OGC Symbology

Winter School 2022

Ms. Prajwalita J. Chavan
IIT, Bombay

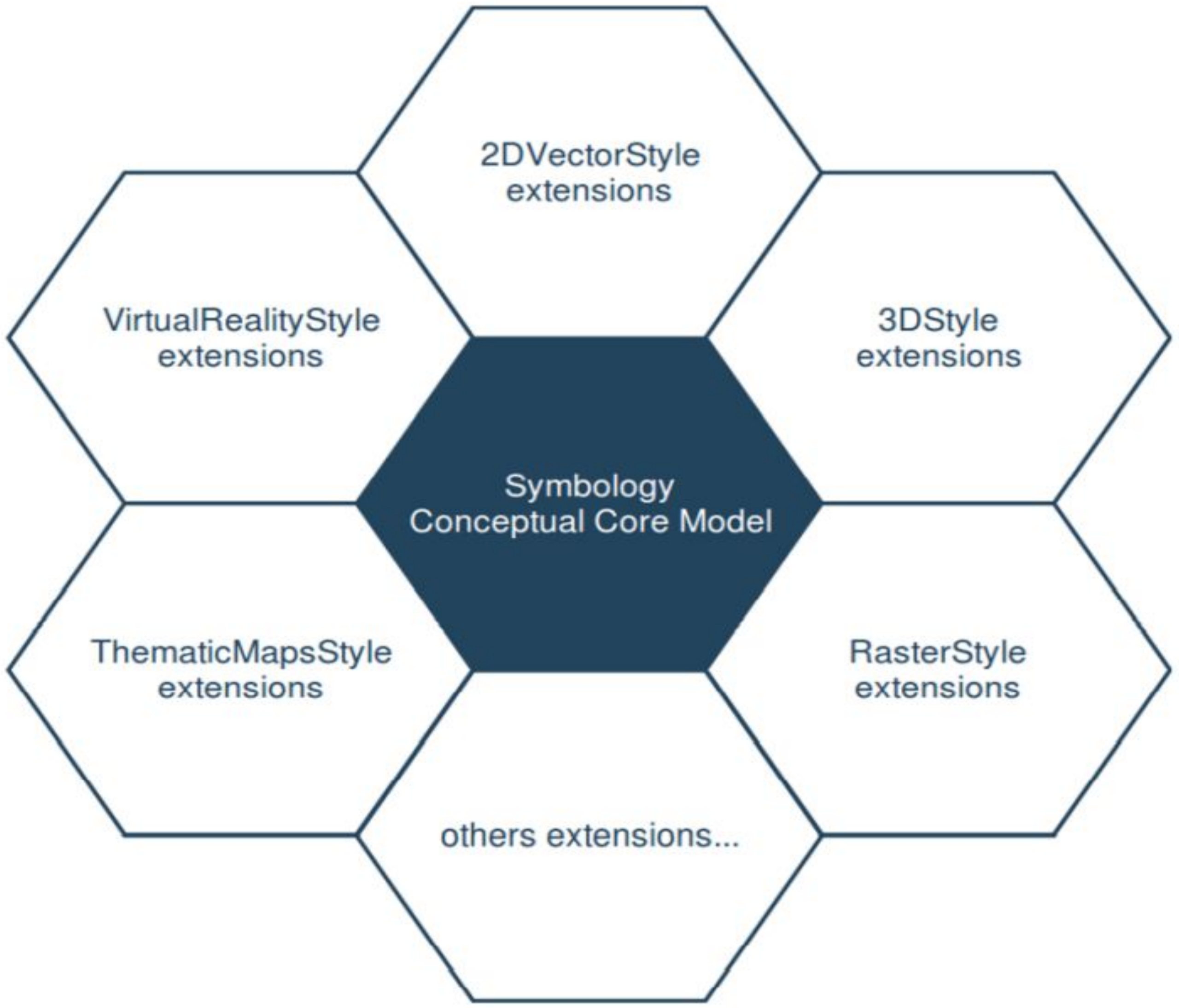
Overview

- About OGC API – Symbology
- SymCore
- Principles of Implementation
- Conceptual Model Core
- Class Style
- Class Rule
- Class Symbolizer
- Class Parameter Value
- Class Literal and color
- Class Fill
- Class Stroke
- Class Graphic and Graphic Size
- Class Label
- Class Font

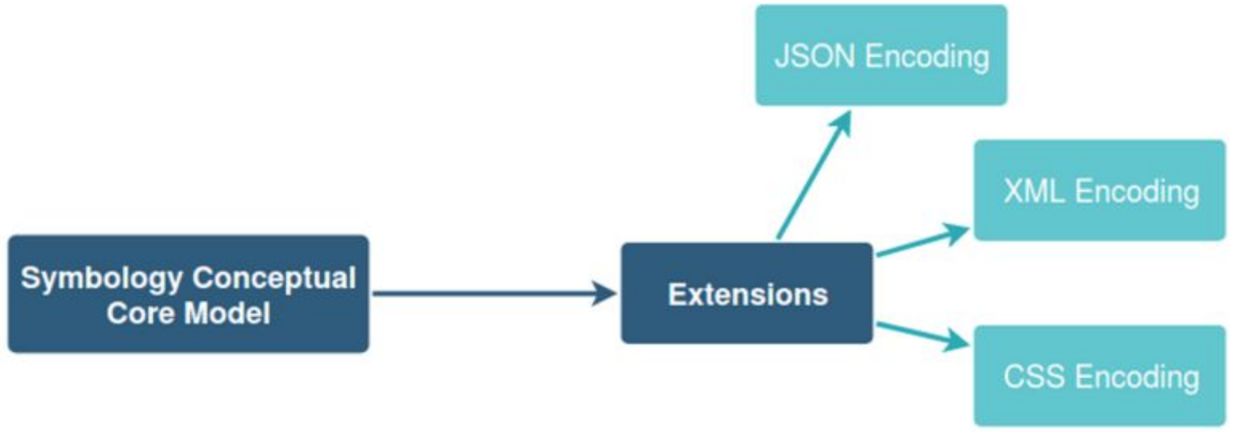
OGC API - Symbology

- **Publication Date:** 2020-10-15
- **Submitter**: : Erwan Bocher, Olivier Ertz: Switzerland
- The **SymCore** is a conceptual, modular, neutral model for the **portrayal of geographical data**
- The SymCore is a new approach to provide the **flexibility** required to achieve adequate **cartographic styling** and fill the needs of a variety of information communities; e.g., aviation symbols, weather symbols, thematic maps, etc
- It achieves high level **styling interoperability** without encoding dependencies

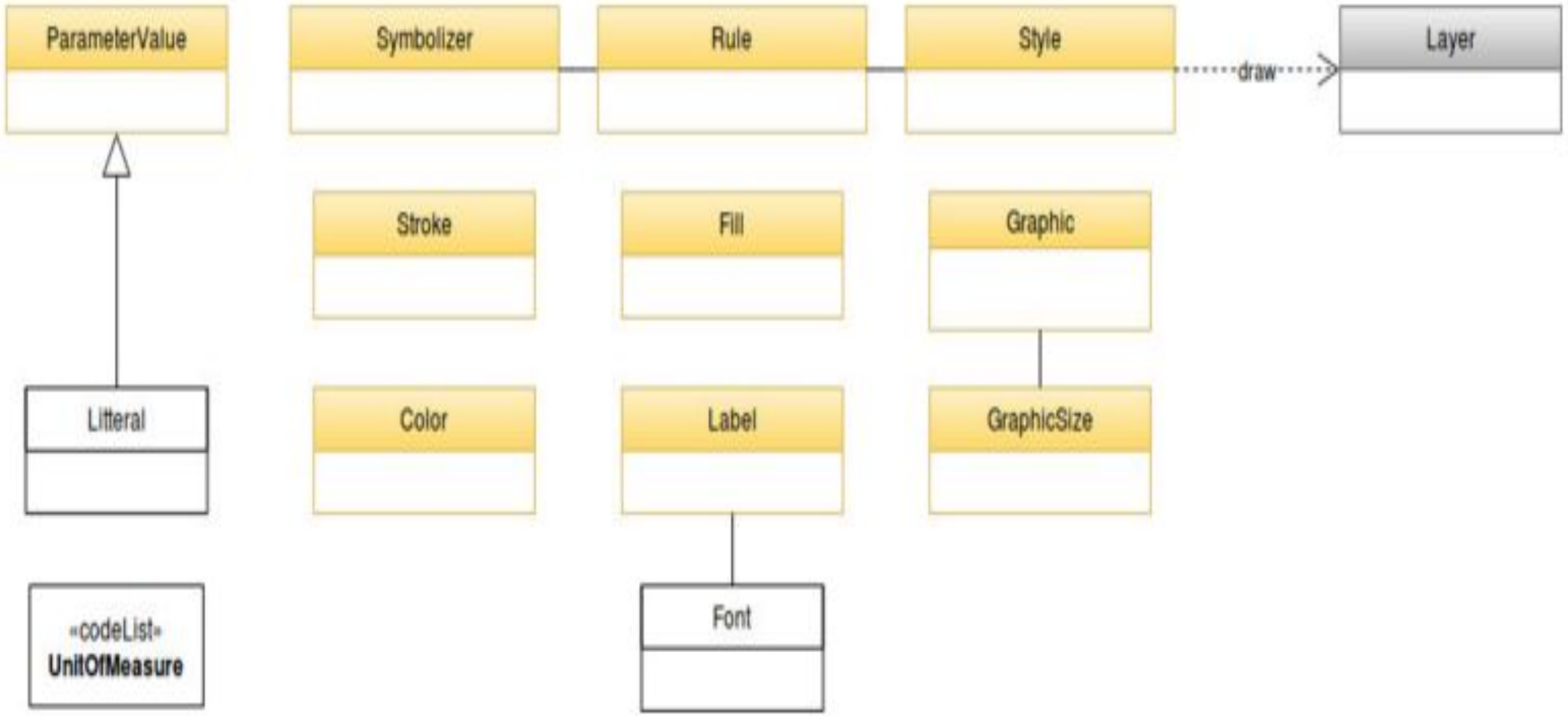
OGC Symbology: Conceptual Model Core (SymCore)


2DVectorStyle extensions
VirtualRealityStyle extensions
3DStyle extensions
Symbology Conceptual Core Model
ThematicMapsStyle extensions
RasterStyle extensions
others extensions...

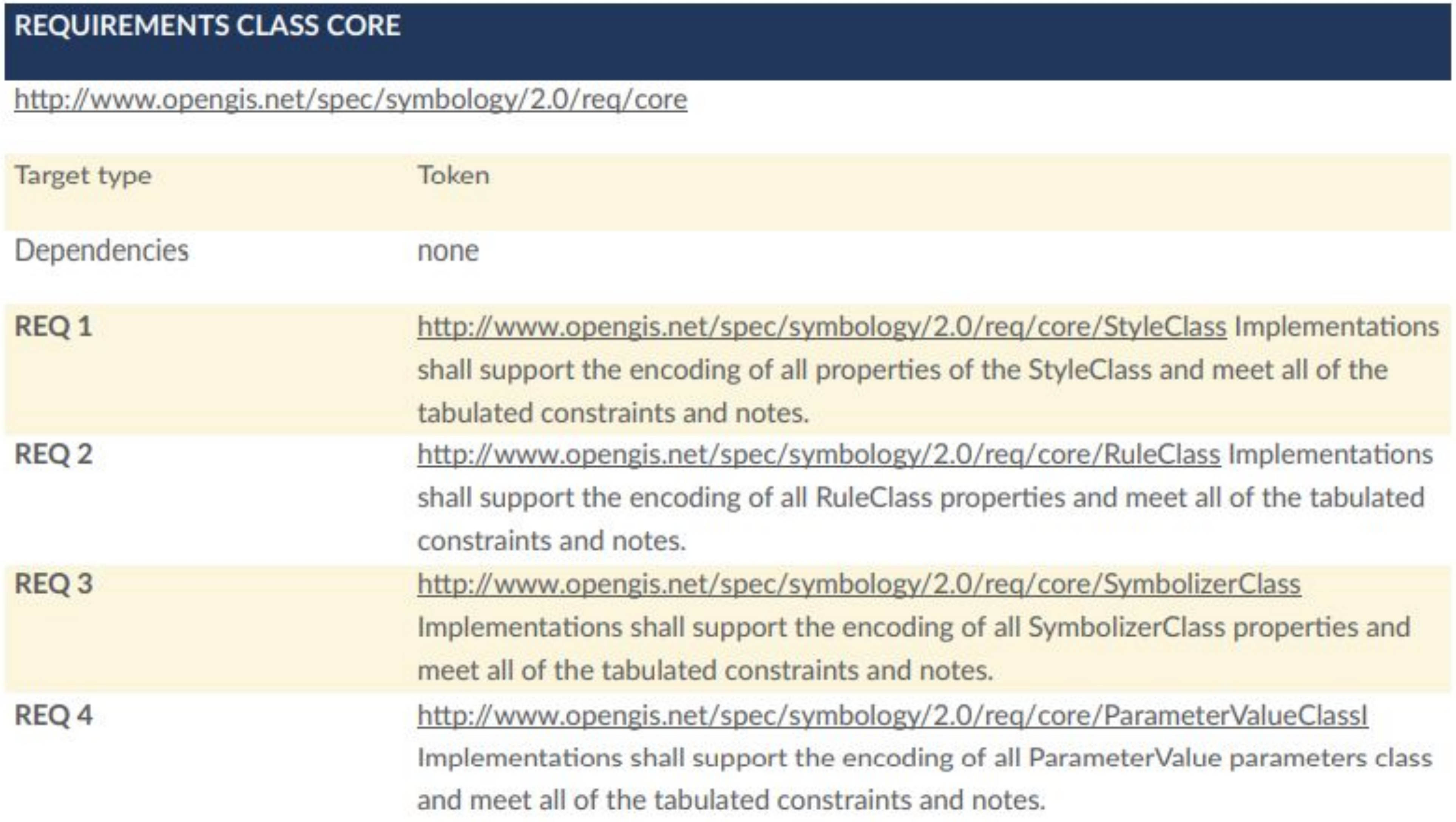
Principles of implementation



Core Conceptual Model

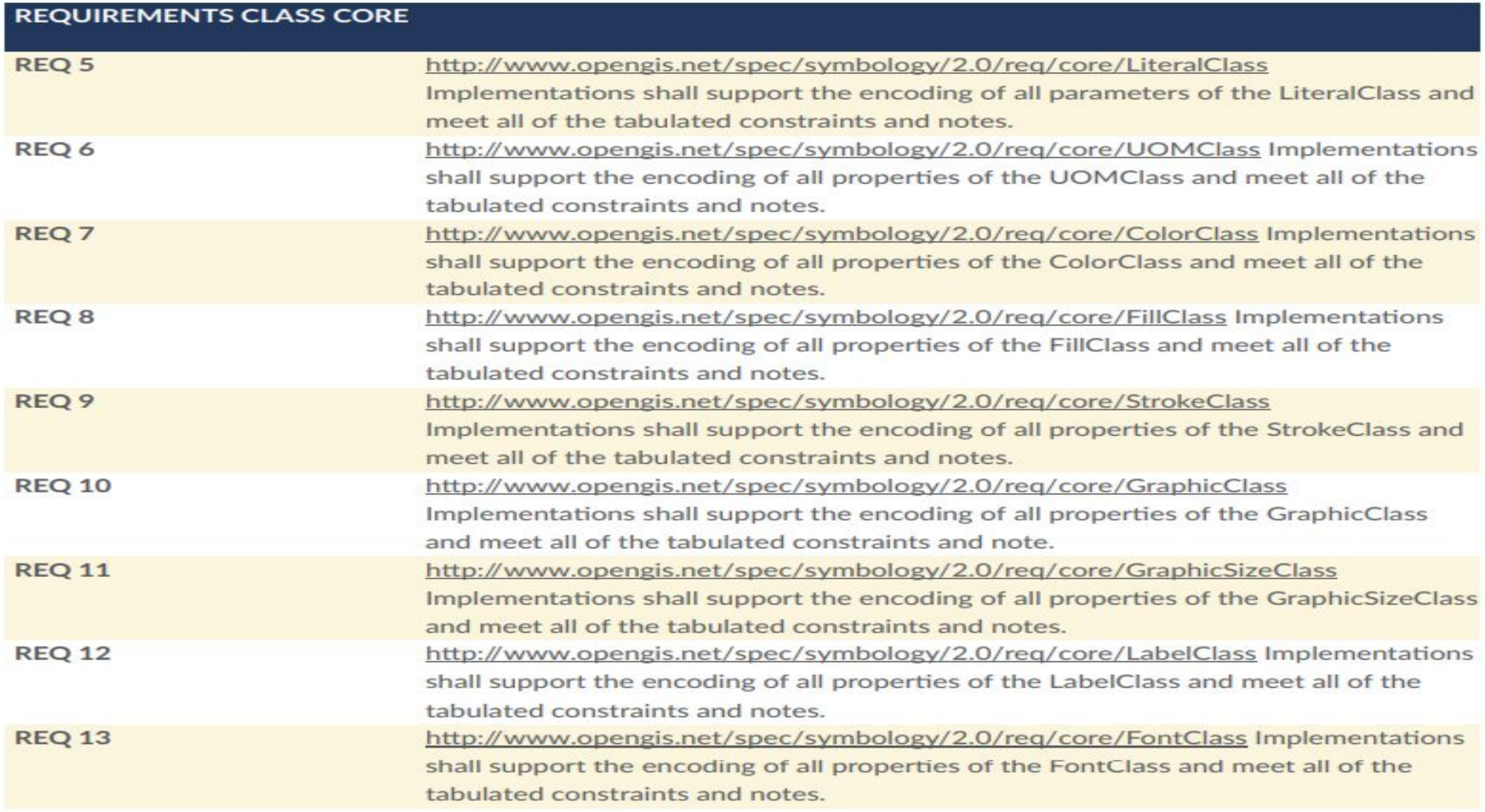


Core Conceptual Model



REQUIREMENTS CLASS CORE	
http://www.opengis.net/spec/symbology/2.0/req/core	
Target type	Token
Dependencies	none
REQ 1	http://www.opengis.net/spec/symbology/2.0/req/core/StyleClass Implementations shall support the encoding of all properties of the StyleClass and meet all of the tabulated constraints and notes.
REQ 2	http://www.opengis.net/spec/symbology/2.0/req/core/RuleClass Implementations shall support the encoding of all RuleClass properties and meet all of the tabulated constraints and notes.
REQ 3	http://www.opengis.net/spec/symbology/2.0/req/core/SymbolizerClass Implementations shall support the encoding of all SymbolizerClass properties and meet all of the tabulated constraints and notes.
REQ 4	http://www.opengis.net/spec/symbology/2.0/req/core/ParameterValueClassl Implementations shall support the encoding of all ParameterValue parameters class and meet all of the tabulated constraints and notes.

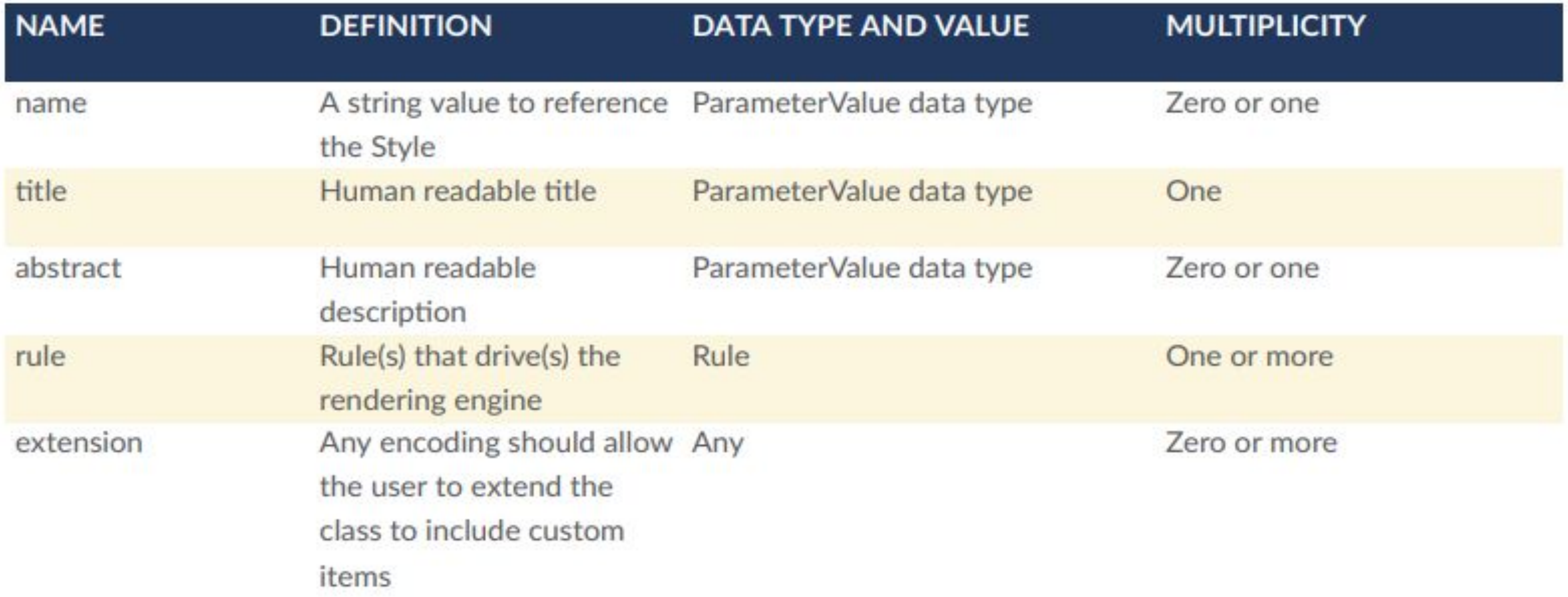
Core Conceptual Model



REQUIREMENTS CLASS CORE	
REQ 5	http://www.opengis.net/spec/symbology/2.0/req/core/LiteralClass Implementations shall support the encoding of all parameters of the LiteralClass and meet all of the tabulated constraints and notes.
REQ 6	http://www.opengis.net/spec/symbology/2.0/req/core/UOMClass Implementations shall support the encoding of all properties of the UOMClass and meet all of the tabulated constraints and notes.
REQ 7	http://www.opengis.net/spec/symbology/2.0/req/core/ColorClass Implementations shall support the encoding of all properties of the ColorClass and meet all of the tabulated constraints and notes.
REQ 8	http://www.opengis.net/spec/symbology/2.0/req/core/FillClass Implementations shall support the encoding of all properties of the FillClass and meet all of the tabulated constraints and notes.
REQ 9	http://www.opengis.net/spec/symbology/2.0/req/core/StrokeClass Implementations shall support the encoding of all properties of the StrokeClass and meet all of the tabulated constraints and notes.
REQ 10	http://www.opengis.net/spec/symbology/2.0/req/core/GraphicClass Implementations shall support the encoding of all properties of the GraphicClass and meet all of the tabulated constraints and note.
REQ 11	http://www.opengis.net/spec/symbology/2.0/req/core/GraphicSizeClass Implementations shall support the encoding of all properties of the GraphicSizeClass and meet all of the tabulated constraints and notes.
REQ 12	http://www.opengis.net/spec/symbology/2.0/req/core/LabelClass Implementations shall support the encoding of all properties of the LabelClass and meet all of the tabulated constraints and notes.
REQ 13	http://www.opengis.net/spec/symbology/2.0/req/core/FontClass Implementations shall support the encoding of all properties of the FontClass and meet all of the tabulated constraints and notes.

Class Style

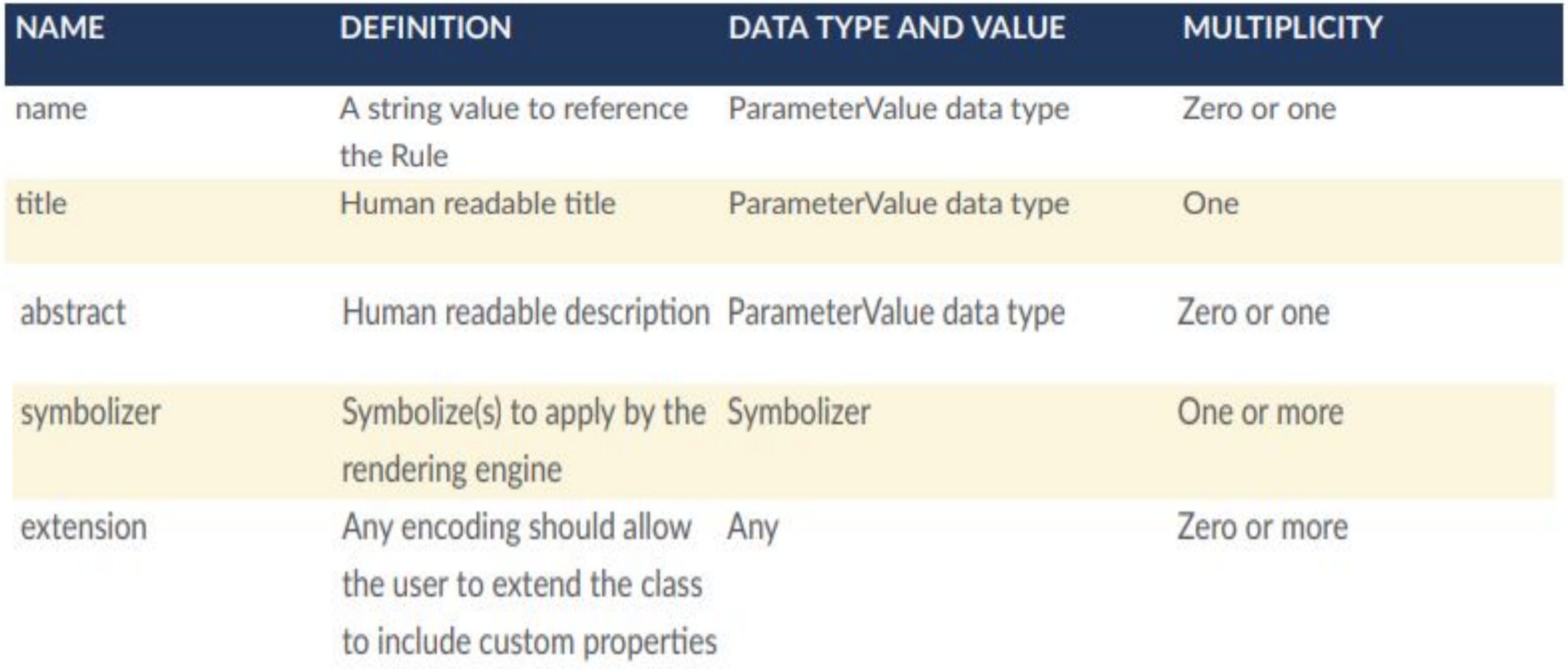
- This class is the **root concept of the Symbology** Conceptual Core Model. This class organizes the **rules of symbolizing instructions** to be applied by a rendering engine on a layer of geographic features



NAME	DEFINITION	DATA TYPE AND VALUE	MULTIPLICITY
name	A string value to reference the Style	ParameterValue data type	Zero or one
title	Human readable title	ParameterValue data type	One
abstract	Human readable description	ParameterValue data type	Zero or one
rule	Rule(s) that drive(s) the rendering engine	Rule	One or more
extension	Any encoding should allow the user to extend the class to include custom items	Any	Zero or more

Class Rule

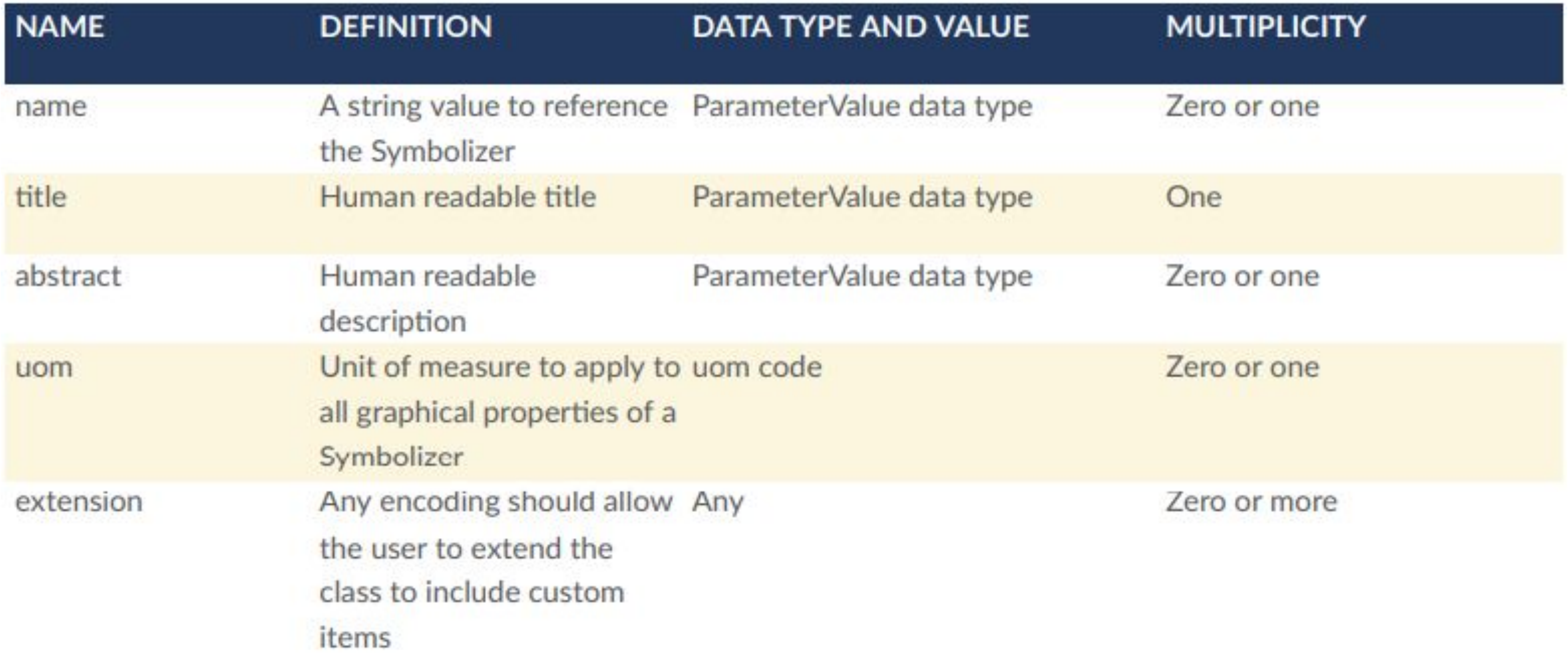
- This core class describes the **concept of a rule** in the Symbology model. Rules are used to organize symbolizing instructions and potentially to **define conditions of application** of these associated symbolizers



NAME	DEFINITION	DATA TYPE AND VALUE	MULTIPLICITY
name	A string value to reference the Rule	ParameterValue data type	Zero or one
title	Human readable title	ParameterValue data type	One
abstract	Human readable description	ParameterValue data type	Zero or one
symbolizer	Symbolize(s) to apply by the rendering engine	Symbolizer	One or more
extension	Any encoding should allow the user to extend the class to include custom properties	Any	Zero or more

Class Symbolizer

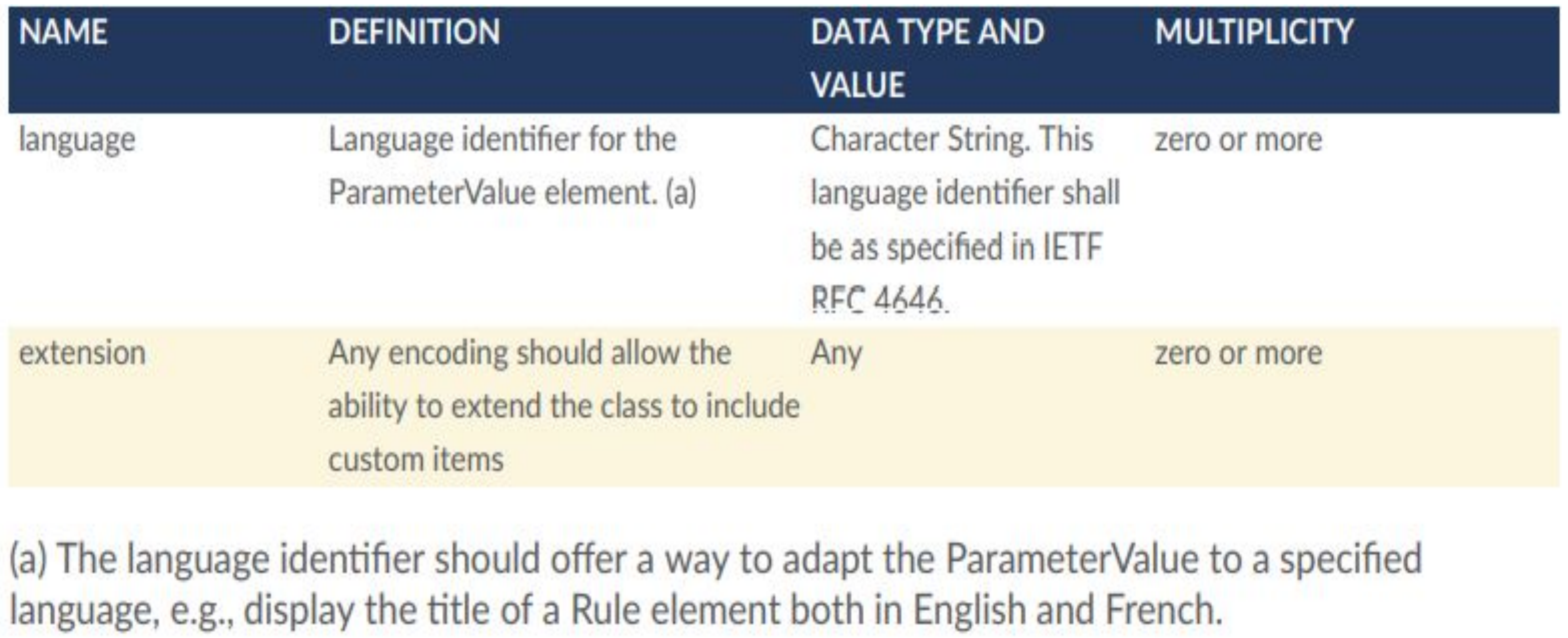
- This class describes how to **portray geographic data given a shape** (e.g., area fill, line stroke, point marker, etc.) **and graphical properties** (e.g., color, opacity, font-family, etc.). As an abstract class, it is designed to be extended



NAME	DEFINITION	DATA TYPE AND VALUE	MULTIPLICITY
name	A string value to reference the Symbolizer	ParameterValue data type	Zero or one
title	Human readable title	ParameterValue data type	One
abstract	Human readable description	ParameterValue data type	Zero or one
uom	Unit of measure to apply to all graphical properties of a Symbolizer	uom code	Zero or one
extension	Any encoding should allow the user to extend the class to include custom items	Any	Zero or more

Class Parameter Value

- The ParameterValue class represents a gateway that **provides the value to be used by a parameter in a styling context** of use (almost all styling parameters such as width, opacity, displacement, etc. are “parameter-values”)

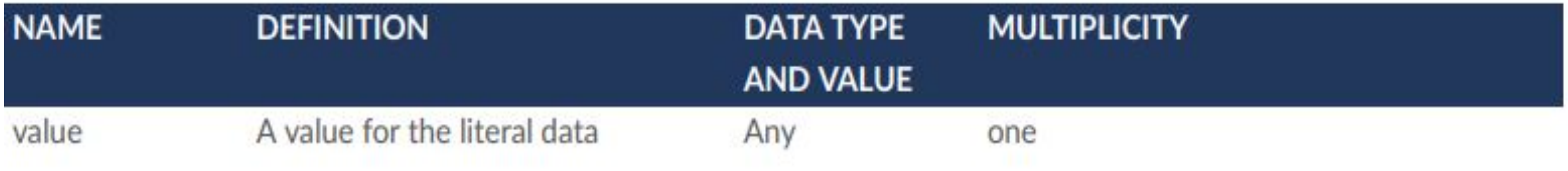


NAME	DEFINITION	DATA TYPE AND VALUE	MULTIPLICITY
language	Language identifier for the ParameterValue element. (a)	Character String. This language identifier shall be as specified in IETF RFC 4646.	zero or more
extension	Any encoding should allow the ability to extend the class to include custom items	Any	zero or more

(a) The language identifier should offer a way to adapt the ParameterValue to a specified language, e.g., display the title of a Rule element both in English and French.

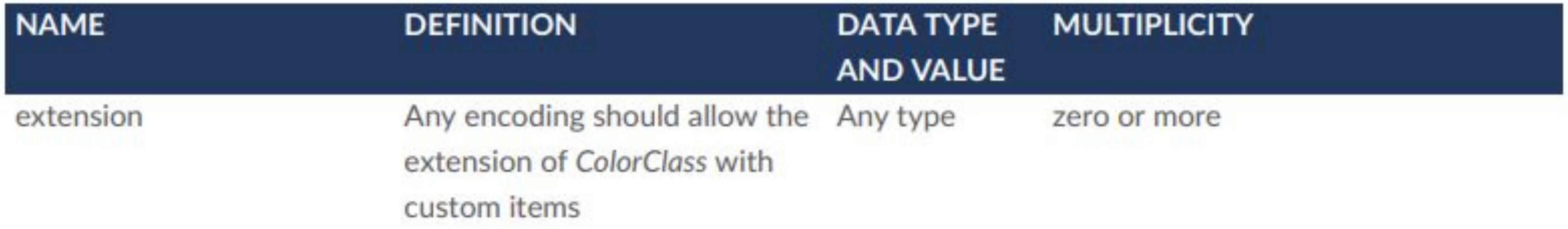
Class Literal and Color

- The Literal class is a concrete implementation of the ParameterValue class. LiteralClass represents a **typed atomic literal value** as a constant explicitly specified



NAME	DEFINITION	DATA TYPE AND VALUE	MULTIPLICITY
value	A value for the literal data	Any	one

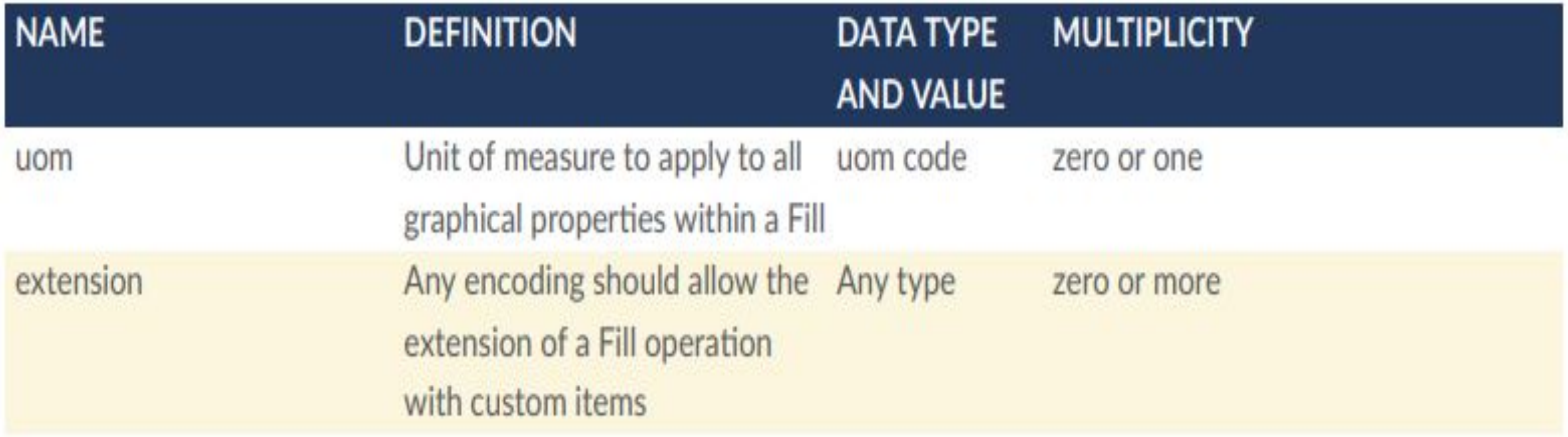
- The ColorClass allows the **definition of color**. As an abstract class and part of the base of the core graphical concepts, this class is a global point of extension for specifying concrete defintions of colors (e.g., RGBColor extension)



NAME	DEFINITION	DATA TYPE AND VALUE	MULTIPLICITY
extension	Any encoding should allow the extension of *ColorClass* with custom items	Any type	zero or more

Class Fill

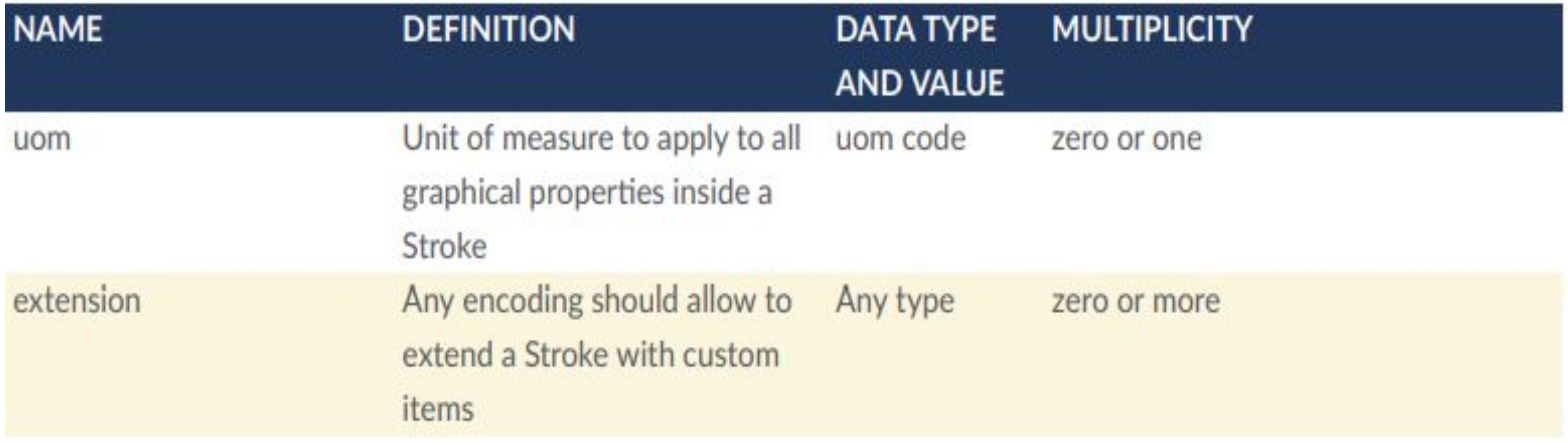
- FillClass defines the graphical symbolizing parameters required to **draw the filling of a twodimensional shape** such as a polygon. As an abstract class and part of the base of the core graphical concepts, FillClass is a global point of extension for specifying concrete definitions for shape fill operations (e.g., the SolidFill and GraphicFill extensions).



NAME	DEFINITION	DATA TYPE AND VALUE	MULTIPLICITY
uom	Unit of measure to apply to all graphical properties within a Fill	uom code	zero or one
extension	Any encoding should allow the extension of a Fill operation with custom items	Any type	zero or more

Class Stroke

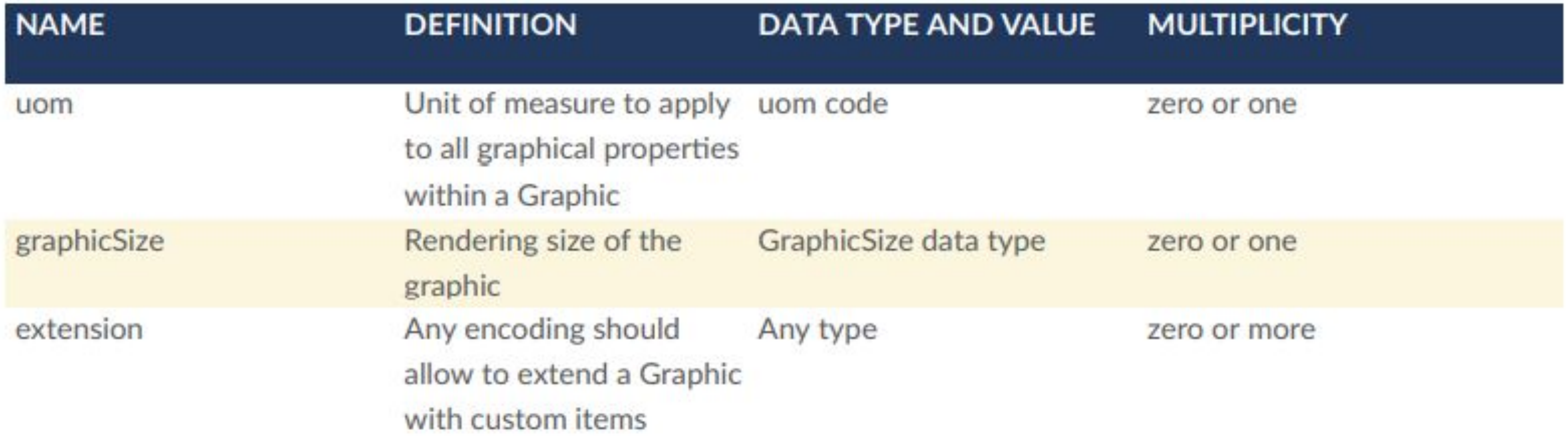
- StrokeClass defines the graphical symbolizing parameters for **drawing an outline** (e.g., for linear geometries or the exterior of a polygon geometry). As an abstract class and part of the base of the core graphical concepts, StrokeClass is a global point of extension to specify concrete ways to draw outlines (e.g., the PenStroke and GraphicStroke extensions).



NAME	DEFINITION	DATA TYPE AND VALUE	MULTIPLICITY
uom	Unit of measure to apply to all graphical properties inside a Stroke	uom code	zero or one
extension	Any encoding should allow to extend a Stroke with custom items	Any type	zero or more

Class Graphic and Graphic Size

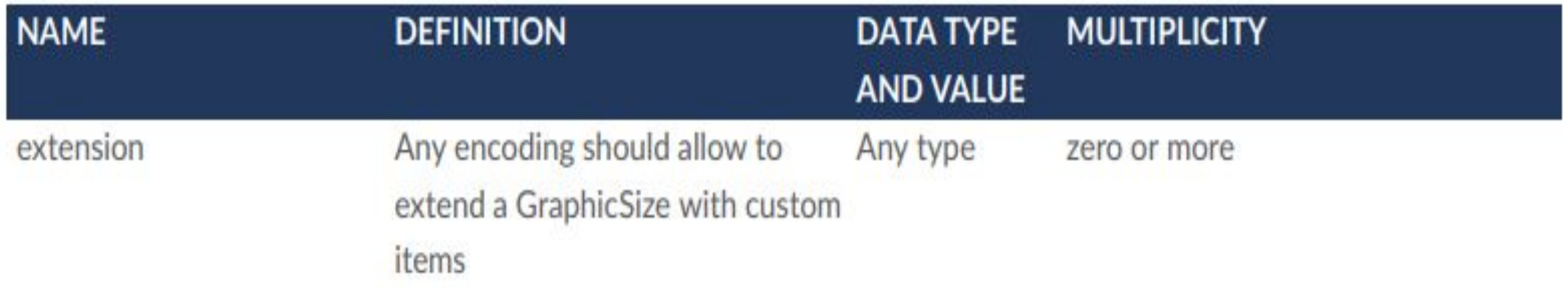
- The Graphic class defines the parameters for **drawing a graphic symbol such as shape, color(s), and size.**



NAME	DEFINITION	DATA TYPE AND VALUE	MULTIPLICITY
uom	Unit of measure to apply to all graphical properties within a Graphic	uom code	zero or one
graphicSize	Rendering size of the graphic	GraphicSize data type	zero or one
extension	Any encoding should allow to extend a Graphic with custom items	Any type	zero or more

Class Graphic and Graphic Size

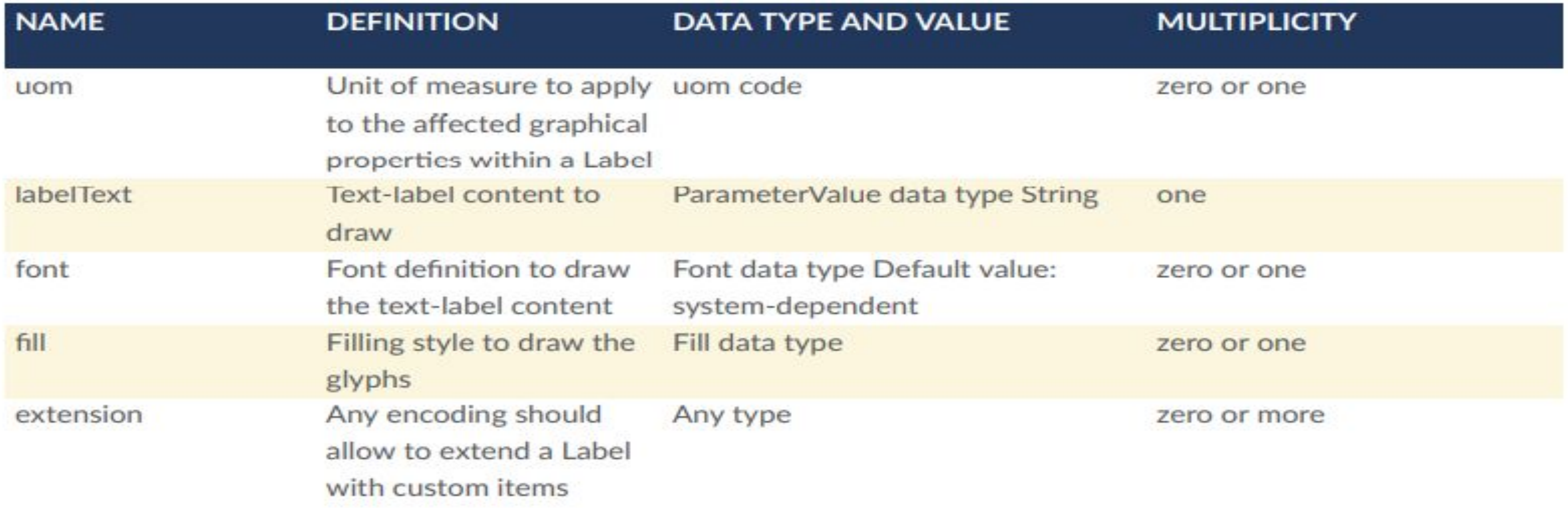
- The GraphicSize class determines the **size of the graphic** when it is rendered. As an abstract class, it is designed to be extended to support the various ways the size could be specified such as by a single value, a rectangular box, or by a three-dimensional cube



NAME	DEFINITION	DATA TYPE AND VALUE	MULTIPLICITY
extension	Any encoding should allow to extend a GraphicSize with custom items	Any type	zero or more

Class Label

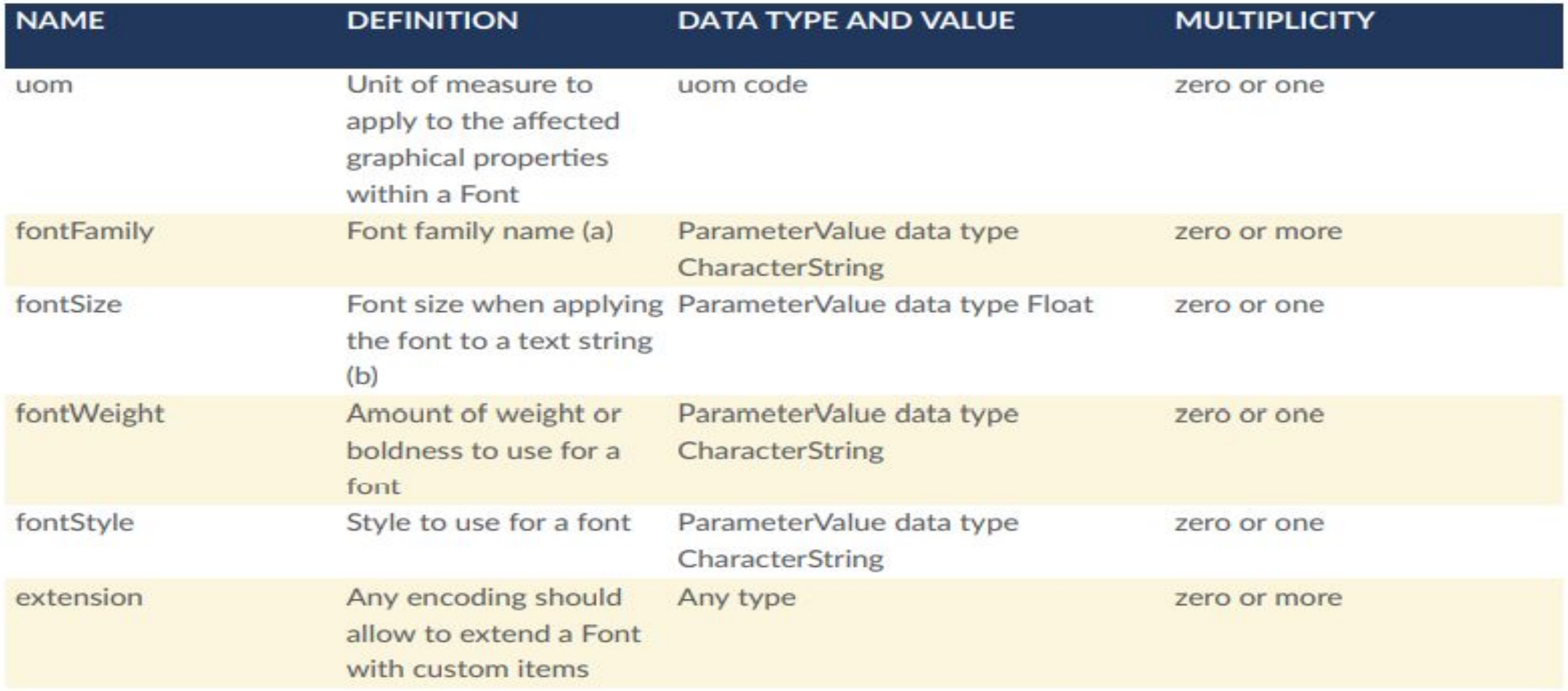
- LabelClass defines the graphical symbolizing properties for **drawing a text label**. As an abstract class and part of the base of the core graphical concepts, LabelClass is a point of extension to specify concrete ways to draw text label according to placement behaviors (e.g., a PointLabel or LineLabel).



NAME	DEFINITION	DATA TYPE AND VALUE	MULTIPLICITY
uom	Unit of measure to apply to the affected graphical properties within a Label	uom code	zero or one
labelText	Text-label content to draw	ParameterValue data type String	one
font	Font definition to draw the text-label content	Font data type Default value: system-dependent	zero or one
fill	Filling style to draw the glyphs	Fill data type	zero or one
extension	Any encoding should allow to extend a Label with custom items	Any type	zero or more

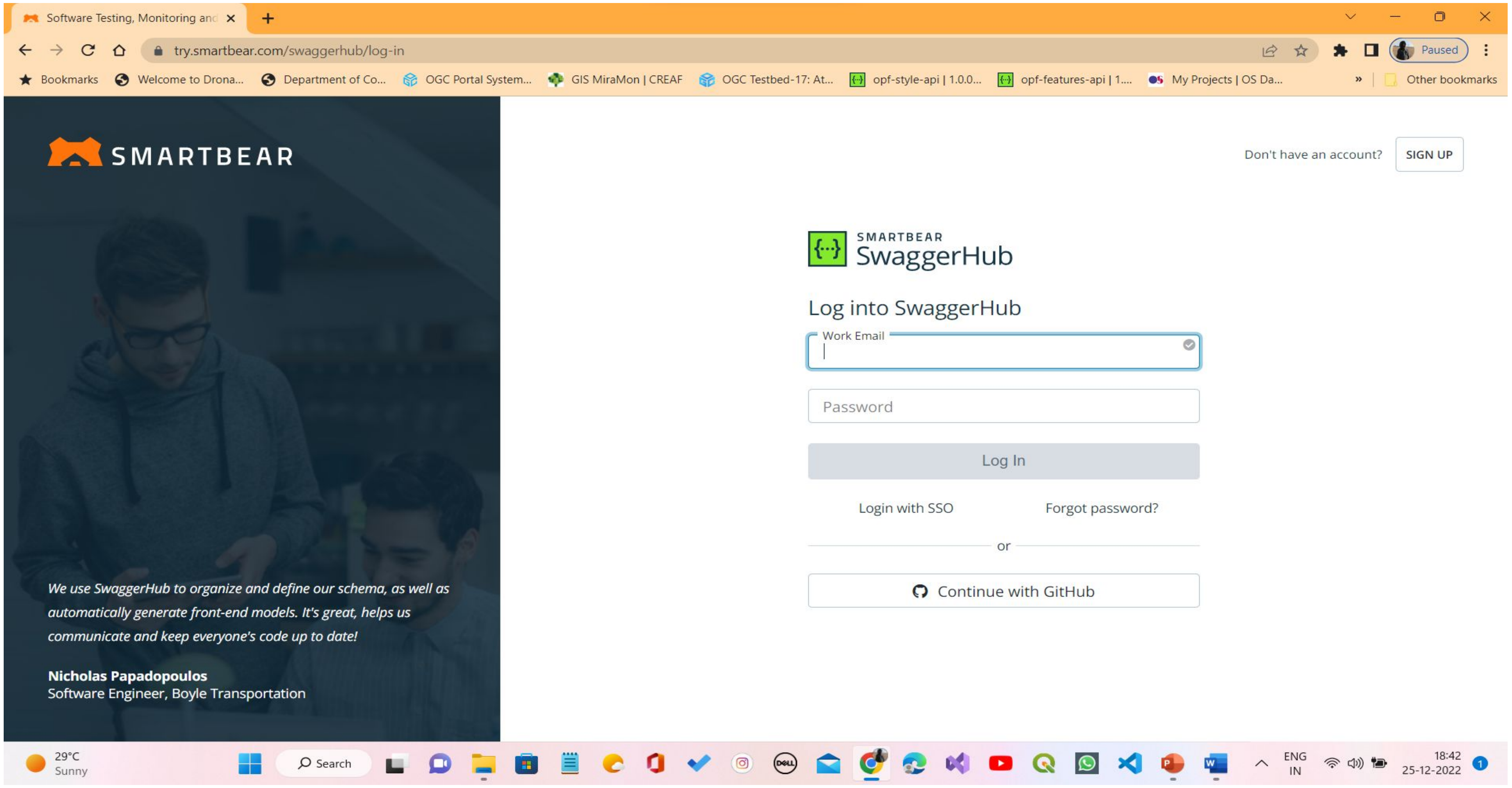
Class Font

- The FontClass describes the **font properties to apply for the rendering of a text** string



NAME	DEFINITION	DATA TYPE AND VALUE	MULTIPLICITY
uom	Unit of measure to apply to the affected graphical properties within a Font	uom code	zero or one
fontFamily	Font family name (a)	ParameterValue data type CharacterString	zero or more
fontSize	Font size when applying the font to a text string (b)	ParameterValue data type Float	zero or one
fontWeight	Amount of weight or boldness to use for a font	ParameterValue data type CharacterString	zero or one
fontStyle	Style to use for a font	ParameterValue data type CharacterString	zero or one
extension	Any encoding should allow to extend a Font with custom items	Any type	zero or more

Hands-on

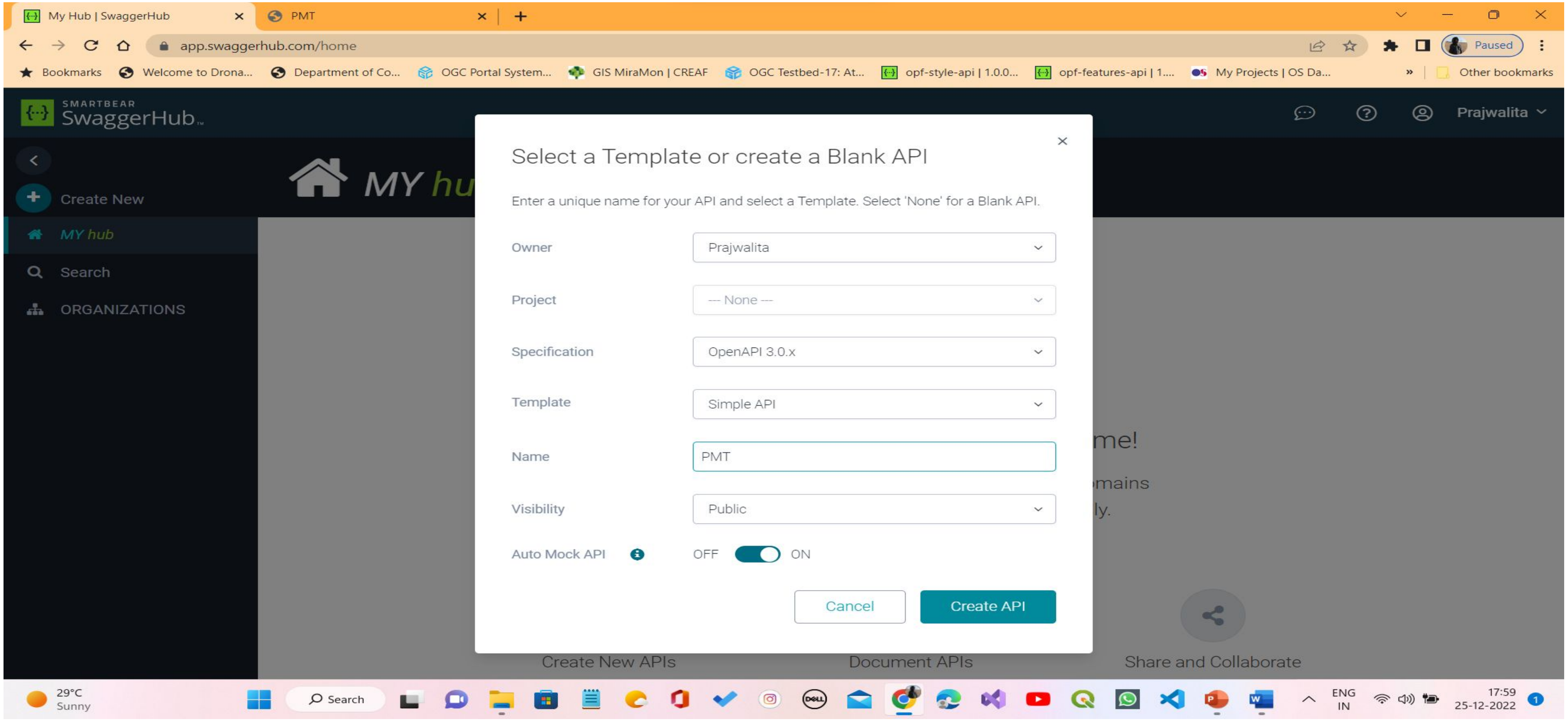


Log into SwaggerHub

We use SwaggerHub to organize and define our schema, as well as automatically generate front-end models. It's great, helps us communicate and keep everyone's code up to date!

Nicholas Papadopoulos
Software Engineer, Boyle Transportation

Hands-on

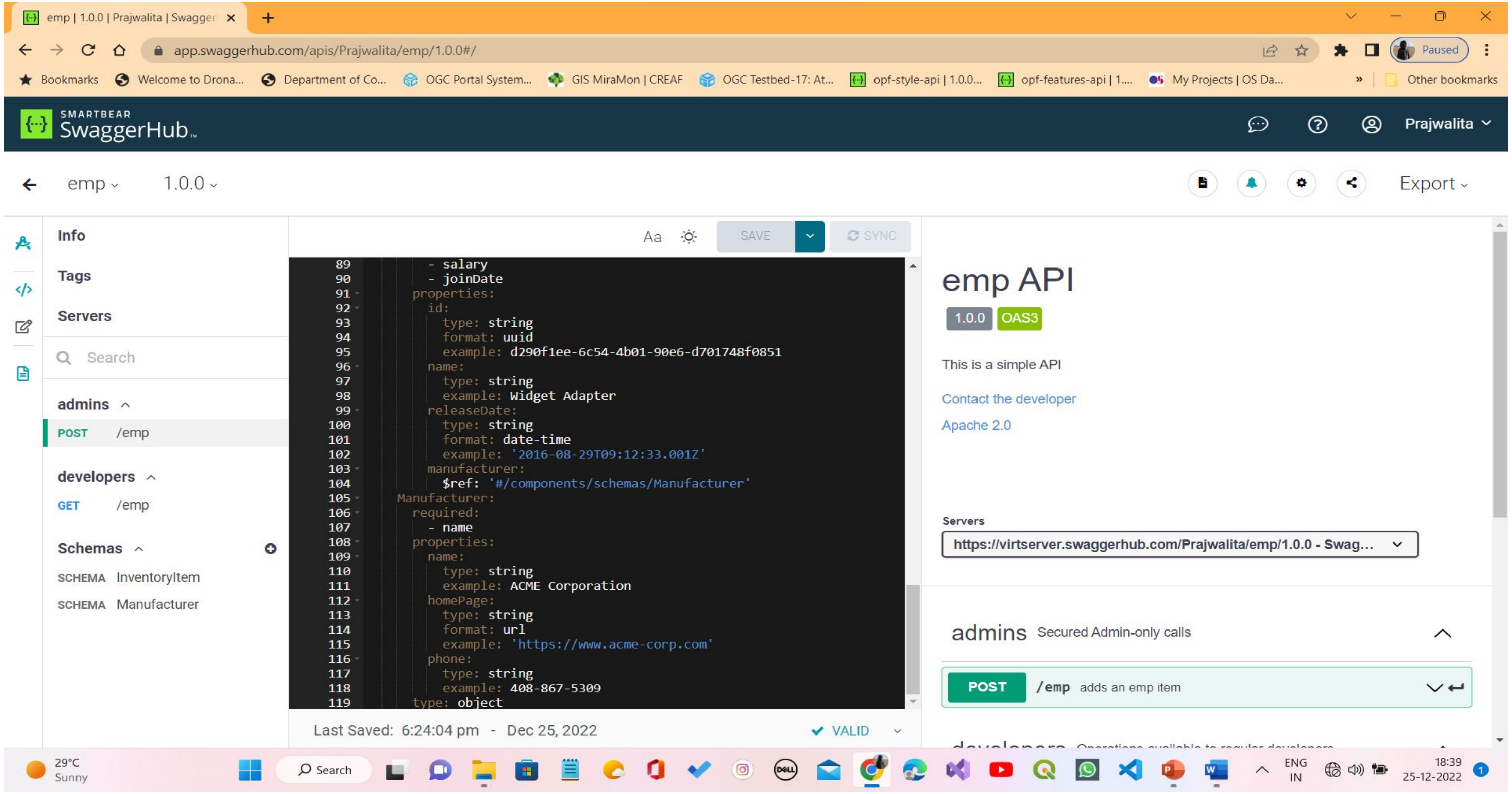


Select a Template or create a Blank API

Enter a unique name for your API and select a Template. Select 'None' for a Blank API.

Owner	Prajwalita
Project	--- None ---
Specification	OpenAPI 3.0.x
Template	Simple API
Name	PMT
Visibility	Public
Auto Mock API	OFF [✓ ON]

Hands-on


emp | 1.0.0 | Prajwalita | Swagger
app.swaggerhub.com/apis/Prajwalita/emp/1.0.0#/
SMARTBEAR SwaggerHub
Prajwalita
emp
1.0.0
Export
Info
Tags
Servers
Search
SAVE
SYNC

admins
POST /emp

developers
GET /emp

Schemas
SCHEMA InventoryItem
SCHEMA Manufacturer

```yaml
        - salary
        - joinDate
      properties:
        id:
          type: string
          format: uuid
          example: d290f1ee-6c54-4b01-90e6-d701748f0851
        name:
          type: string
          example: Widget Adapter
        releaseDate:
          type: string
          format: date-time
          example: '2016-08-29T09:12:33.001Z'
        manufacturer:
          $ref: '#/components/schemas/Manufacturer'
    Manufacturer:
      required:
        - name
      properties:
        name:
          type: string
          example: ACME Corporation
        homePage:
          type: string
          format: url
          example: 'https://www.acme-corp.com'
        phone:
          type: string
          example: 408-867-5309
      type: object
```

Last Saved: 6:24:04 pm - Dec 25, 2022 ✓ VALID

emp API
1.0.0 OAS3

This is a simple API

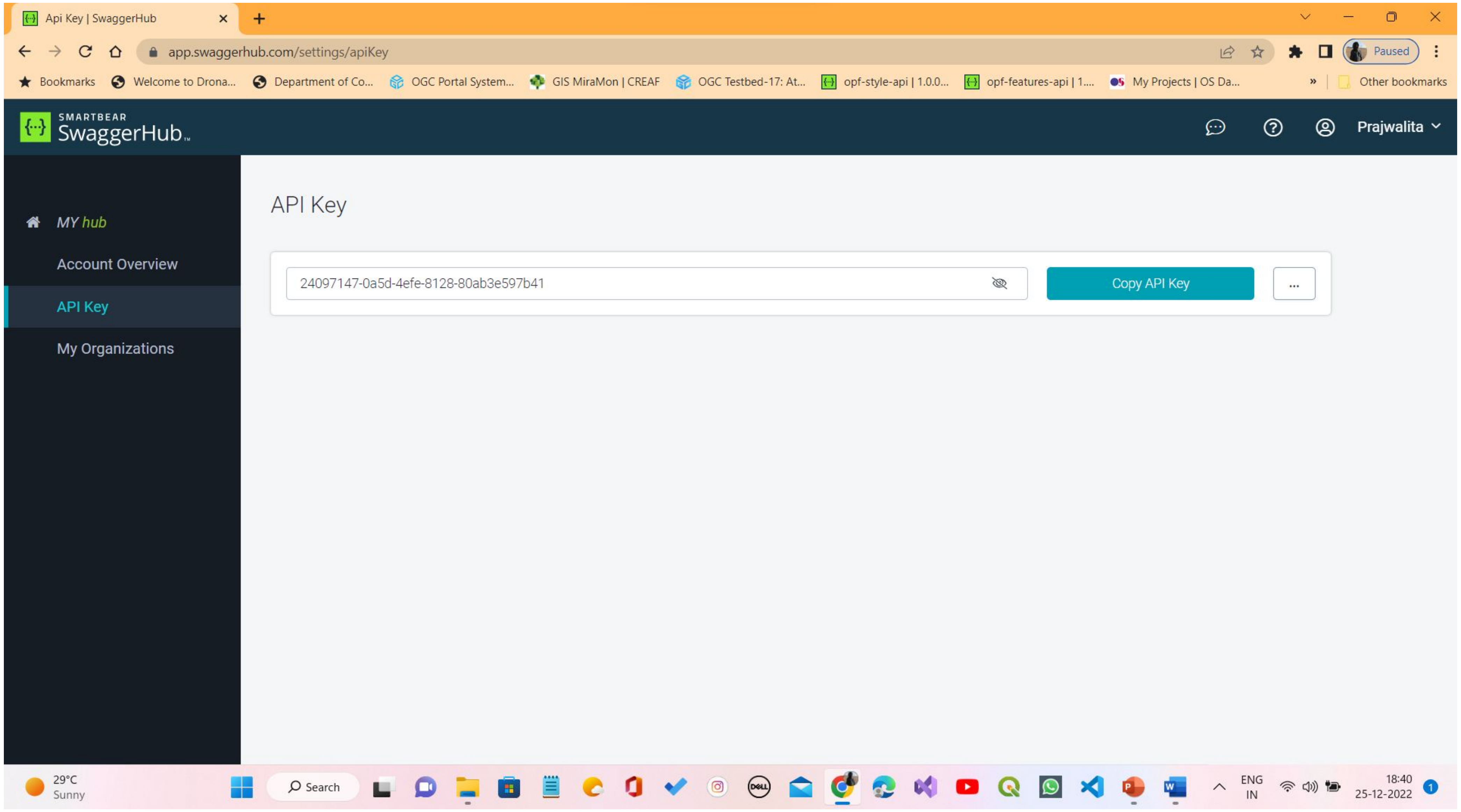
Contact the developer
Apache 2.0

Servers
https://virtserver.swaggerhub.com/Prajwalita/emp/1.0.0 - Swag...

