1. ANALYTICS YAML FILE (SWAGGER EDITOR)

|  |
| --- |
| openapi: 3.0.1  info:  title: Analytics Service API  version: 1.0.0  paths:  /analytics/ingest:  post:  summary: Ingest analytics event from any service  requestBody:  required: true  content:  application/json:  schema:  $ref: '#/components/schemas/AnalyticsPayload'  responses:  '200':  description: Event ingested successfully  content:  application/json:  schema:  type: object  properties:  message:  type: string  example: "Analytics event recorded."  components:  schemas:  PageView:  type: object  required: [eventType, url]  properties:  eventType:  type: string  enum: [PageView]  url:  type: string  timestamp:  type: string  format: date-time  Click:  type: object  required: [eventType, elementId]  properties:  eventType:  type: string  enum: [Click]  elementId:  type: string  timestamp:  type: string  format: date-time  FormSubmission:  type: object  required: [eventType, formId]  properties:  eventType:  type: string  enum: [FormSubmission]  formId:  type: string  timestamp:  type: string  format: date-time  TimeOnPage:  type: object  required: [eventType, url, duration]  properties:  eventType:  type: string  enum: [TimeOnPage]  url:  type: string  duration:  type: number  timestamp:  type: string  format: date-time  ScrollDepth:  type: object  required: [eventType, url, depth]  properties:  eventType:  type: string  enum: [ScrollDepth]  url:  type: string  depth:  type: number  timestamp:  type: string  format: date-time  CustomEvent:  type: object  required: [eventType, name]  properties:  eventType:  type: string  enum: [CustomEvent]  name:  type: string  properties:  type: object  timestamp:  type: string  format: date-time  ErrorEvent:  type: object  required: [eventType, message]  properties:  eventType:  type: string  enum: [ErrorEvent]  message:  type: string  stack:  type: string  timestamp:  type: string  format: date-time  PerformanceMetric:  type: object  required: [eventType, name, value]  properties:  eventType:  type: string  enum: [PerformanceMetric]  name:  type: string  value:  type: number  timestamp:  type: string  format: date-time  UserInteraction:  type: object  required: [eventType, interactionType]  properties:  eventType:  type: string  enum: [UserInteraction]  interactionType:  type: string  metadata:  type: object  timestamp:  type: string  format: date-time  LoginSuccess:  type: object  required: [eventType, userId]  properties:  eventType:  type: string  enum: [LoginSuccess]  userId:  type: string  timestamp:  type: string  format: date-time  ProductViewed:  type: object  required: [eventType, productId]  properties:  eventType:  type: string  enum: [ProductViewed]  productId:  type: string  userId:  type: string  timestamp:  type: string  format: date-time  InventoryChecked:  type: object  required: [eventType, productId]  properties:  eventType:  type: string  enum: [InventoryChecked]  productId:  type: string  availableQuantity:  type: integer  timestamp:  type: string  format: date-time  PaymentSuccess:  type: object  required: [eventType, userId, amount]  properties:  eventType:  type: string  enum: [PaymentSuccess]  userId:  type: string  orderId:  type: string  amount:  type: number  paymentMethod:  type: string  timestamp:  type: string  format: date-time  OrderPlaced:  type: object  required: [eventType, orderId, userId]  properties:  eventType:  type: string  enum: [OrderPlaced]  orderId:  type: string  userId:  type: string  totalAmount:  type: number  timestamp:  type: string  format: date-time  AnalyticsPayload:  type: object  required: [event]  properties:  event:  oneOf:  - $ref: '#/components/schemas/PageView'  - $ref: '#/components/schemas/Click'  - $ref: '#/components/schemas/FormSubmission'  - $ref: '#/components/schemas/TimeOnPage'  - $ref: '#/components/schemas/ScrollDepth'  - $ref: '#/components/schemas/CustomEvent'  - $ref: '#/components/schemas/ErrorEvent'  - $ref: '#/components/schemas/PerformanceMetric'  - $ref: '#/components/schemas/UserInteraction'  - $ref: '#/components/schemas/LoginSuccess'  - $ref: '#/components/schemas/ProductViewed'  - $ref: '#/components/schemas/InventoryChecked'  - $ref: '#/components/schemas/PaymentSuccess'  - $ref: '#/components/schemas/OrderPlaced' |

1. TEMPLATE

|  |
| --- |
| openapi: 3.0.1  info:  title: Templates Service API  version: 1.0.0  paths:  /templates:  get:  summary: Get all templates  parameters:  - in: query  name: type  schema:  type: string  enum: [email, sms, push, ui]  description: Filter templates by type  - in: query  name: tag  schema:  type: string  description: Filter templates by tag  responses:  '200':  description: List of templates  content:  application/json:  schema:  type: array  items:  $ref: '#/components/schemas/Template'  post:  summary: Create a new template  requestBody:  required: true  content:  application/json:  schema:  $ref: '#/components/schemas/NewTemplate'  responses:  '201':  description: Template created  content:  application/json:  schema:  $ref: '#/components/schemas/Template'  /templates/{id}:  get:  summary: Get a template by ID  parameters:  - in: path  name: id  required: true  schema:  type: string  responses:  '200':  description: Template details  content:  application/json:  schema:  $ref: '#/components/schemas/Template'  '404':  description: Template not found  put:  summary: Update a template  parameters:  - in: path  name: id  required: true  schema:  type: string  requestBody:  required: true  content:  application/json:  schema:  $ref: '#/components/schemas/NewTemplate'  responses:  '200':  description: Template updated  content:  application/json:  schema:  $ref: '#/components/schemas/Template'  delete:  summary: Delete a template  parameters:  - in: path  name: id  required: true  schema:  type: string  responses:  '204':  description: Template deleted  /templates/{id}/render:  post:  summary: Render a template with variables  parameters:  - in: path  name: id  required: true  schema:  type: string  requestBody:  required: true  content:  application/json:  schema:  type: object  properties:  variables:  type: object  description: Key-value pairs for dynamic content  responses:  '200':  description: Rendered template  content:  application/json:  schema:  type: object  properties:  rendered:  type: string  example: "Hello John, your order #1234 has been shipped."  components:  schemas:  Template:  type: object  properties:  id:  type: string  name:  type: string  type:  type: string  enum: [email, sms, push, ui]  content:  type: string  createdAt:  type: string  format: date-time  updatedAt:  type: string  format: date-time  tags:  type: array  items:  type: string  NewTemplate:  type: object  required: [name, type, content]  properties:  name:  type: string  type:  type: string  enum: [email, sms, push, ui]  content:  type: string  tags:  type: array  items:  type: string |