SOAP Banking Service - Detailed ReadyAPI Project Documentation

# 1. Introduction

This document provides a comprehensive breakdown of the ReadyAPI project for the SOAP-based Banking Service. It includes explanations of the WSDL structure, mock service dispatching logic, request/response payloads, and the custom Groovy scripting used for mock services.  
  
The project emulates core banking operations such as:  
- Customer onboarding (AddCustomer)  
- Account creation (AddAccount)  
- Branch registration (AddBranch)  
- KYC information handling (AddKYC)  
  
All services are hosted virtually using ReadyAPI's MockService on local endpoints.

# 2. WSDL Service & Operation Definitions

## AddCustomer

* Request Parameters:
* • name (string)
* • email (string)
* • dob (string)
* • panNumber (string)
* Response Parameters:
* • status (string)
* • customerId (string)

## AddAccount

* Request Parameters:
* • accountId (string)
* • customerId (string)
* • type (string)
* • balance (float)
* Response Parameters:
* • status (string)
* • accountId (string)

## AddBranch

* Request Parameters:
* • branchId (string)
* • name (string)
* • ifsc (string)
* Response Parameters:
* • status (string)
* • message (string)

## AddKYC

* Request Parameters:
* • kycId (string)
* • customerId (string)
* • documentType (string)
* • documentNumber (string)
* Response Parameters:
* • status (string)
* • message (string)

# 3. Mock Service Endpoints

* • http://localhost:8080/mockAddCustomerService
* • http://localhost:8088/mockBankingService
* • http://localhost:8089/mockBankingService

# 4. Dispatching Logic

Dispatch types used per operation:

* AddCustomer: SCRIPT (Groovy logic for validation and dynamic ID)
* AddAccount: SCRIPT (Static response template)
* AddBranch: SEQUENCE (Static response)
* AddKYC: SEQUENCE (Static response)

# 5. AddCustomer Groovy Script

This script validates input, generates customer ID, and returns a SOAP response:

import groovy.util.XmlSlurper  
import groovy.xml.MarkupBuilder  
  
def requestXml = mockRequest.requestContent  
def request = new XmlSlurper().parseText(requestXml)  
  
def name = request.name?.text()  
def email = request.email?.text()  
def dob = request.dob?.text()  
def pan = request.panNumber?.text()  
  
def errors = []  
if (!name || !(name ==~ /^[A-Za-z ]{2,}$/)) errors << "Invalid or missing name"  
if (!email || !(email ==~ /^[\w.-]+@[\w.-]+\.\w{2,}$/)) errors << "Invalid or missing email"  
if (!dob || !(dob ==~ /^\d{4}-\d{2}-\d{2}$/)) errors << "Invalid or missing date of birth"  
if (!pan || !(pan ==~ /^[A-Z]{5}[0-9]{4}[A-Z]{1}$/)) errors << "Invalid or missing PAN number"  
  
def project = mockOperation.mockService.project  
def lastNumberProp = "lastCustomerNumber"  
def lastNumber = project.hasProperty(lastNumberProp) ? project.getPropertyValue(lastNumberProp)?.toInteger() : 100000  
def nextNumber = lastNumber + 1  
def customerNumber = "CUST${nextNumber}"  
project.setPropertyValue(lastNumberProp, nextNumber.toString())  
  
def writer = new StringWriter()  
def xml = new MarkupBuilder(writer)  
xml.'ns2:AddCustomerResponse'("xmlns:ns2": "http://example.com/addcustomer") {  
 if (errors) {  
 status("validationfailed")  
 errors.each { errors(it) }  
 } else {  
 status("success")  
 customerNumber(customerNumber)  
 }  
}  
context.responseContent = writer.toString()  
return "OK"

# 6. Sample Request & Response (AddCustomer)

Request:

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:add="http://example.com/addcustomer">  
 <soapenv:Header/>  
 <soapenv:Body>  
 <add:AddCustomerRequest>  
 <add:name>John Doe</add:name>  
 <add:email>john@example.com</add:email>  
 <add:dob>1990-01-01</add:dob>  
 <add:panNumber>ABCDE1234F</add:panNumber>  
 </add:AddCustomerRequest>  
 </soapenv:Body>  
</soapenv:Envelope>

Response:

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:add="http://example.com/addcustomer">  
 <soapenv:Header/>  
 <soapenv:Body>  
 <add:AddCustomerResponse>  
 <add:status>success</add:status>  
 <add:customerNumber>CUST100001</add:customerNumber>  
 </add:AddCustomerResponse>  
 </soapenv:Body>  
</soapenv:Envelope>