

Customer API design

[Motivation](#)

[Requirements](#)

[Functional Design](#)

[Add customers](#)

[Get all customers](#)

[Get customer by email](#)

[Update a customer](#)

[Delete a customer](#)

[Architecture](#)

Motivation

This document captures requirements, functional design and architecture of customer rest API.

Requirements

Functional

ID	Requirement	Priority
1	Ability to to store a full name including prefix, suffix, surname(first name), middle name, family name.	Must have
2	Store email and phone number	Must have
3	Get all customers.	
4	Get unique customer	
5	Update a customer	
6	Delete a customer	

Non-Functional

ID	Requirement	Priority
1	Operations should have sufficient logging, but consider privacy indication with logging	Must have
2	Operations should leverage telemetry for observability	Must have

3	Provide Helm chart for project deployment. Utilize a local container registry (example) for publishing and retrieving image.	Must have
---	--	-----------

Functional Design

Add customers

This rest endpoint will add a customer. The payload of the request will look like this.

End point	Payload	Response
POST /crm/v1/customers Host: localhost:8080 Content-type application/json	<pre>{ "prefix": "Mr.", "suffix": "Sr.", "firstName": "Rob", "middleName": "A", "lastName": "Smith", "email": "rsmith@gmail.com", "phoneNumber": "8607189900", }</pre>	<pre>{ "message": " Customers added.", "count": "1" }</pre>

API payload input constraints:

- firstName, lastName and phoneNumber are required inputs.
- Each filed have max limit of 100 chars

Get all customers

This rest endpoint will return all customers. The payload of the request will look like this.

End point	Payload	Response
GET /crm/v1/customers Host: localhost:8080 Content-type application/json		<pre>[{ "prefix": "Mr.", "suffix": "Sr.", "firstName": "Rob", "middleName": "A", "lastName": "Smith", "email": "rsmith@gmail.com", "phoneNumber": "8607189900", },]</pre>

Get customer by email

This endpoint will return a customer by email. The payload of the request will look like this.

End point	Payload	Response
GET /crm/v1/customers/{email} Host: localhost:8080 Content-type application/json	<pre>{ "email": "rsmith@gmail.com" }</pre>	<pre>{ "customer": { "prefix": "Mr.", "suffix": "Sr.", "firstName": "Rob", "middleName": "A", "lastName": "Smith", "email": "rsmith@gmail.com", "phoneNumber": "8607189900", }, }</pre>

API payload input constraints:

- Email is required.

Update a customer

This endpoint will update a customer. The payload of the request will look like this.

End point	Payload	Response
POST /crm/v1/customers/{id} Host: localhost:8080 Content-type application/json	<pre>{ "firstName": "Rob New", "lastName": "Smith New", "phoneNumber": "1111111", }</pre>	<pre>{ "message": " Updated" }</pre>

Delete a customer

This endpoint will delete a customer. The payload of the request will look like this.

End point	Payload	Response
DELETE /crm/v1/customers{email} Host: localhost:8080 Content-type application/json		<pre>{ "message": "deleted", }</pre>

Architecture