Predefined Elements for Core Services

This list defines consistent elements for the Core Services category, ensuring modularity, reusability, and proper validation across core functionalities like email handling, workflows, task management, logging, and database operations.

---

1. Functions

Predefine function names and purposes for modularity and consistency:

Email Handling:

send\_email: Sends an email using SMTP.

parse\_email: Parses incoming email payloads to extract subject, sender, and body.

validate\_email: Validates the structure and required fields of an email payload.

Workflow Management:

execute\_workflow: Executes a workflow based on routing rules.

validate\_workflow: Validates that a workflow matches defined schemas.

Task Management:

create\_task: Creates a new task with required attributes.

update\_task\_status: Updates the status of a task (pending, in\_progress, completed).

escalate\_task: Escalates a task if overdue or unresolved.

Database Operations:

connect\_to\_database: Establishes a connection to PostgreSQL or SQLite.

fetch\_records: Fetches records from a database table.

insert\_record: Inserts a new record into a table.

Logging:

log\_event: Logs an event into the appropriate category (e.g., workflow, task, security).

rotate\_logs: Rotates logs based on retention policies.

---

2. Required Keys

Define mandatory keys for all core services:

Email Payload:

subject: Email subject.

sender: Email sender.

body: Email content.

Workflow Rules:

workflow.name: Unique name of the workflow.

workflow.rules: YAML file defining routing rules.

workflow.timeout: Time threshold for task escalation.

Task Attributes:

task\_id: Unique identifier for the task.

name: Name or description of the task.

status: Current status (pending, in\_progress, completed).

priority: Task priority level (e.g., low, medium, high).

assignee: User assigned to the task.

Database Configuration:

postgres.host: Hostname of the PostgreSQL server.

postgres.port: Port for PostgreSQL connections.

sqlite.file\_path: Path to the SQLite database file.

---

3. Standardized Outputs

Ensure all core service functions produce consistent outputs:

Email Parsing:

{

"subject": "Maintenance Request",

"sender": "user@example.com",

"body": "There is a leak in room 101."

}

Task Creation:

{

"task\_id": "123",

"name": "Fix plumbing issue",

"status": "pending",

"priority": "high",

"assignee": "jane.doe@example.com"

}

Database Query:

[

{"id": 1, "name": "Task 1", "status": "completed"},

{"id": 2, "name": "Task 2", "status": "pending"}

]

Workflow Execution:

{

"workflow\_name": "Maintenance Workflow",

"task\_id": "123",

"status": "completed",

"log": "Task completed successfully."

}

---

4. Validation Rules

Email Validation:

Required fields: subject, sender, body.

Maximum size: 10MB.

Allowed attachment types: .pdf, .png, .docx.

Workflow Validation:

Ensure workflow.rules includes:

condition: Criteria for routing (e.g., subject contains 'urgent').

action: Action to perform (e.g., route\_to, escalate\_after).

Task Validation:

Required fields: task\_id, name, status, assignee.

Database Validation:

Validate connection strings for PostgreSQL and SQLite.

Ensure queries are parameterized to prevent SQL injection.

---

5. Logging Standards

Ensure all logging adheres to the following fields:

timestamp: Date and time of the event.

log\_level: Severity level (INFO, ERROR, WARNING).

category: Log category (workflow, task, system, security).

message: Description of the event.

identifier: Unique ID (e.g., task\_id or workflow\_name).

Example Log Entry:

[2024-11-25 14:00:00] [INFO] [workflow] Task ID: 123 | Assigned to: jane.doe@example.com

---

6. Modular Design Patterns

Email Handling:

Separate email parsing, validation, and sending into different functions.

Workflow Management:

Modularize rule execution, task creation, and escalation handling.

Database Operations:

Centralize connection logic in a utility function (connect\_to\_database).

Logging:

Provide reusable log handlers for each category.

---

7. Predefined Task States

pending: Task has been created but not started.

in\_progress: Task is currently being executed.

completed: Task has been successfully completed.

failed: Task execution failed.

escalated: Task has been escalated due to unresolved issues.

---

8. Error Messages

Missing Fields:

Error: Missing required field '<FIELD\_NAME>'

Validation Failure:

Error: Invalid value for '<FIELD\_NAME>'

Unauthorized Access:

Error: Unauthorized access for user '<USERNAME>'

Task Failure:

Error: Task '<TASK\_NAME>' failed due to <REASON>

---

9. Configuration Placeholders

<SMTP\_SERVER>: Address of the SMTP server.

<IMAP\_SERVER>: Address of the IMAP server.

<DB\_HOST>: Database host.

<DB\_PORT>: Database port.

<LOG\_DIR>: Directory for storing logs.

---

Implementation Use

This list can be directly referenced when implementing email handling, workflows, task management, logging, or database operations for Core Services. Would you like me to apply this to a specific file or module?