

## Technical Task – API Specialist

### 1. API contract Definition:

Git location:

<https://github.com/rjavarria/customermanagement.git>

### Yaml file Location:

customermanagement/rjava\_CUST\_PROFILE\_OAS3.0\_v1\_swagger.yaml



```
rjava_CUST_PROFIL
E_OAS3.0_v1_swagger
```

API Interface: `au.com.rjavarria.profile.customer.api.CustomersApi`



CustomersApi.java

## Swagger Hub

Swagger-UI (<http://localhost:9089/swagger-ui.html>)

localhost:9089/swagger-ui.html#!/customer/customerAccountGet

**customer** Show/Hide List Operations Expand Operations

**POST** /customers Create a new customer

**Implementation Notes**  
Create a new customer profile.

**Parameters**

Parameter	Value	Description	Parameter Type	Data Type
customer	<pre>"accountId": "rjavaria", "firstname": "Rajesh", "lastname": "Javaria", "title": "Mr.", "dob": "01/01/1985", "email": "rajesh.javaria@gmail.com", "homeaddress": {   "street": "123 Main St",   "city": "New York",   "state": "NY",   "zip": "10001" }, "officeaddress": {   "street": "456 Main St",   "city": "New York",   "state": "NY",   "zip": "10001" }</pre>	customer	body	Model Example Value <pre>Customer {   accountId (string, optional),   dob (string, optional),   email (string, optional),   firstname (string, optional),   homeaddress (Address, optional),   lastname (string, optional),   officeaddress (Address, optional),   title (string, optional) } Address {   postcode (string, optional),   state (string, optional),   street (string, optional),   suburb (string, optional) }</pre>

Parameter content type:

**Response Messages**

HTTP Status Code	Reason	Response Model	Headers
200	Successfully created a new customer.		
201	Created		
401	Authorization information is missing or invalid		

## 2. **Integration:**

The integration and deployment of APIs can be well managed by API Management platforms like Apigee, Axway etc. APIs can be integrated in Cloud, Hybrid environment or at on-premises by one of these APIs management service.

AWS API Gateway is another easiest way to integrate/deploy micro services exposing REST APIs. AWS API Gateway also provides proxy integration with AWS Lambda, we can simply upload Spring Boot Application to Lambda, and create AWS Lambda function to trigger this Spring -boot application.

aws-serverless-java-container supports API Gateway's proxy integration models for requests and responses makes easy to run Spring-boot applications. This library receives events object from Lambda and translates them to a request object for the spring boot framework. Similarly, it translates responses from the framework into valid return values for API Gateway.

## 3. **Security:**

Customer APIs code which is checked in to Git using Basic Authentication.

APIs security must be approached from the APIs consumption and exposure perspectives. Authentication of trusted Apps can be implemented using API key(client credentials) or Basic Authentication.

Authentication of Untrusted Apps can be implemented using Two-way TLS(MASL) or OpenID connect.

Other Authentication mechanism can be LDAP, Active directory or persisting client credentials in database.

APIs should also be protected from Rate-limiting(throttling) and DDoS attacks.

Authorization that could be Granting access to resources and allowing some certain actions should be implemented using OAuth2.0.

#### 4. **APIs delivered:**

Using Java Spring-boot

#### 5. **Git location:**

<https://github.com/rjavarja/customermanagement.git>

### **Request-Response**

User: rajesh

password: pass123

▶ CreateCustomer

POST localhost:9089/customers/ Params Send

Authorization ● Headers (3) Body ● Pre-request Script Tests

TYPE  
Basic Auth ▼

The authorization header will be automatically generated when you send the request. [Learn more about authorization](#)

Username: rajesh  
Password: pass123  
☒ Show Password

▶ CreateCustomer

POST localhost:9089/customers/ Params Send

Authorization ● Headers (3) Body ● Pre-request Script Tests

	Key	Value	Description	...	Bo
	Authorization	Basic cmFqZXNoOnBhc3MxMjM=			
<input checked="" type="checkbox"/>	Accept	application/json			
<input checked="" type="checkbox"/>	Content-Type	application/json			

► CreateCustomer

POST localhost:9089/customers/ Params

Authorization Headers (3) **Body** Pre-request Script Tests

form-data x-www-form-urlencoded **raw** binary **JSON (application/json)**

```
1 {
2   "accountid": "rjavaria",
3   "firstname": "Rajesh",
4   "lastname": "Javaria",
5   "title": "Mr.",
6   "dob": "01/01/1985",
7   "email": "rajesh.javaria@gmail.com",
8   "homeaddress": {
9     "street": "21, Best Road",
10    "suburb": "Seven Hills",
11    "postcode": 2147,
12    "state": "NSW"
13  },
14  "officeaddress": {
15    "street": "21, Best Road",
16    "suburb": "Seven Hills",
17    "postcode": 2147,
18    "state": "NSW"
19  }
20 }
```