ETSI EG 203 647 V1.1.1 (2020-11)

Methods for Testing and Specification (MTS);
Methodology for RESTful APIs specifications and testing

Reference
DEG/MTS-203647

Keywords
API, methodology, testing

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Important notice

The present document can be downloaded from:
http://www.etsi.org/standards-search

The present document may be made available in electronic versions and/or in print. The content of any electronic and/or print versions of the present document shall not be modified without the prior written authorization of ETSI. In case of any existing or perceived difference in contents between such versions and/or in print, the prevailing version of an ETSI deliverable is the one made publicly available in PDF format at www.etsi.org/deliver.

Users of the present document should be aware that the document may be subject to revision or change of status. Information on the current status of this and other ETSI documents is available at https://portal.etsi.org/TB/ETSIDeliverableStatus.aspx

If you find errors in the present document, please send your comment to one of the following services:
https://portal.etsi.org/People/CommiteeSupportStaff.aspx

Contents

Intellectual Property Rights 5
Foreword 5
Modal verbs terminology 5
Executive Summary 5
Introduction 5
1 Scope 7
2 References 7
2.1 Normative references 7
2.2 Informative references 7
3 Definition of terms, symbols and abbreviations 10
3.1 Terms 10
3.2 Symbols 11
3.3 Abbreviations 11
4 Specification methodologies for RESTful APIs 12
4.1 RESTful APIs specification in a staged standardization approach 12
4.2 Introduction on RESTful interfaces 13
4.2.1 Introduction 13
4.2.2 Main Principles of the RESTful paradigm 13
4.2.3 HTTP Methods and their Usage 14
4.2.3.1 Overview 14
4.2.3.2 POST 14
4.2.3.3 GET 14
4.2.3.4 PUT/PATCH 14
4.2.3.5 DELETE 15
4.2.4 Error Reporting 15
4.2.4.1 Overview 15
4.2.4.2 Client Errors 16
4.2.4.3 Server Errors 16
4.3 API specification process 17
4.3.1 RESTful interface description languages 17
4.3.2 Standardizing RESTful interfaces using OpenAPI 17
4.3.2.1 OpenAPI Overview 17
4.3.2.2 Document 18
4.3.2.3 Data types 18
4.3.2.4 Operations 19
4.3.2.5 Requests 20
4.3.2.6 Responses 21
4.3.2.7 Callbacks 22
4.3.2.8 Query parameters 23
4.3.2.9 Extensions 23
4.3.2.10 Other 24
4.3.2.11 Process 25
4.4 Common Patterns 26
4.4.1 Filtering Patterns 26
4.4.1.1 Overview 26
4.4.1.2 Attribute-based filtering for collections 26
4.4.1.3 Attribute Selector 27
4.4.1.4 Pagination 28
4.4.2 Pattern for URI Creation 28
4.4.2.1 Resource URI Structure 28
4.4.2.2 Design Rules for REST API URI 29
4.4.3 Pattern to avoid Update Conflict and Data loss 29
4.4.3.1 Description 29

4.4.4 Authorization and Authentication 30
4.4.4.1 Overview 30
4.4.4.2 API Authorization using OAuth 2.0 Access tokens 31
4.4.4.3 API Authorization using TLS Certificates 31
4.4.4.4 API Authorization using OpenID connect with JWT ID Token 32
4.4.5 Non-CRUD operations 33
4.4.5.1 Description 33
4.5 Naming conventions 34
4.6 Versioning 36
4.6.1 Specifications and OpenAPI definitions versions 36
4.6.2 Modelling version information 37
4.7 Implementation 37

5 Testing methodology for REST APIs 38
5.1 Overview 38
5.2 Testing Frameworks and Methodologies 38
5.3 Conformance and Interoperability Testing 39
5.3.1 General 39
5.3.2 RESTful API-specific 40
5.3.3 Domain-specific 40
5.4 Test Specification Development 40
5.4.1 General 40
5.4.2 RESTful API-specific 42
5.4.3 Domain-specific 52
5.5 Test Deployment and Execution 52
5.6 Test Maintenance and Evolution 53

