, allowing you to switch between services like OpenAI, Claude, and Groq seamlessly. We discuss the advantages of this approach and provide examples of how to implement it effectively.

Prompting Overview before we Dig In

Summary: This chapter delves into the importance of crafting the right prompts and providing the correct context to LLMs. We explore different prompt iterations and demonstrate how to refine them to get the best results, focusing on a real-world example of parsing opt-out emails.

[COMPLETED] Parsing Opt-Out Text

So far, we've covered how to connect to LLMs, pass a request, and get a response.

In this chapter, I'll walk you through a problem I had to solve for a customer that perfectly showcases the power of these tools. The customer receives emails from various opt-out providers requesting to remove individuals from their mailing list. The catch? The data comes in all sorts of formats—some not even plain text, but just HTML!

In the past, I would have written a text parser specific to each domain to extract the Full Name, Phone Number, Address, and Reply-To Address (which, by the way, isn't always in the email header).

But one day, when a new domain started sending us requests, I had a lightbulb moment: why not use an LLM to handle the parsing?

First and foremost, you need to spend some time fine-tuning the prompt that you will passto the llm with the email content. Ultimately, everything boils down to getting the right Prompt (text) and the right Context (text) to the LLM. This principle holds true no matter what problem you're trying to solve.

> It's all about getting the right Prompt (text) and the right Context (text) to the LLM.

With this in mind, using the Driver system we built in a previous chapter, we can plug this into Tinkerwell and start testing our ideas.

```
$results = LlmDriverFacade::driver("ollama")
  ->setFormat("json")
  ->completion($prompt);

$results->content;
```

> Note we are doing `setFormat` since many of these LLMs accept that. We then, depending on the LLM alter the request as needed.

The response we get is:

```
{ "full_name": "John Doe", \n
    "home_address": "123 Main Street, Anytown, USA", \n
    "other_name": null, \n
    "reply_to": "johndoe@example.com", \n
    "valid_user_email": "johndoe@example.com", \n
    "phone": "555-555-5555"\n
}
```

You will notice that in the sample email below, parsing this text is possible BUT to write this for every opt-out vendor, some which had harder to parse emails, it was more work than to do this once.

$email

```
Subject: Request for Deletion of Personal Information

My name is John Doe, and I request that Your Company deletes all of the information \
it has collected about me, whether directly from me, through a third party, or throu
gh a service provider.

The following information is provided as required for validation of my request:

First Name: John
Last Name: Doe
Email: johndoe@example.com
Address: 123 Main Street, Anytown, USA
Phone Number: 555-555-5555

If you need any more information, please let me know as soon as possible. If you can\
not comply with my request—either in whole or in part—please state the reason why yo
u cannot comply. If part of my information is subject to an exception, please delete
 all information that is not subject to an exception. If my request is incomplete, p
lease provide me with specific instructions on how to complete my request.

I have used a consumer identity protection service to send this email, but please re\
spond directly to my personal email address johndoe@example.com.

I am sending this request exercising my rights to my Personal Data under applicable \
privacy laws, including, but not limited to, the General Data Protection Regulation
(“GDPR”) and the California Consumer Privacy Act (“CCPA”)
```

$prompt

```
$prompt = <<<PROMPT
<role>
Help parse opt out emails so we can remove the person from our system or ignore the \
email if it is
not related to opt out.

<task>
Review the email below and then if it is an opt out request pull the data out as def\
ined by the format section.
valid_user_email might be tricky since sometimes the email is generatic and going ba\
ck to the company sending these opt outs.
But if you are confident the email looks like a valid personal or business email add\
 that to valid_user_email.

<format>
JSON FORMAT with key values below, using null if no value for the key. This is valid\
 JSON not markdown.
If there is no data in this about a person just return the JSON below with all field\
s null. No surrounding text.

{
 "full_name": null,
 "home_address": null,
 "other_name": null,
 "reply_to": null,
 "valid_user_email": null,
 "phone": null
}

Only return valid JSON

{$email}
PROMPT;
```

> NOTE this is like mu 4-5 th prompt attempt, “valid JSON” phrase came from the the www.deeplearning.ai trainings.

Then we pass that to the LlmDriver:

```
$results = LlmDriverFacade::driver("claude")->completion($prompt);
$results->content;
```

> Note we will swap out drivers to show how it works.

And we still get:

```
{\n
   "full_name": "John Doe",\n
   "home_address": "123 Main Street, Anytown, USA",\n
   "other_name": null,\n
   "reply_to": null,\n
   "valid_user_email": "johndoe@example.com",\n
   "phone": "555-555-5555"\n
  }
```

As you can see, we use the driver, which can be any LLM—whether it's "ollama", "claude," "openai," or "groq." We then pass it the prompt along with the HTML/Text of the email.

We also enforce a JSON format for the output, which gives us some confidence in the results. From here, we can decide how to proceed:

- Ignore the output if all the fields are null.
- Process the output if the required fields are present.

Also, note that the reply-to address is embedded in the email. Many of these opt-outs require us to reply to the email that sent the request, but as you can see here, we had to adapt to this one.

