ETSI NFV SPECFEST

Hands on activities with NFV SOL specifications and the ETSI Forge

Presented by Michele Carignani (ETSI) for ETSI NFV#19 – Denver – Sept 11th 2017
NFV STAGE 3 APIS AND THE OPENAPI FILES
RESTful APIs and description Languages

RESTful: **resources** accessed with few **uniform operations**

**Main ingredients**
- A tree of resources (**paths**)  
- Supported operations (**methods**) for each resource  
- Exchanged **payloads** (parameters and request/response bodies)  
- (Plus authentication, headers, ...)
RESTful APIs are simple to use but…

How about a formal language to define and API to enable

- Automatic documentation and code generation
- Version control on API “blueprints”

Several initiatives to define an API description language

- WADL, RAML, OpenAPIs (Swagger), API Blueprint, Odata, RSDL, ….
Open API Specification (OAS) Language

A.k.a Swagger (previous name)
Now an initiative under the Linux Foundation
Machine readable specification of RESTful APIs

Syntax
• Tree based structure
• JSON based, can be described via YAML
• Reference for version 2.0

De facto standard
• Lively community of users and tooling developers
• RAML main contributors recently joined OpenAPI initiative (providing a converter tool among the languages)

www.openapis.org
OpenAPIs on NFV SOL APIs. Why?

Your favorite tool here

Base specs

DOCs

Source Code

Test Suites
OpenAPI specification

Info & location

Paths

/path1 + Method

Response Code 1

Response Code 2

Response Body Datatype

Response Header

Definitions

Datatype1

Parameter1

Parameter2

Response Body Datatype

Response Header

© ETSI 2017. All rights reserved
OAS Quick intro: Top level properties

- **Swagger**: Language version (Required)
- **Info**: API Metadata (Required)
- **Host**: API hostname or IP
- **basePath**: API basepath
- **Paths**: Resource tree exposed (Required), each paths contains
  - **Operations**: HTTP methods supported, each contains
    - **Responses**: HTTP status, headers and payloads of each possible response
- **Definitions**: Data types produced and consumed
  - Basic types: `string, integer, long, boolean, date, dateTime, password`
  - Complex types: `object` (with `properties`) and `arrays` (with `items`)
YAML Syntax basics

- **Whitespace indentation** gives scope to parts of documents
- **Comments** begin with `#` and end with the line return/feed
- **Ordered sequences** have each item prefixed with “- “
- **Maps** (lists of key-value pairs) use “:"
# GENERAL INFORMATION OF THE API

```json
swagger: '2.0'
info:
  title: NFV SOL 002 SpecFest - EXAMPLE
  version: v1
host: forge.etsi.org
basePath: /vnflcm

paths:
  /vnf_instances:
    post:
      description: Creates a new VNF instance resource.
      parameters:
        - name: createVnfRequest
          in: body
          description: VNF creation parameter
          schema:
            $ref: '#/definitions/CreateVnfRequest'
          required: true
      responses:
        '201':
          description: Created successfully.
          schema:
            $ref: '#/definitions/VnfInstance'

definitions:
  VnfInstance:
    description: TBD
    required:
      - id
      - vnfdId
    properties:
      id:
        description: Identifier of the VNF instance
        type: string
      vnfdId:
        description: The VNFD on which the VNF instance is based.
        type: '#/definitions/Identifier'
```

© ETSI 2017. All rights reserved
Vnflcm example: Resources and operations

```
swagger: '2.0'
info:
  title: NFV SOL 002 SpecFest - EXA
  version: v1
host: forge.etsi.org
basePath: /vnflcm
paths:
  /vnf_instances:
    post:
      description: Creates a new VNF instance resource.
      parameters:
        - name: createVnfRequest
          in: body
          description: VNF creation parameter
          schema:
            $ref: '#/definitions/CreateVnfRequest'
          required: true
      responses:
        '201':
          description: Created successfully.
          schema:
            $ref: '#/definitions/VnfInstance'
definitions:
  VnfInstance:
    description: TBD
    required: 
    - id
    - vnfdId
    properties:
      id:
        description: Identifier of
        type: string
      vnfdId:
        description: The VNFD on which the VNF instance is based.
        type: $ref: '#/definitions/Identifier'
```
Vnflcm example: Data definitions

```
swagger: '2.0'
info:
  title: NFV SOL 002 SpecFest - EXAMPLE
  version: v1
host: forge.etsi.org
basePath: /vnflcm
paths:
  /vnf_instances:
    x-etsinfv-description: Represents VNF Instances.
    post:
      description: Creates a new VNF instance.
      parameters:
        - name: createVnfRequest
          in: body
          description: VNF creation parameters
          schema:
            $ref: '#/definitions/CreateVnfRequest'
          required: true
      responses:
        '201':
          description: Created successfully.
          schema:
            $ref: '#/definitions/VnfInstance'

# DATATYPES (SCHEMAS) DEFINITIONS

definitions:
  VnfInstance:
    description: TBD
    required:
      - id
      - vnfdId
    properties:
      id:
        description: Identifier of the VNF instance
        type: string
      vnfdId:
        description: The VNFD on which the VNF instance is based.
        type: '#/definitions/VndIdentifier'
  CreateVnfRequest:
    description: VNF creation parameters
    required:
      - id
      - vnfdId
    properties:
      id:
        description: Identifier of the VNF instance
        type: string
      vnfdId:
        description: The VNFD on which the VNF instance is based.
        type: '#/definitions/VndIdentifier'
```

© ETSI 2017. All rights reserved
OpenAPIs Language

… Several Open Source and commercial tools
(non exhaustive list)

SMARTBEAR Tools (OSS)
- Codegen
- Editor
- UI
- … Others

swagger.io
Editing online

- Swagger Editor: in browser editor for OAS
- Released under Apache 2.0
- Try it out at forge.etsi.org/swagger/editor
- Will be used in our hands on activity!
Test suite generation example

More details to come during the next demo!
HANDS ON: OAS AT ETSI FORGE

How to create OAS files in a collaborative way at the ETSI FORGE
Describes RESTful protocols specification for the Ve-Vnfm Reference Point

Contains specification of 5 interfaces

- VNF Life Cycle Management
- VNF Performance Management
- VNF Fault Management
- VNF Indicator
- VNF Configuration
SpecFest workflow

1. Draft API description
2. Upload
3. Aut. syntax and model validation
4. Approval
5. Publication

- Verify (+1)
- Reviewed (+2)
- Public repo
- ETSI FORGE
Outlined of the proposed task

1. Read the detailed instructions on the wiki page
2. Open online editor (link provided in the wiki) and fill in the required information taken from the SOL002 extract provided. Leave the browser window/tab open when you are finished!
3. Upload the produced specification to the ETSI Forge
4. Wait for the automatic validator to execute and verify your solution. Fix any errors and submit again if needed.
Live demo

Navigate to
forge.etsi.org/specfest
Contact Details:
Michele Carignani, Silvia Almagia
Centre for Testing and Interoperability, ETSI
michele.carignani@etsi.org
silvia.almagia@etsi.org

ANY QUESTIONS?

Thank you!