

Charlie Ontiveros

Phone: (323) 492-9493

Email: dev@charlietheprogrammer.com

Location: Sacramento, CA 95824

<https://charlietheprogrammer.com>

<https://github.com/CharlieTheProgrammer>

<https://www.linkedin.com/in/charlie-ontiveros>

Professional Summary

Web application developer looking for front-end position where I can apply my technical and personal skills to deliver high quality web applications.

Work Experience

Meridianlink Inc

SaaS Provider in Credit Lending Space

Technical Support Representative III

May 2011 – April 2018

Costa Mesa, Ca

Overall Duties

- Provided second tier technical support via emails, phone calls, and online chats for cloud-based mortgage loan fulfillment platform.
- Responsible for writing, maintaining, and running automated tests for over 10 APIs during each monthly update. I also conducted functional testing for new features on our web application.
- Served as main technical point of contact for large fintech companies like Black Knight Inc, Credit Plus, Ellie Mae, and Calyx Point. I guided them, and many other companies, through the API integration process for various products Meridianlink offers, like credit reports, tax returns, and verification of employment.
- Handled and resolved top priority issues in a timely manner with companies like Fannie Mae, Equifax, Transunion, Experian, and the IRS.
- I left this job to study software development full time and seek bigger challenges in tech.

Major Accomplishments: Feb. 2016 – April 2018

- Created integration kit with 100+ pages of documentation and industry resources for Meridianlink's SmartAPI XML service.
- Formalized procedures to handle all support aspects of API client integrations.
- Generated report for management on a monthly basis, and when applicable, included quarterly and yearly data on the state of API integrations.
- Led a team-based effort to update and migrate dozens of existing documents from a legacy knowledgebase to a new documentation platform. I also created a style guide to improve documentation.
- Assisted Support Lead with interviewing candidates and provided feedback.

Education

CSU Stanislaus

B.S. Computer Information Systems

May 2006 – May 2011

Golden West Community College

Cisco Networking & Routing

June 2013 – December 2013

Personal Projects

Wunderlist Clone - <https://cocky-poincare-57341e.netlify.com>

June 2019 – Present

- Developing a fully featured clone of Wunderlist; a to-do app that lets users organize tasks into lists, send email, set reminders, and more. #notYourBasicToDoApp.
- **Skills Gained:**
 - How to use Firebase authentication to allow logging in from different OAuth providers and Firestore, a newer version of the Realtime Database, for data storage.
 - How to use React's Context API to share a single instance of information across different components. For example, I'm using the Context API to share one instance of Firebase with multiple components.
 - I learned how build custom animations with CSS using the transitions and keyframe features to improve the UI.
- **Tech Stack:**
 - Javascript (ES6), ReactJs, Firebase, Bootstrap 4 (+ custom CSS), Netlify, Git

Automated Batch Credit Report Processor

June 2019 – June 2019

- Developed fully automated report generation tool that can extract data from 1,000s of credit reports and summarize select data points into an Excel sheet.
- At a high level, this application does the following: Detects when a batch request has been dropped into a Dropbox folder, downloads it, parses it, makes requests to a credit report provider, parses out the results, generates the Excel sheet, then uploads the Excel report to Dropbox and sends out a notification email.
- **Skills Gained:**
 - I learned more about Node's advanced features, specifically event emitters and forking/spawning new processes. The initial process watches the Dropbox folder and then launches additional processes to process the batch.
 - How to create a batch processes and build retry mechanisms for resiliency.
 - Dropbox and AWS SDKs. This application runs in an EC2 instance on AWS.
- **Tech Stack:**
 - Javascript (ES6), Node, AWS

Meetings App - <https://wonderful-tereshkova-782585.netlify.com>

May 2019 – June 2019

- Created an application that allows a user to setup meetings.
- **Skills Gained:**
 - Improved ReactJs skills by getting accustomed with working on a larger number of components. This project has more components than the Hacker News clone.
 - Gained a basic understanding of Firebase's auth capabilities and their Realtime Database service.
- **Tech Stack:**
 - Javascript (ES6), ReactJs, Firebase, Bootstrap 4, Git

Portfolio Site - <https://charlietheprogrammer.com>

May 2019 – May 2019

- Created a portfolio site to demonstrate my projects, blogs, and presentations.
- **Skills Gained:**
 - How to install and configure GatsbyJs to read data from various sources using plugins.
 - How to use ReactJs to model the components GatsbyJs uses to generate static content.
 - I learned and gained a basic understanding of GraphQL by creating GraphQL queries.
 - How to rapidly deploy an application using Netlify.