This approach saved me from manually parsing the text out of these emails and significantly reduced the amount of code, as I previously had parsers for two other vendors.

Later I will talk abouy how to evaluate the accuracy of the results, as well as ideas on testing.

Tools Overview Enhancing LLM Capabilities

Summary: This chapter introduces the concept of tools that enhance the functionality of LLMs. We explore how to build tools that can work with any LLM and how to integrate them into your projects. A real-life example of an event scraping tool is used to demonstrate these concepts.

Making Tools LLM-Agnostic

Summary: Building on the previous chapter, we focus on creating tools that are not tied to any specific LLM. This allows you to switch between LLMs like OpenAI, Ollama, and Claude without rewriting your code. The chapter includes examples of how to structure your tools to be flexible and adaptable.

Web Scraping and Building an Events Database

Summary: Here, we explore how to scrape web data and use LLMs to process the content. The chapter includes examples of tools that can extract relevant information from web pages and convert it into usable formats, such as news articles or recipes

Chaining Tools

Summary: This chapter explores how to iterate over tools in complex workflows. We discuss real-world scenarios where multiple tools need to be chained together to achieve a final result. Examples include processing web content and generating images, highlighting the importance of tool chaining.

Web Searches with LLMs

Summary: A case study of automating web searches using LLMs. The chapter walks through setting up a system that regularly searches the web for relevant content, processes the results, and takes further actions based on predefined criteria.

Marketing Assistant

Summary:

This tool will help us make events via a prompt about helping us with marketing. This will allow the system to then track the status of those events, let us know what is coming up, and ideally automate some of the actions of those events.

Hand Written Notes to Task List

Summary: In this chapter, we explore how to leverage LLMs to transform handwritten notes and voice files into actionable task lists. We'll discuss the process of uploading these inputs, using LLMs to interpret and organize the content, and generating structured tasks that can be integrated into your workflow. Whether you're dealing with meeting notes, brainstorming sessions, or voice memos, this chapter will show you how to automate the process, saving time and ensuring nothing falls through the cracks.

This chapter can demonstrate the practical utility of LLMs in day-to-day productivity, adding another layer of automation to your development toolkit.

RAG and Vector

Summary: In this chapter, we delve into the concept of Retrieval-Augmented Generation (RAG) systems, an architectural approach designed to enhance the effectiveness of LLM applications. We'll cover the foundational aspects of RAG, including how to set it up easily within your projects. However, it's important to understand that while RAG can significantly improve LLM performance, it's not an end in itself. We'll discuss the limitations of RAG and how it should be seen as one tool among many in your development toolkit, rather than a complete solution.

This chapter will provide valuable insights into the benefits and limitations of RAG systems, helping readers make informed decisions about when and how to use this approach in their projects.

SiteMap - Feed to RAG

Summary: In this chapter, we explore how to take a website's sitemap and integrate it into a Retrieval-Augmented Generation (RAG) system. We'll discuss the process of chunking the sitemap content to make it searchable, allowing you to interact with it through search, chat, or even by creating a chatbot. By the end of this chapter, you'll have the knowledge to feed customer content into a RAG system, enabling powerful new ways to engage with the material.

This chapter will guide readers through a practical application of RAG, demonstrating how to use it to enhance customer interactions and content management.

Agents to Evaluate the Results

Summary: This chapter focuses on evaluating the accuracy of LLM outputs. We discuss methods for assessing the quality of the results and provide strategies for improving reliability over time.

Content Verification for Marketing

Summary: A practical example of using LLMs to analyze and verify marketing content. The chapter demonstrates how to vectorize content, create a consistent voice, and ensure that new content meets the required standards. We also discuss how to integrate this into a RAG (Retrieval-Augmented Generation) system

Standards Checking Tool

Summary: This chapter introduces a standards checking tool that evaluates documents against predefined standards. We cover how to set up the tool to work with any LLM, how to interpret the results, and how to integrate this tool into your content management workflow

Final Thoughts

By now, I've shared with you my experiences and insights on integrating LLMs into your day-to-day solutions. We've covered not only the practical aspects of working with these powerful tools but also the strategies for making them a reliable part of your development workflow.

From setting up local environments to save costs, to testing and mocking with confidence, to abstracting LLMs with data objects—these are skills that will empower you to harness the full potential of LLMs. By adopting these techniques, you'll be able to approach complex problems with a new perspective, equipped with tools that simplify and enhance your work.

As we conclude, it's clear that while the initial hype around LLMs might fade, the practical applications and possibilities they offer are here to stay. Whether you're solving age-old problems or tackling new challenges, the integration of LLMs into your toolkit opens up a world of opportunities to innovate and deliver more effective solutions to your clients.

I hope this book has not only provided you with the technical know-how but also inspired you to explore and push the boundaries of what's possible with LLMs. The skills you've gained will serve you well as you continue to evolve in this ever-changing landscape of technology.

Thank you for joining me on this journey, and I look forward to seeing how you apply these insights to create powerful, intelligent solutions in your work.

This expanded conclusion aims to reinforce the key takeaways from the book while encouraging your readers to apply what they've learned in their own projects.