# **OfferFit Integration Engineer**

# **Take Home Assignment**

In this scenario, you are responsible for integrating with a marketing platform that emits structured JSON events. In total, there are four types of events that we care about capturing:

* **email\_open** - This event is generated whenever a customer actually opens an email to read it.
* **email\_click** - This event is generated whenever a person clicks on a link contained within an email that was sent to them (e.g. they click the “purchase now” link).
* **email\_unsubscribe** - This event is generated whenever a person unsubscribes from a mailing list based on a specific email that they opened and clicked the “unsubscribe” link on.
* **purchase** - This event is generated whenever a person makes a purchase that can be linked to an email offer that was sent to them.

Example JSON payloads with these events can be foundin the additional file.

We'd like you to build a simple integration hub for working with third party systems. You'll stand up a very simple API server that can receive records from a third-party marketing platform, persist them, and then allow queries of the stored data. For our purposes, you can imagine that the events will be sent to your API via a web hook. So you will need to receive the JSON payload and handle the logic from there.

The requirements are as follows:

1. The API should persist the data in any *structure* that you feel is reasonable. Please explain your reasoning in the code comments.
2. The *location* in which you persist the data is flexible - feel free to do whatever is easy for you (i.e. storage in a local file is fine).
3. Have a single endpoint that intakes JSON events, parses them, and then persists them.
4. Have a single endpoint that can return all of the email events for a provided customer ID using a structured, common data model that you define. Optionally, the caller can also provide a time range that determines which data to include.
5. The solution should be in Python, but you can use any library/framework.
6. Please include at least minimal testing for the functionality you build.
7. Please provide a README with instructions on how to run your code locally.

Please do not spend more than 2 hours on this project. Feel free to upload whatever you have at the two hour mark. We do not expect a perfect solution, and we want to be respectful of your time.