ALLOY

Thank you for participating in Alloy's Technical Account Manager work assignment! We are excited to learn more about your approach and thought process regarding this role.

We ask that you return your submission back within three business days. This exercise should take roughly 4-5 hours to complete. Please plan to present your work at your final interview phase. Let us know if you have any clarifying questions at any point!

BACKGROUND

You've been tasked as the tech lead at a bank to build out the integration with Alloy's API in order to minimize fraudulent applications from getting approved and cutting down on any manual reviews from the compliance team.

TASK

What better way to learn more about our API than to build a working integration with Alloy? We'll provide everything you need on the Alloy side to get this working - instructions, API docs and credentials - and all you need to do is create an application form which can submit application details via an API request to Alloy, then handle the response and display the outcome to the applicant. Don't worry: You'll be using sandbox mode, so everything will be simulated.

INSTRUCTIONS

1. Watch our demo video to get acquainted with the basics of how Alloy works. You won't need access to the Alloy dashboard to build your application, since the API response will provide everything you need - but this should provide helpful context for what happens when you submit your API call to Alloy

2. Using the tools of your choice (ex. Express or Flask for backend and React, Svelte, or Vue for frontend) to create a basic web app with frontend and backend components:
   a. The frontend should be composed of a simple application form for the new bank or finance app (creativity is encouraged and will be recognized!). Most modern frameworks have simple commands to spin up a frontend quickly, or you can use a traditional templating language like Pug or Jinga.

   b. Your app must be able to submit the applicant's details to Alloy via our API, process the response, and display a message to the applicant based on the response

   c. You can run the app locally and send a screen recording, or if you prefer, serve it on the web using a service like Render.com or Fly.io. **Hosting on the web is not required for this assignment.**

3. Your form should capture the following details from the applicant (note validation requirements in parentheses - for extra credit, include validation elements on your form front end). As long as you follow the validation rules, you can submit whatever details you like in the form (hi, Jessica Rabbit!)
   a. First Name
   b. Last Name
   c. Address
      i. Line 1
      ii. Line 2
      iii. City
      iv. State (must be a two-letter code, ex. NY, CA, etc.)
      v. Zip/Postal Code
      vi. Country (must be "US" for sake of this assignment)
   d. SSN (must be 9 digits, no dashes)
   e. Email Address
   f. Date of Birth (must be in ISO-8601 format: YYYY-MM-DD)

4. Refer to our API documentation to see how to submit the applicant's details to Alloy:
   a. To get the exact format of the required fields (ex. `name_first`, `name_last`, etc.), you can do a GET to the parameters endpoint: `https://sandbox.alloy.co/v1/parameters/`

   b. For submitting the evaluation, the endpoint you'll POST to is `https://sandbox.alloy.co/v1/evaluations/`

5. For authentication, we will provide a token and secret via secure email. We recommend using Basic authentication for this assignment
   a. The format you'll need is `workflow_token:workflow_secret`

   b. If you're unsure how to base64 encode a string, sites such as GeeksForGeeks have great resources

6. Our API provides informative error handling, so any errors you receive should help you to identify the problem with the integration, but please reach out if you have questions or get stuck

7. By default, in Sandbox mode, a successful API request will return a response including the JSON `{"summary":{"outcome":"Approved"}}`. Upon

receiving this response, your application should display a “Success!” screen announcing that the customer has successfully created an account with your service.

8. To fully complete the assignment, use a test method we refer to as Sandbox Personas to coerce a different response from the API
    a. Submitting an application with the last name Review (so you’d submit an applicant with the name Jessica Review) will result in an “outcome” of “Manual Review”, and the last name Deny will result in an “outcome” of “Deny”.

    b. Your app should show a screen with the message along the lines of “Thanks for submitting your application, we’ll be in touch shortly” for a Manual Review outcome, and “Sorry, your application was not successful” for a Denied outcome. Feel free to have fun with your messaging and user experience here!

THE DELIVERABLES

Once you’ve completed the assignment, we ask that you share the following deliverables with us:

- Your code in a GitHub repository (please follow security best practices and do not include the API token and secret in your repo)

- You’ll be asked to talk through the assignment, your learnings, and any “gotchas” (error handling, validations, etc). You can review your code directly, do a demo of your app in action, create a slide presentation, etc. (whatever you’re most comfortable with)

- If you’re hosting your assignment on the web, share a link to the app so we can try it for ourselves, or if you’re hosting locally, share a screen recording of you submitting three different applications for the Approve, Deny and Manual Review outcomes.