# ADVANCED CONCEPTS

| String length and pattern | | |
|---|---|---|
| **Property** | **Type** | **Description** |
| minLength | number | minimum length of string |
| maxLength | number | the maximum length of string |
| pattern | string | using regular expressions |

```
movieId:
  name: movieId
  in: path
  required: true
  description: movieId of the movie
  type: string
  pattern: "[A-Z][0-9]{1,3}"
  minLength: 2
  maxLength: 4
```

```yaml
Movie:
  required:
    - movieId
  properties:
    movieTitle:
      type: string
    leadActor:
      type: string
    movieId:
      type: string
    releaseDate:
      type: string
      format: date
```

| Value | Type | Format |
|-------|------|--------|
| integer | integer | int32 |
| long | integer | int64 |
| float | number | float |
| double | number | |

| Property | Type | Description |
|----------|------|-------------|
| minimum | number | Minimum value |
| maximum | number | Maximum value |
| exclusiveMinimum | boolean | value > minimum |
| exclusiveMaximum | boolean | value < maximum |
| multipleOf | number | value is the multipleOf |

```yaml
collection:
  type: number
  format: double
  minimum: 0
  readOnly: true




segmentSize:
  name: segmentSize
  in: query
  description: number of movies returned
  type: integer
  format: int32
  minimum: 0
  exclusiveMinimum: true
  maximum: 100
  exclusiveMaximum: false
  multipleOf: 10




language:
  type: string
  enum:
    - Hindi
    - English
    - French
```

```yaml
Movies:
  properties:
    items:
      type: array
      minItems: 10
      maxItems: 100
      uniqueItems: true
      items:
        $ref: "#/definitions/Movie"
```

```yaml
collection:
  type: number
  format: double
  minimum: 0
  readOnly: true
```

```yaml
swagger: '2.0'

info:

  version: '1.0'

  title: 'Movie Store API'

  description: 'This API will provide information regarding
the titles catalog in the system '

# Added by API Auto Mocking Plugin

host: virtserver.swaggerhub.com

basePath: /arjunsingh_qwerty/MovieStore/1.0

schemes:

 - https

paths:

  /movies:

    get:

      summary: Get list of movies

      description: Returns a list containing all the movies

      parameters:

        - $ref: "#/parameters/segmentSize"
```

```yaml
      - $ref: "#/parameters/segmentNumber"
    responses:
      200:
        description: A list of movies
        schema:
          type: array
          items:
            $ref: "#/definitions/Movies"
      500:
        $ref: "#/responses/Standard500ErrorResponse"
  post:
    summary: Creates a movie record
    description: adds a new movie to the list
    parameters:
      - name: movie
        in: body
        description: the movie to be created
        schema:
```

```yaml
          $ref: "#/definitions/Movie"
      responses:
        200:
          description: Movies successfully added
        400:
          description: Movies couldn't be added
        500:
          $ref: "#/responses/Standard500ErrorResponse"
  /movies/{movieId}:
    parameters:
      - $ref: "#/parameters/movieId"
    get:
      summary: Gets a movie
      description: Returns a single movie for its movieId
      responses:
        200:
          description: A movie
          schema:
```

```yaml
        $ref: "#/definitions/Movie"
    404:
      description: The movie does not exist in the database
    500:
      $ref: "#/responses/Standard500ErrorResponse"
  delete:
    summary: Deletes a movie
    description: Deletes a single movie for its movieId
    responses:
      204:
        description: Movie successfully deleted
      404:
        $ref: "#/responses/Standard404ErrorResponse"
      500:
        $ref: "#/responses/Standard500ErrorResponse"
```

```yaml
definitions:
  Movie:
    required:
      - movieId
    properties:
      movieTitle:
        type: string
      leadActor:
        type: string
      movieId:
        type: string
      releaseDate:
        type: string
        format: date
      language:
        type: string
        enum:
          - Hindi
```

```yaml
        - English

        - French

    collection:

      type: number

      format: double

      minimum: 0

      readOnly: true

Movies:

  properties:

    items:

      type: array

      minItems: 10

      maxItems: 100

      uniqueItems: true

      items:

        $ref: "#/definitions/Movie"

Error:

  properties:
```

```yaml
      code:
        type: string
      message:
        type: string


responses:
  Standard500ErrorResponse:
    description: An unexpected error has occured
    schema:
      $ref: "#/definitions/Error"
  Standard404ErrorResponse:
    description: An unexpected error has occured
    schema:
      $ref: "#/definitions/Error"


parameters:
  movieId:
    name: movieId
```

in: path

required: true

description: movieId of the movie

type: string

pattern: "[A-Z][0-9]{1,3}"

minLength: 2

maxLength: 4

segmentSize:

  name: segmentSize

  in: query

  description: number of movies returned

  type: integer

  format: int32

  minimum: 0

  exclusiveMinimum: true

  maximum: 100

  exclusiveMaximum: false

  multipleOf: 10

```yaml
segmentNumber:
  name: segmentNumber
  in: query
  description: segment number
  type: integer
```