6 Tooling recommendations 53
6.1 Introduction 53
6.2 Design and drafting 53
6.2.1 Overview 53
6.2.2 Recommendations on editing tool selection 54
6.3 Coordination and collaboration 54
6.4 Validation and quality check 54
6.5 Post processing 55

7 Working Examples 56

8 Survey of Activities at ETSI and Beyond 56
8.1 Review of base documents 56
8.1.1 ETSI GS MEC 009 (V2.1.1) 56
8.1.2 ETSI GR MEC-DEC 025 (V2.1.1) 57
8.1.3 Draft ETSI GS MEC-DEC 032-1 (V0.0.3) 57
8.1.4 ETSI GS CIM 009 (V1.2.1) 57
8.1.5 ETSI GS QKD 014 (V1.1.1) 57
8.1.6 ETSI TS 129 501 (V15.3.0) 57
8.1.7 ETSI GS NFV-SOL 013 (V2.7.1) 58
8.1.8 ETSI GS NFV-SOL 015 (V1.1.1) 58
8.1.9 ETSI GS NFV-TST 010 (V2.4.1) 58
8.1.10 ETSI TS 118 115 (V2.0.0) 58
8.1.11 oneM2M TS-0018 (V3.2.0) 59
8.1.12 TM Forum Open APIs initiative 59
8.1.13 OMG hData RESTful Transport 59
8.1.14 OASIS OData v4.01 59
8.2 API adoption survey 60

Annex A (informative): Bibliography 62

Annex B (informative): Change History 63

History 64

Intellectual Property Rights

Essential patents

IPRs essential or potentially essential to normative deliverables may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (https://ipr.etsi.org/).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Trademarks

The present document may include trademarks and/or tradenames which are asserted and/or registered by their owners. ETSI claims no ownership of these except for any which are indicated as being the property of ETSI, and conveys no right to use or reproduce any trademark and/or tradename. Mention of those trademarks in the present document does not constitute an endorsement by ETSI of products, services or organizations associated with those trademarks.

Foreword

This ETSI Guide (EG) has been produced by ETSI Technical Committee Methods for Testing and Specification (MTS).

Modal verbs terminology

In the present document "**should**", "**should not**", "**may**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the ETSI Drafting Rules (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

Executive Summary

The present document offers a report of standardization activities for telecommunication interfaces and application programming interfaces based on the REpresentational State Transfer paradigm (RESTful APIs).

The guide collects conventions, methodology and design patterns from ETSI groups and from the industry and proposes consolidated guidelines to serve the complete lifecycle of standardization, from design to validation.

Introduction

More and more telecommunication and digital interfaces are being implemented as software-based solutions.
A well-known and largely adopted design methodology is taking place across several standardization activities: using the REpresentational State Transfer (REST) paradigm and resource-oriented protocols (e.g. HTTP(S), CoAP) or other possibly applicable protocols (MQTT, AMQP).

This phenomenon is becoming common practice in ETSI Technical Bodies (TBs) and Industry Specification Groups (ISGs) as well as in ETSI's Partnership Projects standardization activities, across several technologies, often quite different in scope and user community.

As adoption of standardizing RESTful APIs rises, it is becoming clear that specification of "RESTful APIs" needs to be:

- **Fast**, as the interfaces are simpler than other approaches and tend to have a shorter lifespan.

- **Automatable**, given the high number of conventions in the design of an API, parts of the specification, implementation and testing process are well suited to be automated.

- **Developer friendly**, since developers need support in the discovery and implementation of the interfaces by using tools and methodologies more closely aligned with software development.

In this regard, the present Guide for RESTful API specification and testing intends to support:

- **Consolidation of efforts** among different standardization groups and activities, who would be able to leverage from others' experience.

- **Delivery time** of specifications to be spent on the design of the application level features, more than re-assessing the principles and details at a transport protocol level.

- **Standards quality** to meet the excellence expected in the whole lifecycle of standardization, such as design, specification, testability and interoperability validation.

Several TBs and ISGs have already specified RESTful APIs and documented their conventions and processes in group specific guidelines. Further initiatives will be carried out during the upcoming years by the same groups as well as new ones, therefore it is strategic to align and consolidate the standardization efforts among ETSI membership.

The present document is structured as follows:

- clause 4 introduces the main concepts and terminology for the RESTful approach, then presents recommendations for RESTful API specifications development, with the introduction of a *code-first* approach and discussion of the foreseen benefits of its application;

- clause 5 presents recommendation and methodology for development of test specifications for RESTful APIs;

- clause 6 collects best practices and references to the available tools to manipulate and present the code needed artefacts;

- clause 7 contains a collection of examples on the expected outcomes of the different parts of the presented methodology;

- clause 8 reports on the outcomes from the analysis of the base documents - from ETSI groups or from other organizations - for the preparation of the present work and the results of a survey conducted among ETSI delegates on REST APIs adoption.

1 Scope

The scope of the present document is to present a methodology for specification and testing of RESTful APIs, i.e. telecommunication interfaces based on the Representational State Transfer paradigm, suitable for application in the standardization context.

In particular, the present guide is meant to serve ETSI membership and groups in the effort to unify and consolidate the approaches and practices in current and future standardization activities at ETSI and its Partnership Projects.

The Guide collects the best practices from standardization, industry and research in order to provide a modern and future-proofed approach to the subject.

The intended audience is primarily standardization groups at ETSI, but the guide may also serve as reference for users and vendors in industry, with a special focus in Open Source communities.

The Guide recommendations on conventions, methodologies, design-patterns and architectural choices to be used in standardization of RESTful APIs, specification and execution of standardized conformance and interoperability test suites.

2 References

2.1 Normative references

Normative references are not applicable in the present document.

2.2 Informative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long-term validity.

The following referenced documents are not necessary for the application of the present document, but they assist the user with regard to a particular subject area.

[i.1] "A Guide to Writing World Class Standards".

NOTE: Available at https://portal.etsi.org/Portals/0/TBpages/edithelp/Docs/AGuideToWritingWorldClassStandards.pdf?ver=2014-05-19-124137-453.

[i.2] Recommendation ITU-T I.130: "Method for the characterization of telecommunication services supported by an ISDN and network capabilities of an ISDN".

[i.3] IETF RFC 5023: "The Atom Publishing Protocol".

[i.4] ETSI EG 203 130: "Methods for Testing and Specification (MTS); Model-Based Testing (MBT); Methodology for standardized test specification development".

[i.5] IETF RFC 7231: "Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content".

[i.6] IETF RFC 8259: "The JavaScript Object Notation (JSON) Data Interchange Format".

[i.7] "The State of API 2019 survey", SmartBear.

[i.8] ETSI EG 201 015: "Methods for Testing and Specification (MTS); Standards engineering process; A Handbook of validation methods".

[i.9] ETSI EG 201 058 (V1.2.4): "Methods for Testing and Specification (MTS); Implementation Conformance Statement (ICS); proforma style guide".

[i.10] Repository of attachments for ETSI EG 203 647 on ETSI Forge.

NOTE: Available at https://forge.etsi.org/rep/mts/eg-203647-restful-api-guide.

[i.11] ETSI Forge Platform.

NOTE: Available at https://forge.etsi.org.

[i.12] ETSI Labs Platform.

NOTE: Available at https://labs.etsi.org.

[i.13] TDL Open Source Project.

NOTE: Available at https://top.etsi.org.

[i.14] "YAML Ain't Markup Language (YAML™) Version 1.2" specification.

NOTE: Available at https://yaml.org/spec/1.2/spec.html.

[i.15] OpenAPI™ specification, Version 3.0.3.

NOTE: Available at https://github.com/OAI/OpenAPI-Specification/blob/master/versions/3.0.3.md.

[i.16] JSON Schema definition.

NOTE: Available at https://json-schema.org/.

[i.17] IETF RFC 3986: "Uniform Resource Identifier (URI): Generic Syntax".

NOTE: Available at https://tools.ietf.org/html/rfc3986.

[i.18] IETF RFC 2388: "Returning Values from Forms: multipart/form-data".

NOTE: Available at https://tools.ietf.org/html/rfc2388.

[i.19] IETF RFC 1738: "Uniform Resource Locators (URL)".

NOTE: Available at https://tools.ietf.org/html/rfc1738.

[i.20] ETSI Drafting Rules.

NOTE: Available at https://portal.etsi.org/Services/editHelp/How-to-start/ETSI-Drafting-Rules.

[i.21] ETSI TS 129 501 (V15.3.0): "5G; 5G System; Principles and Guidelines for Services Definition; Stage 3 (3GPP TS 29.501 version 15.3.0 Release 15)".

[i.22] ETSI GS NFV-SOL 015 (V1.1.1): "Network Functions Virtualisation (NFV); Protocols and Data Models; Specification of Patterns and Conventions for RESTful NFV-MANO APIs".

[i.23] ETSI GS MEC 009 (V2.1.1): "Multi-access Edge Computing (MEC); General principles for MEC Service APIs".

[i.24] ETSI GR MEC-DEC 025 (V2.1.1): "Multi-access Edge Computing (MEC); MEC Testing Framework".

[i.25] ETSI GS NFV-TST 002: "Network Functions Virtualisation (NFV); Testing Methodology; Report on NFV Interoperability Testing Methodology".

[i.26] ETSI EG 202 568: "Methods for Testing and Specification (MTS); Internet Protocol Testing (IPT); Testing: Methodology and Framework".

[i.27] ETSI ES 203 119-4: "Methods for Testing and Specification (MTS); The Test Description Language (TDL); Part 4: Structured Test Objective Specification (Extension)".

[i.28] ETSI ES 203 119-1: "Methods for Testing and Specification (MTS); The Test Description Language (TDL); Part 1: Abstract Syntax and Associated Semantics".

[i.29] ETSI ES 201 873-1: "Methods for Testing and Specification (MTS); The Testing and Test Control Notation version 3; Part 1: TTCN-3 Core Language".

[i.30] ETSI ES 203 119-2: "Methods for Testing and Specification (MTS); The Test Description Language (TDL); Part 2: Graphical Syntax".

[i.31] ETSI ES 203 119-6: "Methods for Testing and Specification (MTS); The Test Description Language (TDL); Part 6: Mapping to TTCN-3".

[i.32] ETSI ES 201 873-11: "Methods for Testing and Specification (MTS); The Testing and Test Control Notation version 3; Part 11: Using JSON with TTCN-3".

[i.33] ETSI GR NFV-TST 007: "Network Functions Virtualisation (NFV) Release 2; Testing; Guidelines on Interoperability Testing for MANO".

[i.34] ETSI TS 118 115 (V2.0.0): "oneM2M; Testing Framework (oneM2M TS-0015 version 2.0.0 Release 2)".

[i.35] oneM2M TS-0018 (V3.2.0): "Test Suite Structure & Test Purposes".

[i.36] OpenAPI™ Tools website.

NOTE: Available at https://openapi.tools/.

[i.37] 3GPP validation tools repository.

NOTE: Available at https://forge.3gpp.org/rep/tools/3gpp-scripts.

[i.38] MEC API validation script example.

NOTE: Available at https://forge.etsi.org/rep/mec/gs011-app-enablement-api/blob/v2.1.1/.jenkins.sh.

[i.39] NFV API validation script example.

NOTE: Available at https://forge.etsi.org/rep/nfv/SOL002-SOL003/blob/v2.7.1/.jenkins.sh.

[i.40] RapiPDF project.

NOTE: Available at https://mrin9.github.io/RapiPdf/.

[i.41] REST API Design Rulebook by Mark Massé.

