PENFIELD DIGITAL
Johann Siegerstraat 10
1096 BH Amsterdam
KvK-nr: 72803916

INTERVIEW USE CASE

Candidate info	
Name	Juan Carlos Garcia
Email address	jc.gesto@gmail.com
Date	26-10-2021

SFMC Guest login creds	
Username	Penfield_Guest_User_3
Password	See email
Login url	https://mc.s10.exacttarget.com/cloud/?wl=MTAsNTUwMDEzOTksZmU1Zg2

Application	
Role	Salesforce Marketing Cloud Technical Architect / Developer
Goal	To test if the candidate is able to showcase its development skills and solution thinking based on a use case. Candidates will be evaluated on output to decide whether the required knowledge level is present to continue with the next step.

Use case	
Case	Nike shoes
Case description	Nike wants to have a web page where customers can submit a survey (Customer Satisfaction). By submitting the form, the customer will receive a discount of 10%. When the form is successfully submitted, the contact (customer) should enter the Customer Journey with an auto-generated and unique discount voucher code. The email that will be sent out via the journey contains that auto-generated voucher.

1. Prerequisites and assumptions

- The vouchers codes are stored within the Marketing Cloud in a Data Extension (preferably) or in a database of choice (mysql, mongo db etc)
- The web page can be built locally based on custom html/css or within Web Studio as a cloud Page in the Salesforce Marketing Cloud account. Latter is not required and only if the right knowledge is available.
- Journey could contain other channels than email only

2. In scope

- Create one web page with a survey form including field validation, a button to submit data and some styling (front-end)
- Create the API request (Endpoint and payload) to trigger a Salesforce Marketing Cloud journey for this use case. **Note:** it's ok if the API call isn't actually working. As long as you can explain how the token setup and payload should work based on findings in provided documentation.
- Setup a database to store auto-generated voucher codes. **Note**: this could be an external database or somewhere in SF Marketing Cloud. Find the solution that you think fits best.
- Create a diagram of the full process incl. front-end, back-end, data flow and (expected) flow in Marketing Cloud. Make it presentable as you were presenting it to a client.

3. Out of scope

- Validation page of voucher codes for the Stores will be built in the future, it's not included in this User Story.

4. Other remarks

- Expect some technical questions about the architecture and code during the interview.
- All kinds of coding language and databases are allowed, with a preference for the js stack; JavaScript, Nodejs, React, TypeScript or any web application Framework.

5. Deliverables

- A non-technical presentation of the solution incl a flow diagram and short description
    - Present the solution as you would normally do towards a (non-technical) client team
- A working application (webpage) including styling that has to function as intended in all browsers and all devices
    - Make sure to test the components and related processes properly
- Clean, readable, robust and scalable code

6. Appendix A - resources

Tips on resources to use:

- Salesforce Trailhead online learning: https://trailhead.salesforce.com/modules (you can filter on product 'Marketing Cloud' to get all Marketing Cloud modules + you can create an account for free). Interesting modules for this use case:
    - Data Extensions: https://trailhead.salesforce.com/en/content/learn/modules/marketing-cloud-contact-management/learn-about-data-extensions
    - Marketing Cloud API's: https://trailhead.salesforce.com/content/learn/modules/marketing-cloud-apis
    - Journey Builder: https://trailhead.salesforce.com/en/content/learn/modules/journey-builder-campaigns
    - Journey API event entry source: https://trailhead.salesforce.com/en/content/learn/modules/journey-builder-campaigns/prepare-the-entry-source

- Salesforce Marketing Cloud API integration documentation: https://developer.salesforce.com/docs/atlas.en-us.mc-app-development.meta/mc-app-development/create-integration-enhanced.htm. Mainly the steps are:
    - 1. Create a Installed API Package within the SFMC account: https://developer.salesforce.com/docs/atlas.en-us.mc-app-development.meta/mc-app-development/install-packages.htm

    - 2. Add an API component to the package. Pick the component named 'API integration'. With this component you get things like the client id + secret for authentication to get API access

    - 3. For type use the Server-to-Server integration. Concept how it works can be found here: https://developer.salesforce.com/docs/atlas.en-us.mc-app-development.meta/mc-app-development/integration-s2s-client-credentials.htm
      The process is quite straightforward; get auth token > use token in payload > fire API to push data in journey via right API endpoint
    - 4. Setup API payload to retrieve auth key via oAuth 2.0 process: https://developer.salesforce.com/docs/atlas.en-us.mc-app-development.meta/mc-app-development/access-token-s2s.htm
      Hint: token expires after 20 min. So normally you would need to have something in your app to be sure your token won't expire.

    - 5. Once you got the token you can add this in the journey API payload. Also add more attributes and fire the API. Doc to the API endpoint to push data into a journey: https://developer.salesforce.com/docs/atlas.en-us.noversion.mc-apis.meta/mc-apis/postEvent.ht

- When triggering an email after the survey data submit it's recommended to use the API event in a journey (Journey Builder module). When creating a new journey the entry source API event must be selected. In that config screen a Data Extension (= SFMC data table to capture the data) must be selected. After selecting a Data Extension the config can be completed.
    - Tip 1 - create a Data extension: https://help.salesforce.com/articleView?id=sf.mc_es_create_data_extension.htm

    - Tip 2 - API def key: when creating the API event entry source the API definition key is located in the top right which is needed in the API call from your page to inject the data into the journey: