WEB SERVICE API

SMDG Work Group

Tamme Bohlen – Hapag-Lloyd AG

17.09.2019, Singapore

AGENDA

Web Service API Work Group

- Introduction
- Risks & Opportunities for WS API
- Status WS Standardisation
  - SMDG
  - UN/CEFACT
  - DCSA
- SMDG Pilot: Voyage Schedule API

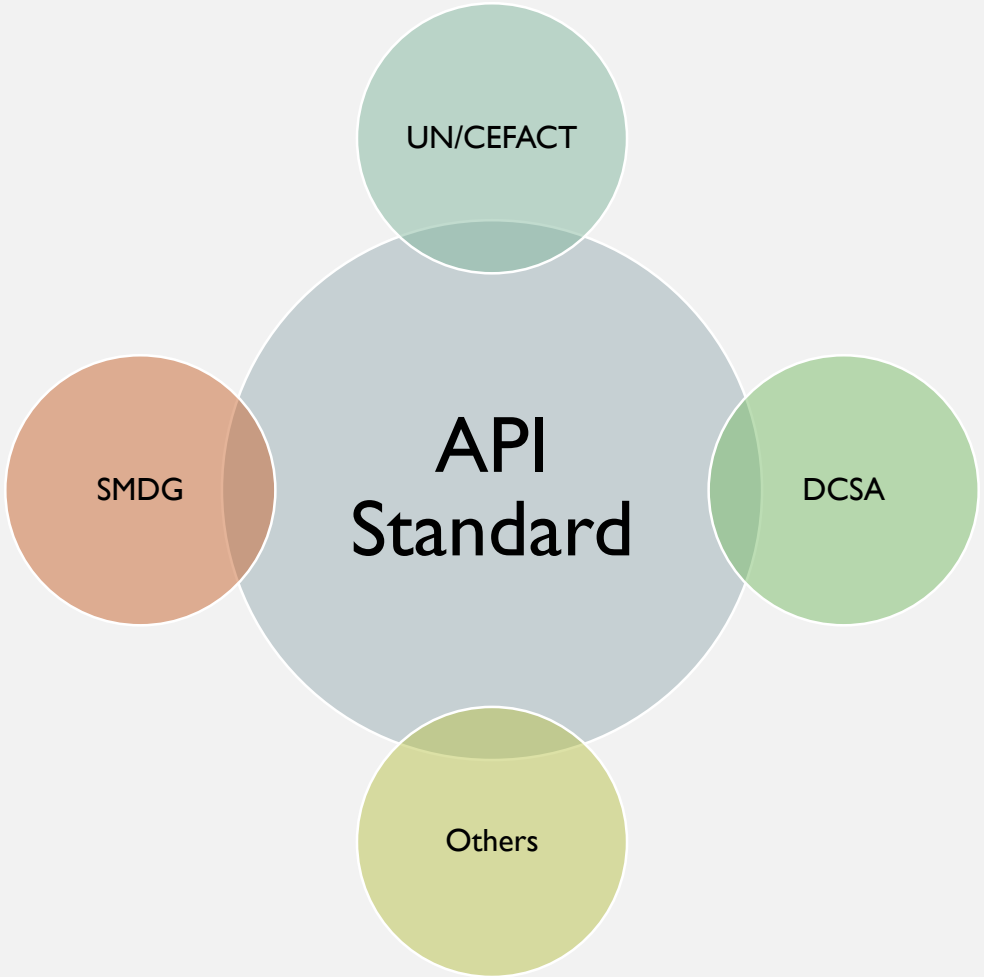
PERSONAL INTRODUCTION

- Tamme Bohlen
- IT Project Manager

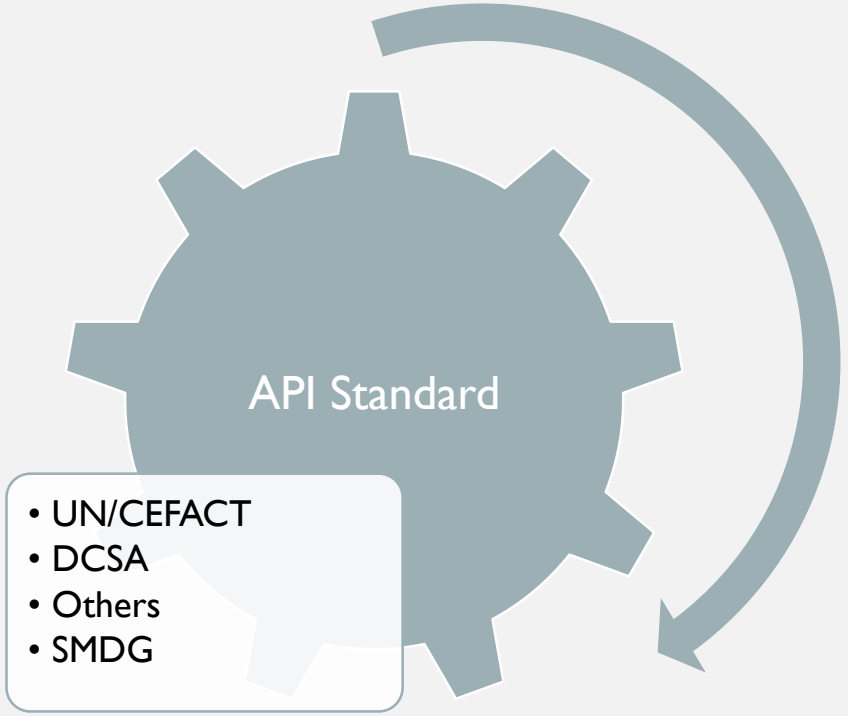
- Hapag-Lloyd AG
- Chairman Web Service API Work Group
- UN/CEFACT EDI2API
- DCSA
- MBA at Kühne Logistics University

WEB SERVICE API STANDARDISATION ORGANISATIONS

RISK: MULTIPLE STANDARDS



OUR GOAL: ONE STANDARD



HURDLES

- No practical experience with WS API standards
- No agreed Data Model
- Our industry lacks technical know-how
- Many Software frameworks not ready

UPCOMING HURDLES

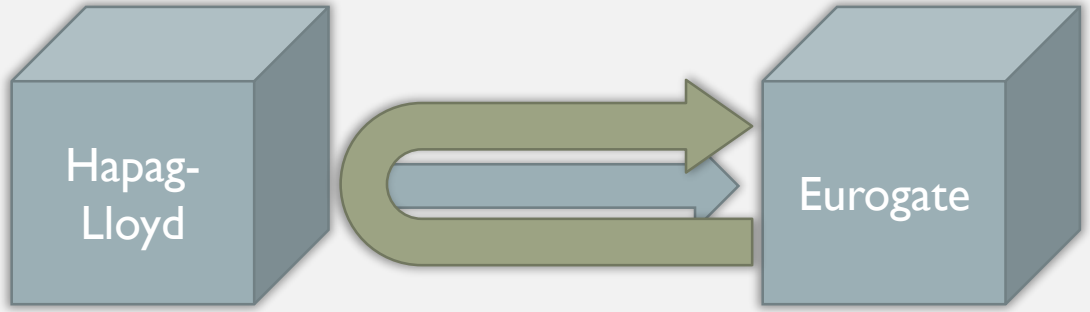
UPCOMING HURDLES

- Requirements and use case identification
  - What does the user need?
- How to implement an API?
  - Effort? (vs. EDIFACT)
- Data model updates
- Agile changes
- Versioning
- Performance
- Security

OPPORTUNITIES

- Synchronous data exchange
- Data on demand
- Direct system integration

→ Example: API Container Announcement



CURRENT STATUS

- SMDG
- UN/CEFACT
- DCSA

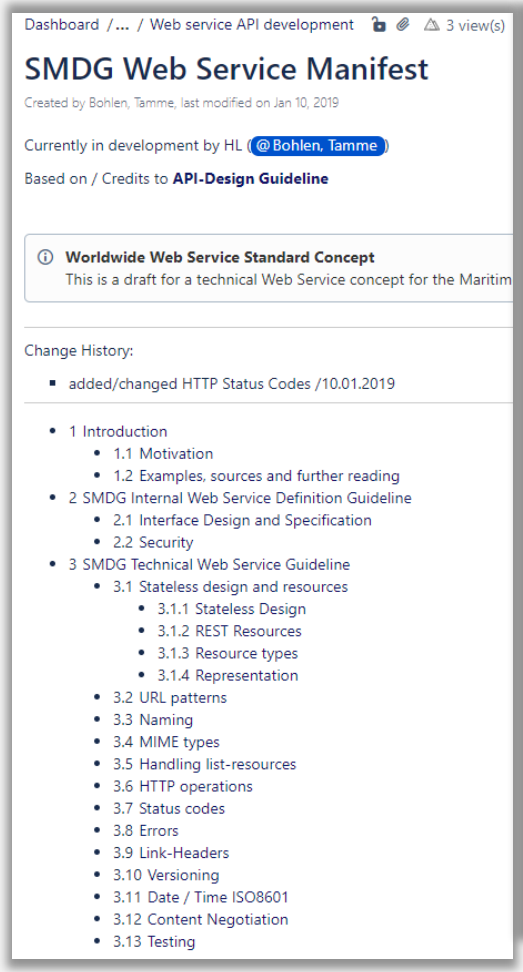
STATUS SMDG I

- First technical guideline finished
- First pilot created

- Voyage Schedule API
  - Use-case: Carrier Voyage Schedule for Terminals
  - Goal of first pilot: gather experience, learning by doing

- More to come in the workshop on Thursday!

STATUS SMDG II



Dashboard / ... / Web service API development 3 view(s)

SMDG Web Service Manifest

Created by Bohlen, Tamme, last modified on Jan 10, 2019

Currently in development by HL (@Bohlen, Tamme)

Based on / Credits to **API-Design Guideline**

Worldwide Web Service Standard Concept
This is a draft for a technical Web Service concept for the Maritim

Change History:

- added/changed HTTP Status Codes /10.01.2019

Errors

Errors **MUST** be in the format described in RFC 7807 with the media types *application/problem+json or application/problem+xml.*

Example for an error (application/problem+json)

```
{
  "type": "http://example.de/problem/service-unavailable",
  "title": "Service Unavailable",
  "status": 503,
  "detail": "Connection to database timed out"
}
```

The format **MAY** be extended so that the API can express all error situations that are needed.

SMDG recommends web service gateways to inject a global transaction ID into every request. This ID can be found in the HTTP header `X-Global-Transaction-ID`. This ID **MUST** be included in every error response. If a client experiences an error t
reason for the error and thus help to improve the API.

An API **MUST NOT** in any way display implementation details to the client.

→ More Information

Link-Headers

In order to enable API users to navigate through associated entities developers are encouraged to create appropriate links.

- When creating an entity the corresponding location **MUST** be sent in the `Location` header of the HTTP response.

  Example Location-Header

- Retrieving entity information **SHOULD** also return a `Link` header in the HTTP response to refer to associated entities.

  Example Link-Header

- When implementing a list resource links **MAY** be used to directly link to listed entities.

→ More Information

Versioning

SMDG recommends versioning via URL. This means that every API URL will contain the version of that specific API.

example

```
https://api.hlag.com/mobile/smdg2/container
```

→ More Information

Date / Time ISO8601

ISO8601 **MUST** be used as the standard date and time format. This format includes date, time *and timezone.*

It is possible to only use parts of the timestamp format. You **MAY** combine date and time values or just use one if them alone.

2017-06-08T15:49:47,123456789+02:00

Don't use it if you don't need it
ISO 8601 also allows us the flexibility to provide a date without a time. In scer
return the time.

While it seems like no harm done in just storing 11:59pm, or some other rand
that date.

STATUS UN/CEFACT I

- UN/CEFACT Forum in Geneva in April 2019
- Agreed on SMDG Pilot (EDI2API)
- EDIFACT to Web Service API

- Other API Projects:
  - RDM to API
  - API Town plan

STATUS UN/CEFACT II

- Edi3.org / GitHub / Slack

- Why EDI3?
  - EDI1 – EDIFACT
  - EDI2 – XML
  - EDI3 – Web Service API

STATUS DCSA

- More carriers joined the DCSA
- First planned milestone: One API for Tracking
- Focus on data model and industry blueprint
- First publications to come
  - Will be published on dcsa.org
  - Comments are possible directly on the webpage

API TECH AND SMDG PILOT

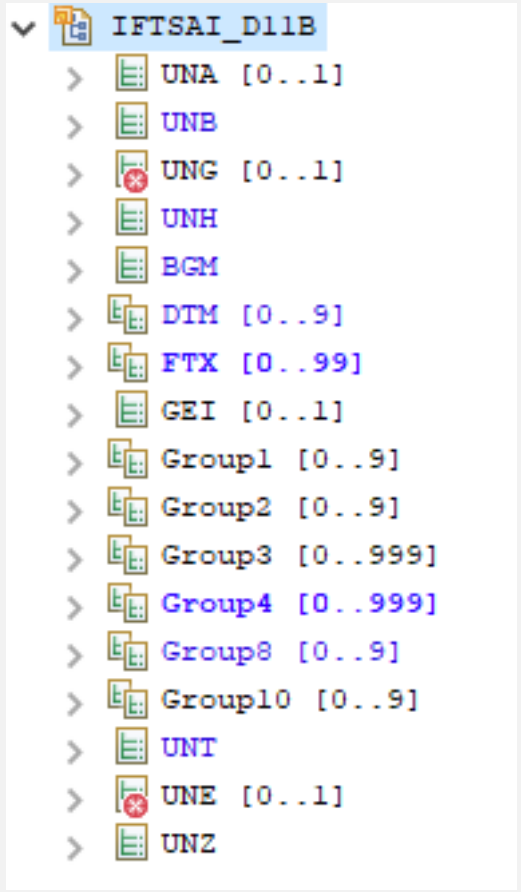
TECH BEHIND API

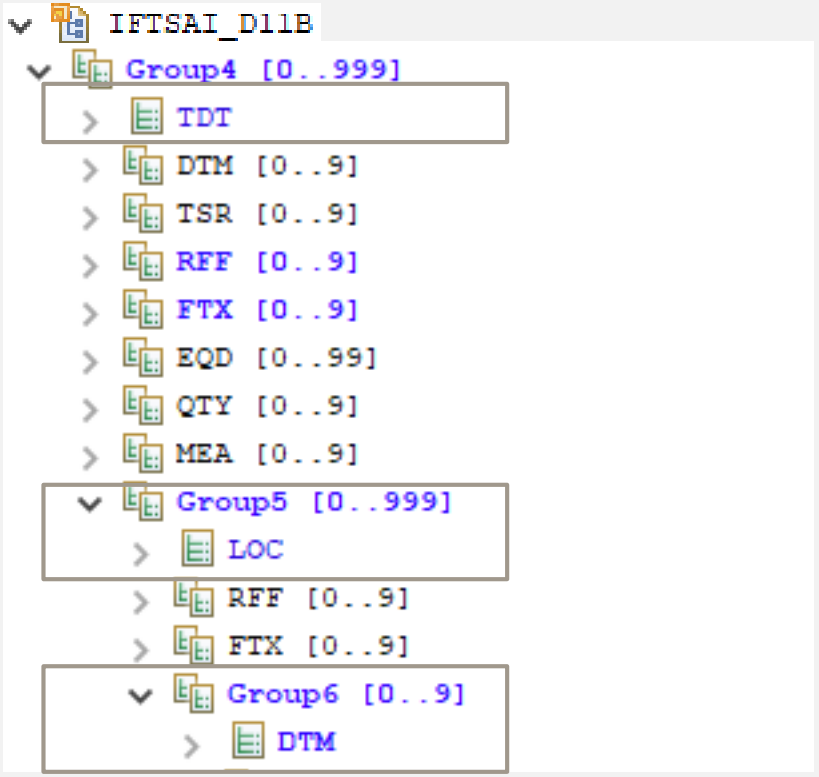
- What‘s an API?
  - EDIFACT vs Web Service
- What‘s a Web Service API?
  - Based on HTTP
- Which technology & design is connected?
  - Data model
  - Swagger
- Examples
  - SkyScanner
  - Google Flights

SMDG PILOT

- EDIFACT Data Model → Voyage Schedule API

PILOT NO.1: VESSEL SCHEDULE API





Vessel and Voyage

1

0..999

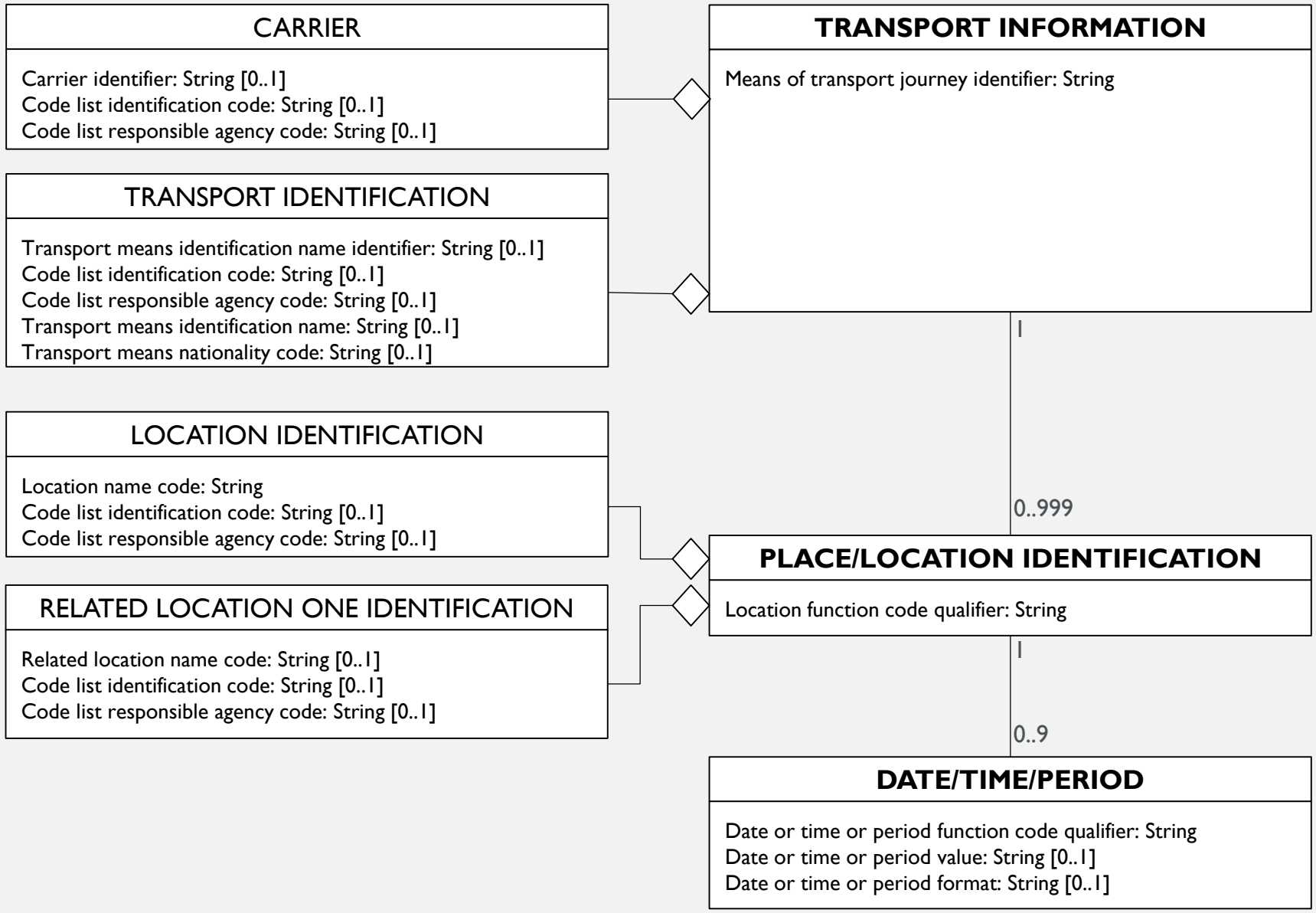
Location

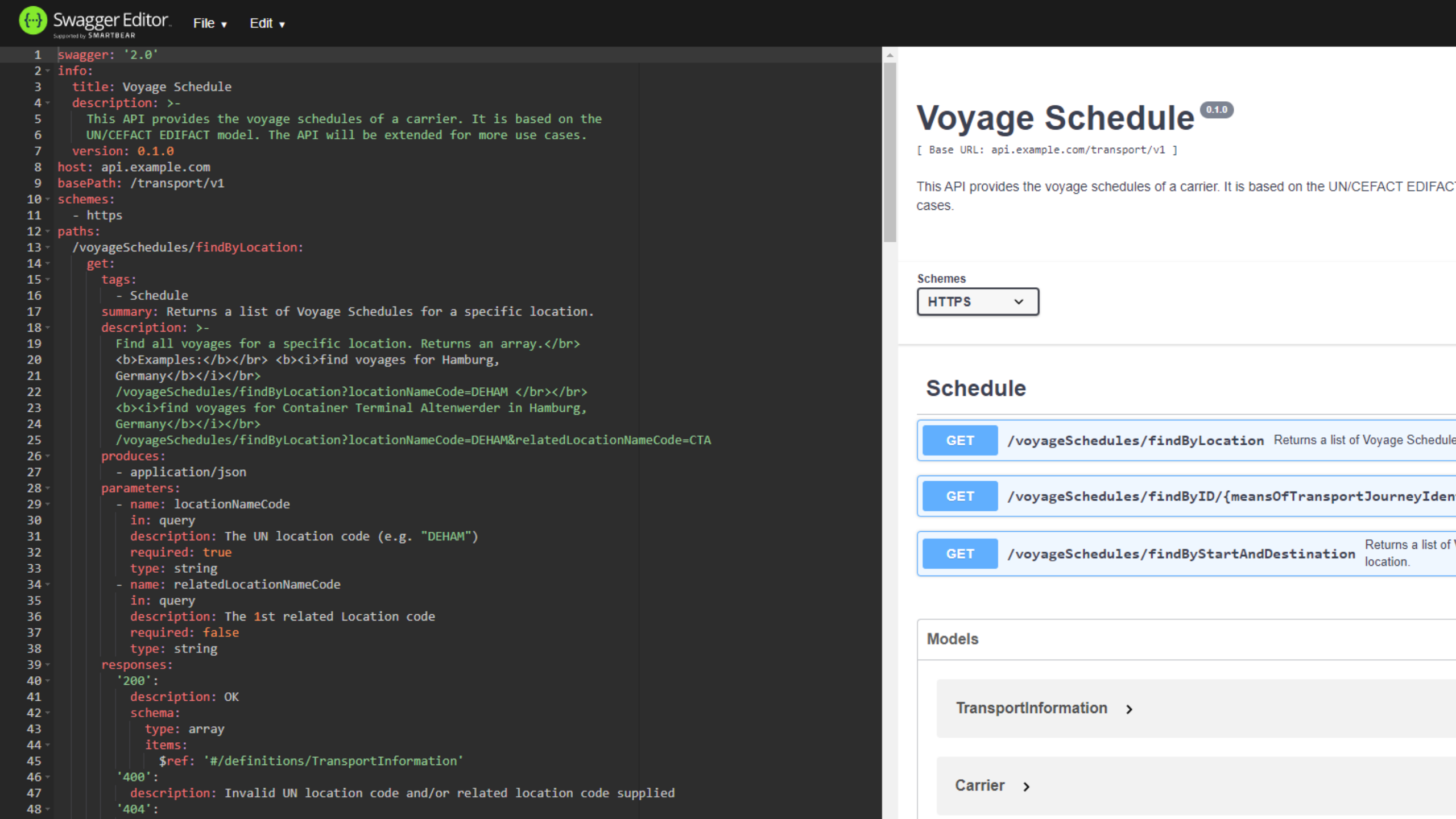
1

0..9

Date/Time

PILOT NO.1: VESSEL SCHEDULE API




Swagger Editor
Supported by SMARTBEAR
File
Edit
swagger: '2.0'
info:
title: Voyage Schedule
description: >-
This API provides the voyage schedules of a carrier. It is based on the
UN/CEFACT EDIFACT model. The API will be extended for more use cases.
version: 0.1.0
host: api.example.com
basePath: /transport/v1
schemes:
- https
paths:
/voyageSchedules/findByLocation:
get:
tags:
- Schedule
summary: Returns a list of Voyage Schedules for a specific location.
description: >-
Find all voyages for a specific location. Returns an array.</br>
Examples:</br> find voyages for Hamburg,
Germany</br>
/voyageSchedules/findByLocation?locationNameCode=DEHAM </br></br>
find voyages for Container Terminal Altenwerder in Hamburg,
Germany</br>
/voyageSchedules/findByLocation?locationNameCode=DEHAM&relatedLocationNameCode=CTA
produces:
- application/json
parameters:
- name: locationNameCode
in: query
description: The UN location code (e.g. "DEHAM")
required: true
type: string
- name: relatedLocationNameCode
in: query
description: The 1st related Location code
required: false
type: string
responses:
'200':
description: OK
schema:
type: array
items:
$ref: '#/definitions/TransportInformation'
'400':
description: Invalid UN location code and/or related location code supplied
'404':
Voyage Schedule
0.1.0
[Base URL: api.example.com/transport/v1]
This API provides the voyage schedules of a carrier. It is based on the UN/CEFACT EDIFAC
cases.
Schemes
HTTPS
Schedule
GET
/voyageSchedules/findByLocation
Returns a list of Voyage Schedule
GET
/voyageSchedules/findByID/{meansOfTransportJourneyIden
GET
/voyageSchedules/findByStartAndDestination
Returns a list of
location.
Models
TransportInformation
Carrier

WORKSHOP

- Workshop on Thursday
- Goals:
  - Exchange learnings
  - gather requirements for Voyage Schedule API