

```

openapi: "3.0.1"
info:
  version: "0.1.0"
  title: "Digital Health Check - Lab Results API"
tags:
  - name: Lab Results Upload Service
    description: Service for uploading blood test
results for DHC's
servers:
  - url: "${server_url}"
    variables:
      basePath:
        default: "/"
paths:
  /results:
    post:
      tags:
        - Result
      summary: Upload test result
      parameters:
        - in: header
          name: x-api-key
          required: true
          description: Api key header
          schema:
            type: string
        - in: header
          name: X-Correlation-ID
          required: false
          description: used to track requests across
systems
          schema:
            type: string
            format: uuid
        - in: header
          name: X-User-ID
          required: false
          description: user ID of tester
          schema:
            type: string
      requestBody:
        required: true
        content:
          application/json:

```

```

        schema:
          $ref:
'#/components/schemas/SingleSampleResponse'
      responses:
        "201":
          description: "201 response - Test Result
upload successful"
          headers:
            Access-Control-Allow-Origin:
              schema:
                $ref:
'#/components/schemas/CreateResultResponse'
        400:
          description: Invalid input<hr><strong>Please
note that the response may have different formats as shown
in the examples. Content may vary depending on the data
provided.</strong>
          content:
            application/json:
              schema:
                $ref:
'#/components/schemas/ErrorResponse'
          examples:
            Additional properties:
              value:
                code: INVALID_INPUT
                message: "[object instance has
properties which are not allowed by the schema:
[\"extraField\"]]"
            Duplicated tests:
              value:
                code: INVALID_INPUT
                message: "[array must not contain
duplicate elements]"
            Invalid correlation ID:
              value:
                code: INVALID_INPUT
                message: "X-Correlation-Id is not
in the correct UUID format"
            Invalid email address:
              value:
                code: INVALID_INPUT
                message: "[string \"a@.com\" is not
a valid email address]"

```

```

Invalid JSON body:
  value:
    code: INVALID_INPUT
    message: "[Unknown error parsing
request body]"
Invalid type:
  value:
    code: INVALID_INPUT
    message: "[instance type (integer)
does not match any allowed primitive type (allowed:
[\"string\"])]"
Invalid subject ID:
  value:
    code: INVALID_INPUT
    message: "[input string \"a\" is not
a valid UUID]"
Missing required properties:
  value:
    code: INVALID_INPUT
    message: "[object has missing
required properties
([\"email\", \"firstName\", \"lastName\"])]"
Not allowed value:
  value:
    code: INVALID_INPUT
    message: "[instance value (\"abc\")
not found in enum (possible values:
[\"email\", \"mobile\", \"none\"])]"
Property does not match the pattern:
  value:
    code: INVALID_INPUT
    message: "[ECMA 262 regex
\"^\\+[\\d,\\ -()]{6,20}$|^0[\\ -]?7[\\d\\
()-]{9,12}$\" does not match input string \"044\"]"
Too long value:
  value:
    code: INVALID_INPUT
    message: "[string
\"aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaa\" is too long (length: 66, maximum
allowed: 64)]"
Too short value:
  value:
    code: INVALID_INPUT

```

```
        message: "[string \"\" is too short
(length: 0, required minimum: 1)]"
    headers:
        X-Correlation-Id:
            schema:
                type: string
                format: uuid
            description: The correlation ID of the
original request
    401:
        $ref: '#/components/responses/401'
    403:
        $ref: '#/components/responses/403'
    415:
        $ref: '#/components/responses/415'
    406:
        $ref: '#/components/responses/406'
    500:
        $ref: '#/components/responses/500'

components:
    parameters:
        Authorization:
            in: header
            name: Authorization
            required: true
            description: JWT token
            schema:
                $ref:
"#/components/securitySchemes/RequestAuthorizer"
        ApiKey:
            name: X-Api-Key
            in: header
            required: true
            schema:
                type: string
    CorrelationId:
        name: X-Correlation-Id
        in: header
        required: true
        schema:
            type: string
            format: uuid
    schemas:
```

```

TestId:
  type: string
  description: The Sample's Identifier
  pattern:
    "^((([a-zA-Z]{1,3})[0-9]{8})|(([a-zA-Z]{2})[0-9]{7})|
    ([a-zA-Z]{1,3})[0-9]{9}))$"
  example: "CHE87654321"
SingleSampleResponse:
  type: object
  $ref: "#/components/schemas/Result"
Result:
  type: object
  required:
    - testId
    - testType
    - patientId
    - resultData
    - abnormalResult
    - dateTime
    - notes
  properties:
    testId:
      $ref: '#/components/schemas/TestId'
    testType:
      type: string
      description: test type identifier
      example: "LFD"
    patientID:
      type: string
      description: patient identifier - NHS number
    resultData:
      type: array
      items:
        type: object
        properties:
          testType:
            type: string
            example: "Cholesterol"
          units:
            type: string
            example: "mmol/L"
          value:
            type: string
            example: "6.3"

