**MPS**

Mzizi Promary School

**Integration Technical Design Document**

**mps-students-system-api technical design**

|  |  |  |
| --- | --- | --- |
| **Project/Service Identifier:** |  | mps students system api |
| **Prepared By:** | Sboniso Gordon Mzizi |
| **Document Version:** | Version 1.0.0 |
| **Published Date:** | 27/04/2024 |

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version Number** | **Revision Date** | **Summary of Changes** | **Modified by** |
| 0.1 | *27/04/2024* | Initial document created | *Sboniso Gordon Mzizi* |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Contributors**

| **Name** | **Title** |
| --- | --- |
| *Sboniso Gordon Mzizi* | *MuleSoft Technical Architect* |
|  |  |
|  |  |
|  |  |

**Table of Contents**

# Introduction

## Document Overview

### Intended Audience

This document is intended to be used by the integration developers, Integration support team, architects, and testers who will be designing, implementing, and maintaining these middleware services. This document is not intended to be used by the API consumers; API documentation is provided on the Anypoint platform.

### Structure

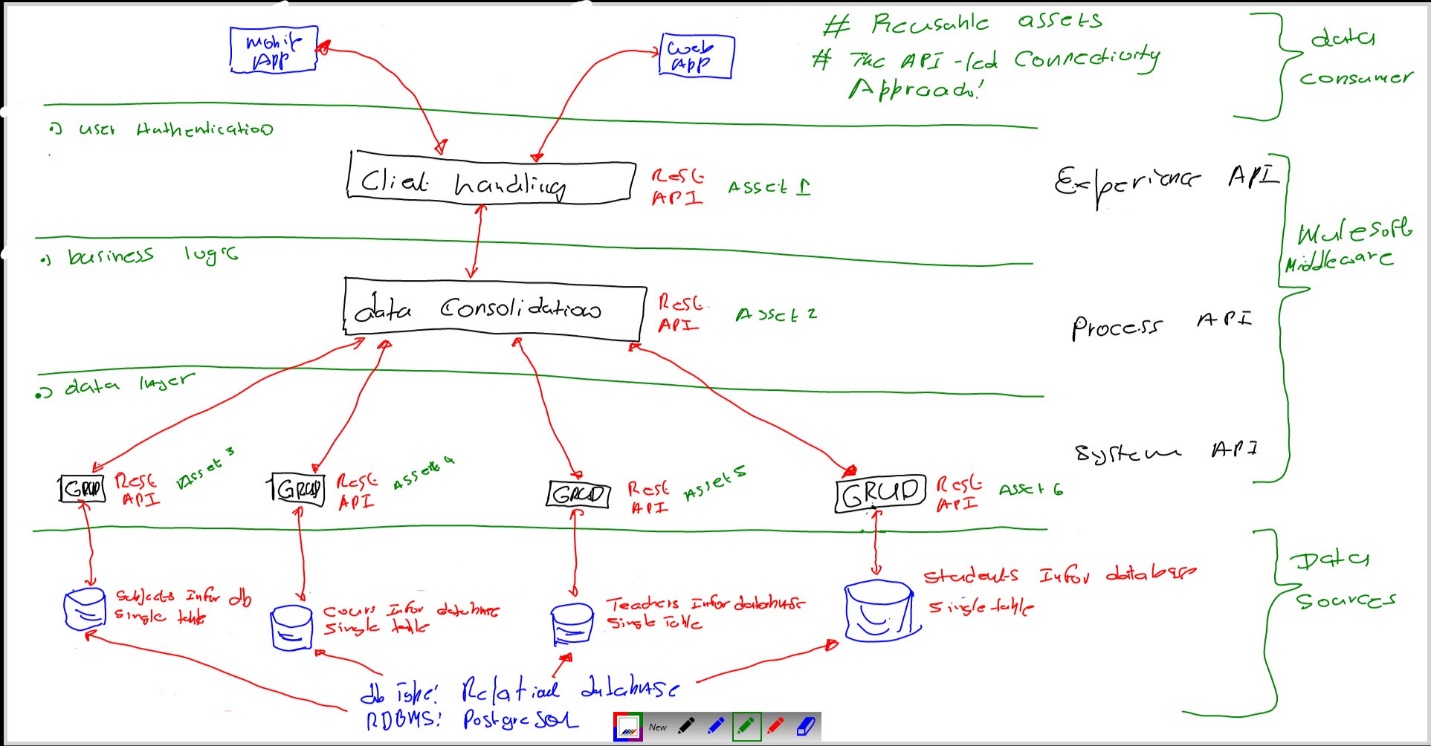
This document contains an overview and depiction of all the services. Document includes detailed designs for each service, including intended API consumers, business rules, technical considerations, and result codes.

### Related Documents

The following list provides links to related documents:

|  |  |  |
| --- | --- | --- |
| Document | Intended Use | Share point Link |
| Solution Architecture Document |  |  |
| Data Mapping document |  |  |

# Technical Architecture



# System APIs High level Specification

## Get student by id : https://mps.co.za/mps-students-system-api/v1/student/{studentId}

This is a GET based API takes studentId as url parameter and respond back with 200 status code, Success status message and json payload containing student information. On failure the API will respond back corresponding error code and message along with description.