- **Tech Stack:**
 - Javascript (ES6), ReactJs, GatsbyJs, Bootstrap 4, Git

Hacker News Clone

April 2019 – May 2019

- Created an application that allows a user to search for articles on Hacker News via the Hacker News API. Features include search bar to search for articles, dismissing articles from results, and pagination.
- This project is based on The Road to Learn React book written by Robin Wieruch.
- **Skills Gained:**
 - First major introduction to ReactJs that taught me about how React works, functional components, class components, handling event listeners, updating state, passing props, etc.
- **Tech Stack:**
 - ReactJs, Javascript (ES6)

CreditScale - <https://creditscale.io>

Dec. 2018 – April 2019

- Spent hundreds of hours designing, developing, and deploying production ready JSON REST API intended for commercial purposes. This API cuts integration times down to 1-day vs 7+ days over existing XML API integration.
- This product takes a single point non-REST XML API from Meridianlink and offers a RESTful JSON API. This means integrators only need an API key, credit data provider credentials, and working knowledge of JSON to get up and running. The alternative is needing to understand XML and then use XPath or related tools to parse out this information, which is time consuming.
- Created over 100 integration tests in Postman that covers 6 routes. I also created comprehensive documentation for this API: <https://documenter.getpostman.com/view/3379028/S17xr5ZJ>.
- I used Mocha, a popular JS unit testing tool, to run automated BDD testing on my SQL query, API response, and XML generation modules.
- **Skills Gained**
 - I learned how to make an application production ready by configuring it with appropriate settings and testing through various use cases in a QA environment closely resembling the production environment.
 - I learned how to build a REST API with Node and Express.
 - How to use Postman for API testing and documentation.
 - Parsing XML with SQL Server's OpenXML functionality.
 - Validating user input using JSON Schemas and AJV.
 - Improved code organization and abstraction. This project was large enough that it forced me to think about how to separate concerns between different parts of the application, like the routes, database layer, business logic/controllers, etc.
 - Advanced my usage of agile processes by implementing a monthly release schedule. I reserved 1 week for product design, 2 weeks of sprints, and 1 week for a maintenance sprint. I also started adding estimates to all tasks and reviewed progress daily. I used Trello as a Kanban board.
- **Tech Stack:**
 - Javascript, Node, Express, SQL Server

TaskTrackr – <https://tasktrackr.net>

Feb. 2018 – Nov. 2018

- Designed, developed, and deployed full-stack web application to help users keep track of time spent on projects and associated tasks.
- Design process included breaking up features into tasks on Trello board.

- I deployed the application on Google's Cloud Platform (GCP) Cloud Engine service, setup Apache, configured the DNS for custom domain usage, configured PM2 to automatically restart my application in case errors or system restart occurred. Essentially, I setup everything I needed for this full stack app the hard way.
- I rebuilt most of the application in August 2018 because I was not satisfied with the first iteration, which I initially built as a hybrid web application rather than a true single page application.
- **Skills Gained:**
 - I learned the difference between a hybrid application and a single page application (SPA) and how to build them.
 - Application deployment on Google's Cloud Platform (GCP) using a Compute Engine instance running Ubuntu 16.04, Apache, PM2.
 - Gained an understanding of how Kanban boards contribute to agile and scrum processes by breaking up features into tasks on Trello board.
- **Tech Stack:**
 - Vue.js, Javascript, Node, Express, and SQLite