```

```

        refRange:
            type: string
            example: "<4.5"
        abnormalResult:
            type: boolean
        description: The outcome of the Sample that was
tested
        dateTime:
            type: string
            format: date-time
            example: "2020-07-23T08:30:00.000Z"
        notes:
            type: string
            example: Some optional additional notes
provided by lab/patient
        labId:
            type: string
            example: the identifier of the lab allowing the
DHC to determine the source of the result
        CreateResultResponse:
            type: object
        properties:
            patientId:
                type: string
                format: uuid
                example:
"3f1af1db-ba2d-4524-b3fc-91d4dce78bb4"
                description: "Unique ID of the health check
patient"
            ErrorResponse:
                type: object
            properties:
                code:
                    type: string
                message:
                    type: string
        UndeliverableErrorResponse:
            type: object
        properties:
            code:
                type: string
            message:
                type: string
            undeliverableReason:

```

```

        type: string
responses:
  '401':
    description: Unauthorised user
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/ErrorResponse'
        example:
          code: UNAUTHORIZED
          message: Unauthorized
    headers:
      X-Correlation-Id:
        schema:
          type: string
          format: uuid
          description: The correlation ID of the
original request
  '403':
    description: Access denied
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/ErrorResponse'
        example:
          code: ACCESS_DENIED
          message: User is not authorized to access
this resource with an explicit deny
    headers:
      X-Correlation-Id:
        schema:
          type: string
          format: uuid
          description: The correlation ID of the
original request
  '406':
    description: Unable to deliver stock
    content:
      application/json:
        schema:
          oneOf:
            - $ref:
'#/components/schemas/ErrorResponse'
            - $ref:

```

```

'#/components/schemas/UndeliverableErrorResponse'
  examples:
    stock available:
      value:
        code: NO_STOCK_AVAILABLE
        message: There is no stock available for
the test type you have requested
    Undeliverable postcode:
      value:
        code: UNDELIVERABLE_POSTCODE
        message: The given postcode is not
deliverable
        undeliverableReason: scottish_island
  headers:
    X-Correlation-Id:
      schema:
        type: string
        format: uuid
      description: The correlation ID of the
original request
    '415':
      description: Unsupported Media Type
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/ErrorResponse'
          example:
            code: UNSUPPORTED_MEDIA_TYPE
            message: Unsupported Media Type
      headers:
        X-Correlation-Id:
          schema:
            type: string
            format: uuid
          description: The correlation ID of the
original request
    '422':
      description: Unprocessable entity
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/ErrorResponse'
          example:
            code: UNPROCESSABLE_ENTITY

```



```

        message: Failed to validate data
headers:
  X-Correlation-Id:
    schema:
      type: string
      format: uuid
    description: The correlation ID of the
original request
  '429':
    description: Too Many Requests
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/ErrorResponse'
        example:
          code: TOO_MANY_REQUESTS
          message: Too many requests
    headers:
      X-Correlation-Id:
        schema:
          type: string
          format: uuid
        description: The correlation ID of the
original request
  '500':
    description: 'Server error, something has gone
wrong'
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/ErrorResponse'
        example:
          code: SERVER_ERROR
          message: 'Oops, something went wrong on our
end'
securitySchemes:
  RequestAuthorizer:
    type: apiKey
    name: Authorization
    in: header
    x-amazon-apigateway-authtype: custom
    x-amazon-apigateway-authorizer:
      type: request
      identitySource:

```

```

method.request.header.Authorization
    authorizerResultTtlInSeconds: 0
x-amazon-apigateway-gateway-responses:
    DEFAULT_4XX:
        responseParameters:
            gatewayresponse.header.Access-Control-Allow-Origin: "'*'"
            gatewayresponse.header.X-Correlation-Id:
method.request.header.X-Correlation-Id
            gatewayresponse.header.Strict-Transport-Security: "'max-age=31536000; includeSubdomains; preload'"
        responseTemplates:
            application/json: '{"code": "INVALID_REQUEST",
"message": "$context.error.message" }'
    BAD_REQUEST_PARAMETERS:
        statusCode: 400
        responseParameters:
            gatewayresponse.header.Access-Control-Allow-Origin: "'*'"
            gatewayresponse.header.X-Correlation-Id:
method.request.header.X-Correlation-Id
            gatewayresponse.header.Strict-Transport-Security: "'max-age=31536000; includeSubdomains; preload'"
        responseTemplates:
            application/json: '{"code": "INVALID_INPUT",
"message": "$context.error.validationErrorString"}'
    BAD_REQUEST_BODY:
        statusCode: 400
        responseParameters:
            gatewayresponse.header.Access-Control-Allow-Origin: "'*'"
            gatewayresponse.header.X-Correlation-Id:
method.request.header.X-Correlation-Id
            gatewayresponse.header.Strict-Transport-Security: "'max-age=31536000; includeSubdomains; preload'"
        responseTemplates:
            application/json: '{"code": "INVALID_INPUT",
"message": "$context.error.validationErrorString"}'
    UNAUTHORIZED:
        statusCode: 401
        responseParameters:
            gatewayresponse.header.Access-Control-Allow-Origin: "'*'"
            gatewayresponse.header.X-Correlation-Id:

```

```

method.request.header.X-Correlation-Id
    gatewayresponse.header.Strict-Transport-Security: "'max-age=31536000; includeSubdomains; preload'"
    responseTemplates:
        application/json: '{"code": "UNAUTHORIZED",
"message": "$context.error.message"}'
    ACCESS_DENIED:
        statusCode: 403
        responseParameters:
            gatewayresponse.header.Access-Control-Allow-Origin: "'*'"
            gatewayresponse.header.X-Correlation-Id:
method.request.header.X-Correlation-Id
    gatewayresponse.header.Strict-Transport-Security: "'max-age=31536000; includeSubdomains; preload'"
    responseTemplates:
        application/json: '{"code": "ACCESS_DENIED",
"message": "$context.error.message"}'
    EXPIRED_TOKEN:
        statusCode: 403
        responseParameters:
            gatewayresponse.header.Access-Control-Allow-Origin: "'*'"
            gatewayresponse.header.X-Correlation-Id:
method.request.header.X-Correlation-Id
    gatewayresponse.header.Strict-Transport-Security: "'max-age=31536000; includeSubdomains; preload'"
    responseTemplates:
        application/json: '{"code": "EXPIRED_TOKEN",
"message": "$context.error.message"}'
    MISSING_AUTHENTICATION_TOKEN:
        statusCode: 403
        responseParameters:
            gatewayresponse.header.Access-Control-Allow-Origin: "'*'"
            gatewayresponse.header.X-Correlation-Id:
method.request.header.X-Correlation-Id
    gatewayresponse.header.Strict-Transport-Security: "'max-age=31536000; includeSubdomains; preload'"
    responseTemplates:
        application/json: '{"code":
"MISSING_AUTHENTICATION_TOKEN", "message":
"$context.error.message"}'
    REQUEST_TOO_LARGE:

```

```

        statusCode: 413
        responseParameters:
            gatewayresponse.header.Access-Control-Allow-Origin: '*'
            gatewayresponse.header.X-Correlation-Id:
method.request.header.X-Correlation-Id
            gatewayresponse.header.Strict-Transport-Security: "max-age=31536000; includeSubdomains; preload"
        responseTemplates:
            application/json: '{"code": "REQUEST_TOO_LARGE",
"message": "Request payload exceeds max size for APIGW"}'
        UNSUPPORTED_MEDIA_TYPE:
            statusCode: 415
            responseParameters:
                gatewayresponse.header.Access-Control-Allow-Origin: '*'
                gatewayresponse.header.X-Correlation-Id:
method.request.header.X-Correlation-Id
                gatewayresponse.header.Strict-Transport-Security: "max-age=31536000; includeSubdomains; preload"
            responseTemplates:
                application/json: '{"code":
"UNSUPPORTED_MEDIA_TYPE", "message":
"$context.error.message"}'
        THROTTLED:
            statusCode: 429
            responseParameters:
                gatewayresponse.header.Access-Control-Allow-Origin: '*'
                gatewayresponse.header.X-Correlation-Id:
method.request.header.X-Correlation-Id
                gatewayresponse.header.Strict-Transport-Security: "max-age=31536000; includeSubdomains; preload"
            responseTemplates:
                application/json: '{"code": "TOO_MANY_REQUESTS",
"message": "$context.error.message"}'
        DEFAULT_5XX:
            responseParameters:
                gatewayresponse.header.Access-Control-Allow-Origin: '*'
                gatewayresponse.header.X-Correlation-Id:
method.request.header.X-Correlation-Id
                gatewayresponse.header.Strict-Transport-Security: "max-age=31536000; includeSubdomains; preload"

```

```
responseTemplates:  
  application/json: '{"code":  
"INTERNAL_SERVER_ERROR", "message":  
"$context.error.message"}'
```