admins Secured Admin-only calls

POST /emp adds an emp item

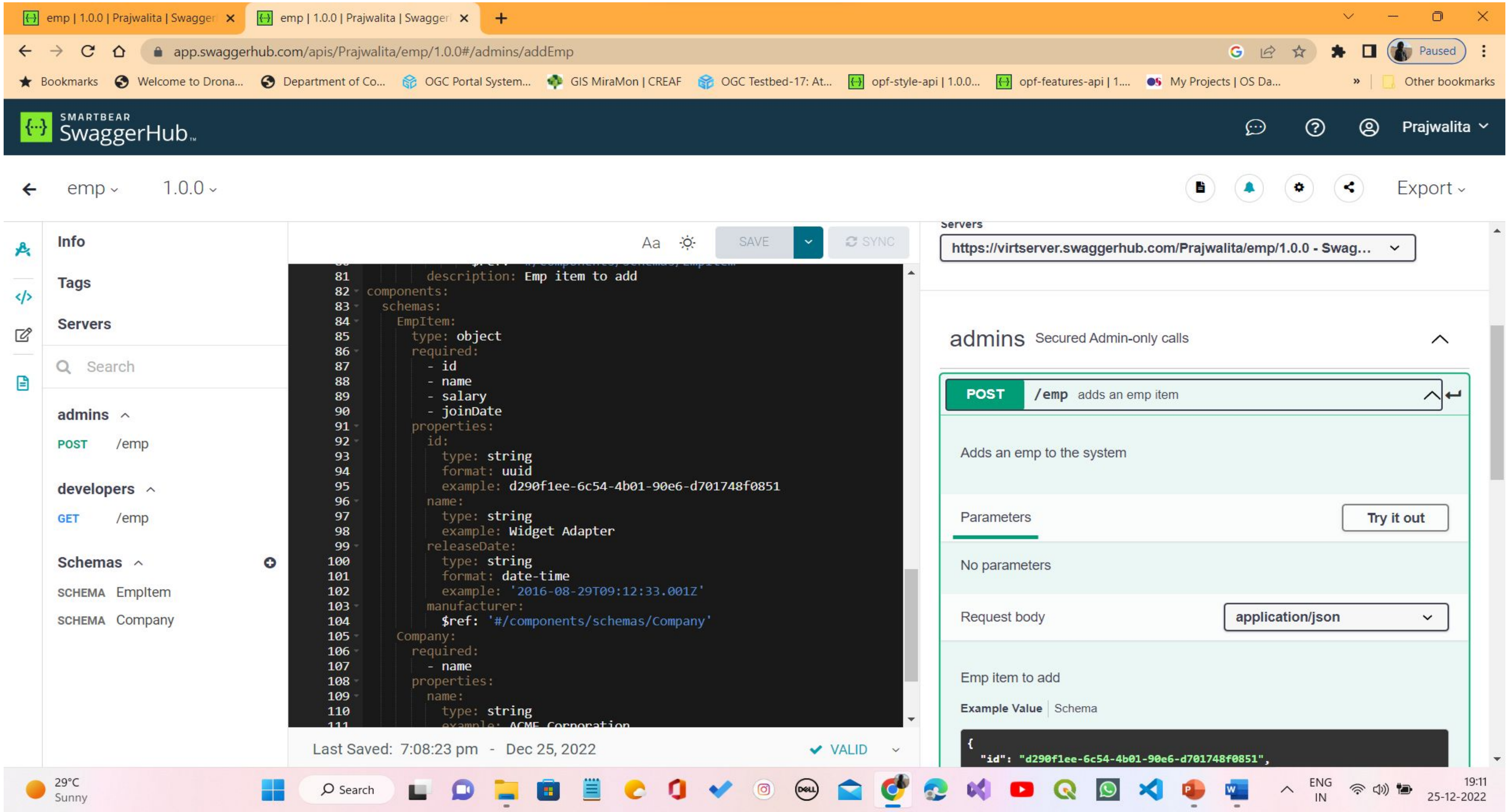
Hands-on



API Key

24097147-0a5d-4efe-8128-80ab3e597b41

Hands-on



emp | 1.0.0 | Prajwalita | Swagger
app.swaggerhub.com/apis/Prajwalita/emp/1.0.0#/admins/addEmp

emp 1.0.0

Info
Tags
Servers

admins
POST /emp
developers
GET /emp
Schemas
SCHEMA EmpItem
SCHEMA Company

```
          description: Emp item to add
components:
  schemas:
    EmpItem:
      type: object
      required:
        - id
        - name
        - salary
        - joinDate
      properties:
        id:
          type: string
          format: uuid
          example: d290f1ee-6c54-4b01-90e6-d701748f0851
        name:
          type: string
          example: Widget Adapter
        releaseDate:
          type: string
          format: date-time
          example: '2016-08-29T09:12:33.001Z'
        manufacturer:
          $ref: '#/components/schemas/Company'
    Company:
      required:
        - name
      properties:
        name:
          type: string
```

Last Saved: 7:08:23 pm - Dec 25, 2022 ✓ VALID

Servers
https://virtserver.swaggerhub.com/Prajwalita/emp/1.0.0 - Swag...

admins Secured Admin-only calls

POST /emp adds an emp item

Adds an emp to the system

Parameters
No parameters

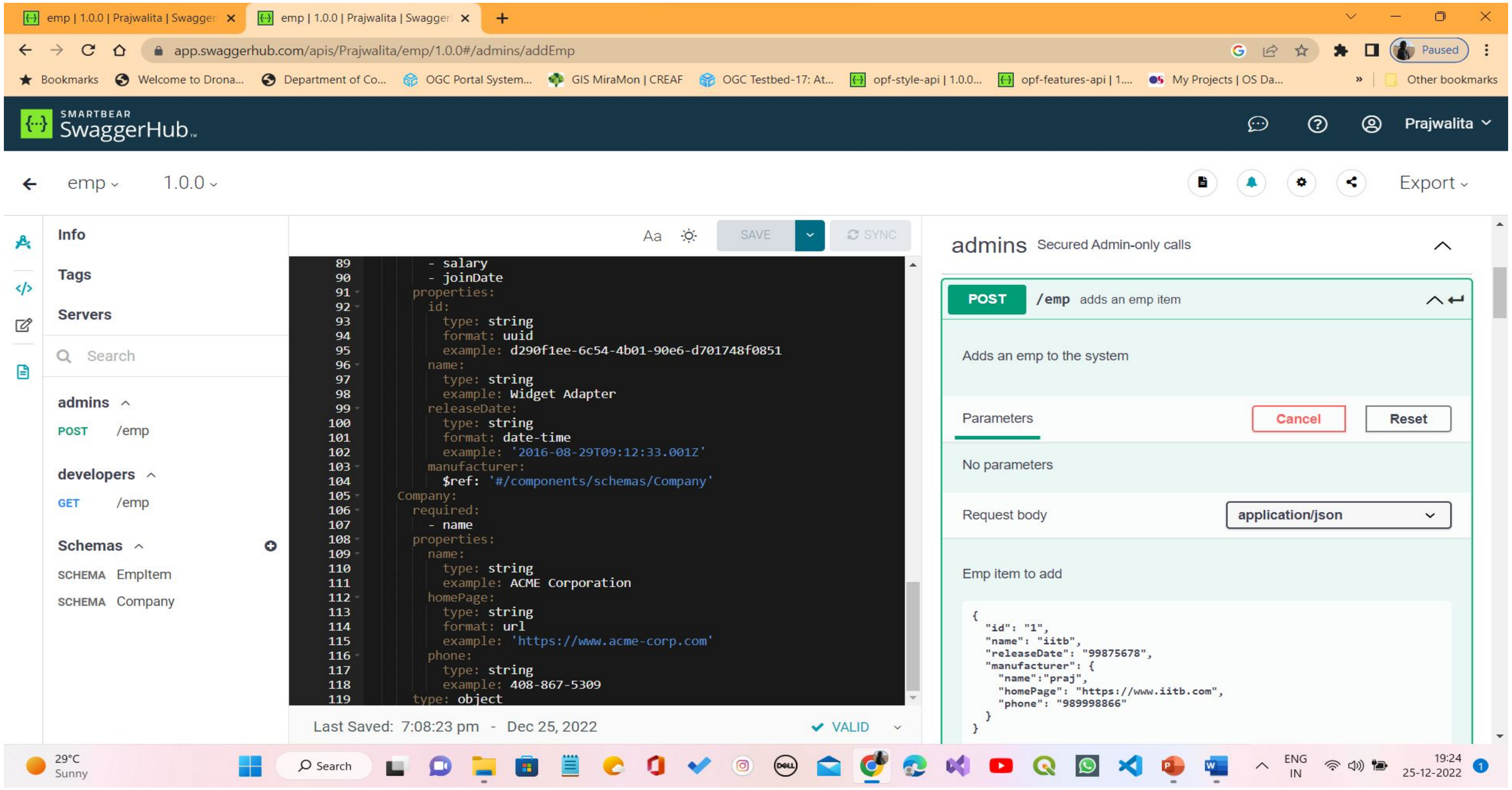
Request body application/json

Emp item to add

Example Value | Schema

```
{
  "id": "d290f1ee-6c54-4b01-90e6-d701748f0851",
```

Hands-on



```
            - salary
            - joinDate
          properties:
            id:
              type: string
              format: uuid
              example: d290f1ee-6c54-4b01-90e6-d701748f0851
            name:
              type: string
              example: Widget Adapter
            releaseDate:
              type: string
              format: date-time
              example: '2016-08-29T09:12:33.001Z'
            manufacturer:
              $ref: '#/components/schemas/Company'
        Company:
          required:
            - name
          properties:
            name:
              type: string
              example: ACME Corporation
            homePage:
              type: string
              format: url
              example: 'https://www.acme-corp.com'
            phone:
              type: string
              example: 408-867-5309
          type: object
```

Last Saved: 7:08:23 pm - Dec 25, 2022 ✓ VALID

admins Secured Admin-only calls

POST /emp adds an emp item

Adds an emp to the system

Parameters — Cancel — Reset

No parameters

Request body: application/json

Emp item to add

```
{
  "id": "1",
  "name": "iitb",
  "releaseDate": "99875678",
  "manufacturer": {
    "name":"praj",
    "homePage": "https://www.iitb.com",
    "phone": "989998866"
  }
}
```

THANK YOU!

prajwalita.chavan@gmail.com
prajwalita@iitb.ac.in

#OGCAPI