[i.42] ETSI GS NFV-SOL 013 (V2.7.1): "Network Functions Virtualisation (NFV) Release 2; Protocols and Data Models; Specification of common aspects for RESTful NFV MANO APIs".

[i.43] IETF RFC 7807: "Problem Details for HTTP APIs".

NOTE: Available at https://tools.ietf.org/html/rfc7807.

[i.44] IETF RFC 5246: "The Transport Layer Security (TLS) Protocol Version 1.2".

NOTE: Available at https://tools.ietf.org/html/rfc5246.

[i.45] IETF RFC 7519: "JSON Web Token (JWT)".

NOTE: Available at https://tools.ietf.org/html/rfc7519.

[i.46] ETSI TErms and Definitions Database Interactive (TEDDI).

NOTE: Available at https://webapp.etsi.org/Teddi/.

[i.47] IETF RFC 7396: "JSON Merge Patch".

[i.48] IETF RFC 6902: "JavaScript Object Notation (JSON) Patch".

[i.49] IETF RFC 5261: "An Extensible Markup Language (XML) Patch Operations Framework Utilizing XML Path Language (XPath) Selectors".

[i.50] IETF RFC 8288: "Web Linking".

[i.51] IETF RFC 7232: "Hypertext Transfer Protocol (HTTP/1.1): Conditional Requests".

[i.52] Git website.

NOTE: Available at https://git-scm.com.

[i.53] ETSI GS MEC-DEC 032-1: "Multi-access Edge Computing (MEC); API Conformance Test Specification; Part 1: Test Requirements and Implementation Conformance Statement (ICS)".

[i.54] ETSI GS CIM 009 (V1.2.1): "Context Information Management (CIM); NGSI-LD API".

[i.55] ETSI GS QKD 014 (V1.1.1): "Quantum Key Distribution (QKD); Protocol and data format of REST-based key delivery API".

[i.56] ETSI GS NFV-TST 010 (V2.4.1): "Network Function Virtualisation (NFV) Release 2; Testing; API Conformance Testing Specification".

[i.57] TMF630 API Design Guidelines 4.0.1.

NOTE: Available at https://www.tmforum.org/resources/how-to-guide/tmf630-api-design-guidelines-4-0/.

[i.58] ETSI GS NFV-SOL 002: "Network Functions Virtualisation (NFV) Release 2; Protocols and Data Models; RESTful protocols specification for the Ve-Vnfm Reference Point".

[i.59] ETSI GS NFV-SOL 003: "Network Functions Virtualisation (NFV) Release 2; Protocols and Data Models; RESTful protocols specification for the Or-Vnfm Reference Point".

[i.60] ETSI GS NFV-SOL 005: "Network Functions Virtualisation (NFV) Release 2; Protocols and Data Models; RESTful protocols specification for the Os-Ma-nfvo Reference Point".

3 Definition of terms, symbols and abbreviations

3.1 Terms

For the purposes of the present document, the following terms apply:

Application Programming Interface (API): interface implemented by a software program to be able to interact with other software programs

ATOM: Atom Publishing Protocol

NOTE: For more information see IETF RFC 5023 [i.3].

collections: set of resources

Hypertext Transfer Protocol (HTTP): application level protocol, on layer 7 of the ISO/OSI model

OpenAPI™ Specification (OAS™): standard, language-agnostic interface to RESTful APIs which allows both humans and computers to discover and understand the capabilities of the service without access to source code, documentation, or through network traffic inspection

representation: concrete entity, which encodes a resource in e.g. HTML, JSON or XML

NOTE: A resource may be available in multiple representation, such as a JSON message and as an XML message.

resource: object with a type, associated data, a set of methods that operate on it, and, if applicable, relationships to other resources

NOTE: A resource is a fundamental concept in a RESTful API. Resources are acted upon by the RESTful API using the Methods (e.g. POST, GET, PUT, DELETE, etc.). Operations on Resources affect the state of the corresponding managed entities.

SDO: Standards Developing or Standards Setting Organization

Uniform Resource Identifier (URI): address of the resource and identification of the resource

3.2 Symbols

Void.

