The Ultimate Toy Monster Truck API by Spin Master (created by Fiona Ho)

Objective: Design an API that allows users to obtain/retrieve information about all discontinued, existing, physical, and virtual toy monster trucks from Spin Master in company DB. The virtual properties are especially difficult to track, so a well-structured database along with API functions should be implemented.

Tool:
SwaggerHub is the API design and documentation platform built for teams to drive consistency and discipline across their API development workflow.

Estimated Team:
API Designer
API Documentation
API Development

Planning Resource: Monster Truck

GET - gives info for an array of trucks
Query Parameters - body
Responses - 200 payload: ID, Name, Type, Year, Status, Image File Name
404 not found

GET - gives info for a single truck
Query Parameters - ID
Responses - 200 payload: ID, Name, Type, Year, Status, Image File Name
404 not found

POST - insert info for a single truck
Query Parameters - ID, Name, Type, Year, Status, Image File Name
Responses - 200
400 invalid input
404 not found

PUT - Update info for a single truck
Query Parameters - ID, Name, Type, Year, Status, Image File Name
Responses - 200
404 not found

DELETE - remove info for a single truck
Query Parameters - ID
Responses - 200
404 not found

Implementation: Spin Master Monster Jam Encyclopedia App (Internal consumption)
Spin Master can create this application to help employees familiarize themselves with all the Monster Jam Trucks that are created instead of flipping through documentations after documentations. In the design department, errors in truck name referencing and or duplication can be avoided.

Limitation:
Technical - This Monster Jam API starts with a basic structure and will expand in the future as an open-source API when a stable version has been released.
Legal - At this point, the Monster Jam API is only available for internal use only. All third-party usage is prohibited.
Pricing - Free for all Spin Master employees to use.

API design only practice, no actual API was created.

```
openapi: 3.0.0
info:
  version: 2.0.0
  title: Monster Jam API
  description: 'An API that allows users to obtain/retrieve information about all discontinued,
existing, physical, and virtual toy monster trucks from Spin Master in company DB.'
  termsOfService: https://spinmaster.com/term-of-use
  contact:
    name: Fiona
    url: spinmaster.com
```

```
    email: fiona@spinmaster.com
  license:
    name: Spin Master License
    url: http://license.foo.com

servers:
  - url: https://dev.spinmaster.com/v1
    description: Dev server
  - url: https://dev.foo.com/v1
    description: Prod Server

paths:
  /trucks:
    get:
      description: Obtain information about all discontinued, existing, physical, and virtual toy
monster trucks from Spin Master in company DB
      parameters:
        - name: bodyLimit
          in: query
          description: The amount of trucks returned per query
          schema:
            type: integer
            minimum: 10
            maximum: 20
            example: 15

        - name: pageLimit
          in: query
          description: The pages to return truck info
          schema:
            type: integer
            minimum: 1
            maximum: 5
            example: 2

      responses:
        200:
          description: Successful pull of truck info
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/Trucks'
            application/xml:
              schema:
                $ref: '#/components/schemas/Trucks'
        404:
          description: Trucks not found


    post:
```

```
    description: Insert information for a single toy monster trucks from Spin Master in company
DB
    requestBody:
      required: true
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/Truck'
        application/xml:
            schema:
              $ref: '#/components/schemas/Truck'

    responses:
      200:
        description: Successfully inserted a new truck
      400:
        description: Invalid inputs

  /trucks/{id}:
    get:
      description: Obtain information of a single truck from Spin Master in company DB
      parameters:
        - in: path
          name: id
          required: true
          description: The ID of the toy truck
          schema:
            type: integer
            example: 4

      responses:
        200:
          description: Successfully loaded truck info based on ID
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/Truck'
            application/xml:
              schema:
                $ref: '#/components/schemas/Truck'
        404:
          description: Truck not found

    put:
      description: Update a single truck info with id
      parameters:
        - in: path
          name: id
          description: Truck id to update
          required: true
```

```
        schema:
          type: integer
          example: 4

    responses:
      200:
          description: Successfully update truck and return update truck info
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/Truck'
            application/xml:
              schema:
                $ref: '#/components/schemas/Truck'
      400:
        description: Invalid ID supplied
      404:
        description: Truck not found

  delete:
    description: Delete a single truck by id
    parameters:
      - in: path
        name: id
        description: Truck id to delete
        required: true
        schema:
          type: integer
          example: 4
    responses:
      200:
          description: Successfully delete truck and return delete truck info
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/Truck'
            application/xml:
              schema:
                $ref: '#/components/schemas/Truck'
      400:
        description: Invalid ID supplied
      404:
        description: Truck not found

components:
  schemas:

    Trucks:
      description: Array of truck info
      type: array
```

```
  items:
    $ref: '#/components/schemas/Truck'


Truck:
  description: Model containing truck info
  properties:
    id:
      type: integer
      example: 4

    truck name:
      type: string
      example: Grave Digger

    truck type:
      type: string
      example: virtual

    year made:
      type: number
      example: 1998

    discontinued:
      type: boolean
      default: false

    image file name:
      type: string
      example: gd_1998.png
```