1. The system API’s consumes relational database management system for student information.
   1. The mps-students-systems-api retrieve student information from the student RDBMS.
      1. The student RDBMS response with an appropriate response .

1.1.1.1 The mps-students-systems-api make student information available to any process api.

### Input and Output Definitions of API

**Output: Status code -> 200, Status message -> Success**

{

“student\_id”: 1234,

   "first\_name":"Bob",

   "last\_name":"Smith",

"age": 21

   "gender":”M”

}

Mapping



## Get student all students imformation : https://mps.co.za/mps-student-systems-api/v1/students

This is a GET based API for retrieving all students information respond back with 200 status code, Success status message and Json payload containing students information. On failure the API will respond back corresponding error code and message along with description.

1. The system API’s consumes relational database management system for students information.
   1. The mps-students-systems-api retrieve student information from the student RDBMS.
      1. The student RDBMS response with an appropriate response .

1.1.1.1 The mps-students-systems-api make student information available to any process api.

### Input and Output Definitions of API

**Output: Status code -> 200, Status message -> Success**

**{**

{

“student\_id”: 1234,

   "first\_name":"Bob",

   "last\_name":"Smith",

"age": 21

   "gender":”M”

},

{

“student\_id”: 1235,

   "first\_name":"Jessica",

   "last\_name":"Zulu",

"age": 16

   "gender":”F”

}

}

Mapping



## Create new student information : https://mps.co.za/mps-students-system-api/v1/student

This is a POST based API for creating new student information and respond back with 200 status code, Success status message and Json payload containing new created student information. On failure the API will respond back corresponding error code and message along with description.

1. The system API’s consumes relational database management system for students information.
   1. The mps-students-systems-api upload student information to the student RDBMS.
      1. The student RDBMS response with an appropriate response .

1.1.1.1 The mps-students-systems-api make student information available to any process api.

### Input and Output Definitions of API

**Output: Status code -> 201, Status message -> Created**

{

“student\_id”: 1234,

   "first\_name":"Bob",

   "last\_name":"Smith",

"age": 21

   "gender":”M”

}

Mapping



## Delete student by id : https://mps.co.za/mps-students-system-api/v1/student/{studentId}

This is a DELETE based API takes studentId as url parameter and respond back with 200 status code, Success status message and json payload containing deleted student information. On failure the API will respond back corresponding error code and message along with description.

1. The system API’s consumes relational database management system for student information.
   1. The mps-students-systems-api delete student information from the student RDBMS.
      1. The student RDBMS response with an appropriate response .

1.1.1.1 The mps-students-systems-api make student information available to any process api.

### Input and Output Definitions of API

**Output: Status code -> 200, Status message -> Success**

{

“student\_id”: 1234,

   "first\_name":"Bob",

   "last\_name":"Smith",

"age": 21

   "gender":”M”

}

Mapping



### Technical details

|  |  |
| --- | --- |
| API Name | mps-students-systems-api |
| Business Requirements | As mentioned in integration solution document |
| API Communication Type | REST |
| Interface Pattern | Request/Reply |
| Interface Definitions | RAML API Specification |
| Sample Request | JSON(Section 3.1.1) |
| Sample Response | JSON(Section 3.1.1) |

### Non Functional

|  |  |
| --- | --- |
| Post Execution Notification Details | Data-Related Failures Handling: (SALESFORCE as logging system)  Currently all the errors will be logged into logs with log4j.  Consumer API is responsible to log about error details.  Can be visible in runtime manager.  Technical Errors Handling: (SALESFORCE as logging system)  Currently all the errors will be logged into logs with log4j.  Consumer API is responsible to log about error details.  Can be visible in runtime manager. |
| Process Log Audit Details | Audit & Logging: (SALESFORCE as logging system)  Created Logs for the following   * Request message at entry point * Request message at system call entry point * Request message at system call entry point * Request completed |

### Error Handling

|  |  |  |
| --- | --- | --- |
| No. | Type of Exception | Exception Handling |
| 1 | Any technical issue | * API shall respond corresponding response message. * API will record the error in logs. * Runtime manager notifications will be configured - TBD * Application Administrative team will be notified using an email notification service- TBD |

### Integration test cases

| **Description** | **Result** |
| --- | --- |
| Successful case registration | 200 - Success |
| System connectivity failure (Database is down) | 503 - The (upstream) service is temporarily not available |
| Invalid JSON payload request | 400 – Bad request |
| Internal server error | 500 – Internal server error |
| API Kit Router errors | Global errors |

# API Policies, Access and Network Security

|  |
| --- |
| Define API policies to be configured / customized |

| **Requirement** | **Details** |
| --- | --- |
| Access to APIs - Authentication | client\_id enforcement |
| API Policy | CORS should be enabled |
| Communication over API’s | x-transaction-id and correlation-id |