3.3 Abbreviations

For the purposes of the present document, the following abbreviations apply:

AMQP	Advanced Message Queuing Protocol
API	Application Programming Interface
ATS	Abstract Test Suite
BWC	BackWard Compatible
CCI	Co-Channel Interference
CIM	Cross-cutting Information Management
CR	Change Request
CRUD	Create, Read, Update, Delete
CTC	Change Type Code
CTK	Conformance ToolKit
EDM	Entity Data Model
EHR	Electronic Health Record
EM	Element Manager
ETag	Entity Tag
ETSI	European Telecommunications Standards Institute
GS	Group Specification
HATEOAS	Hypermedia As The Engine Of Application State
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
HTTPS	HTTP Secure
ICS	Implementation Conformance Statement
IETF	Internet Engineering Task Force
IFA	InterFaces and Architecture
IFS	Interoperable Function Statement

NOTE: The acronym IFS may also refer to Interoperable Feature Statement, Implementable Functions Statement and other similar terminology, all referring to the identification of a communication behaviour which has relevance for successful interoperability among communicating entities. The list of usages of IFS in different ETSI specification may be retrieved using the TEDDI tool at the ETSI Portal [i.46].

ISG	Industry Specification Group
IT	Information Technology
ITU-T	International Telecommunication Union - Telecommunication standardization sector
IUT	Implementation Under Test
JSON	JavaScript Object Notation
JWE	JSON Web Encryption
JWS	JSON Web Signature
JWT	JSON Web Token
MANO	Manager and Orchestrator
MEC	Multi-access Edge Computing
MQTT	Message Queuing Telemetry Transport
MSC	Message Sequence Chart

MTS	Methods for Testing and Specification
NBWC	Non-BackWard Compatible
NFV	Network Functions Virtualisation
OAS	OpenAPI™ Specification
OASIS	Organization for the Advancement of Structured Information Standards
OMG	Object Management Group
PICS	Protocol Implementation Conformance Statement
QKD	Quantum Key Distribution
RAML	RESTful API Modelling Language
REST	REpresentational State Transfer
RFC	Request For Comments
RPC	Remote Procedure Call
SBI	South Bound Interface
SDO	Standard Development Organization
SOL	SOLutions
STF	Specialist Task Force
SUT	System Under Test
TC	Technical Committee
TCP	Transmission Control Protocol
TCP/IP	Transmission Control Protocol over the Internet Protocol
TD	Test Description
TDL	Test Description Language
TDL-TO	Test Description Language - Test Objective extension
TDL-GR	Test Description Language - Graphical notation
TLS	Transport Layer Security
TM	Tele Management
TOP	TDL Open Source Project
TP	Test Purpose
TSS	Test Suite Structure
TTCN	Test and Test Control Notation
UML	Unified Modelling Language
URI	Uniform Resource Identifier
URL	Unified Resource Locator
VCS	Version Control System
WADL	Web Application Description Language
XML	eXtensible Markup Language
YAML	YAML Ain't Markup Language
ZSM	Zero-touch Service Management

4 Specification methodologies for RESTful APIs

4.1 RESTful APIs specification in a staged standardization approach

Standardization best practices for telecommunications recommend that a staged approach is taken in the definition of communications systems. The methodology recommended in "A Guide to Writing World Class Standards" [i.1], page 31, builds upon the Recommendation ITU-T I.130 [i.2] and indicates three stages to design telecommunication standards:

- Stage 1: Specify objectives from the user perspective.
- Stage 2: Develop a functional model to meet those objectives.
- Stage 3: Develop a specification of the detailed implementation requirements.

These three steps may be refined as follows:

- Stage 1: Specification of high-level user requirements on the technology, i.e. the expectations for the communications system to meet.

- Stage 2: Develop architectures, identify atomic components and reference points interconnecting them and within the reference points identify the interfaces and information models required.
- Stage 3: Specify implementation level requirements for the interfaces identified, i.e. the protocols, data models and serialization techniques expected to be seen at a "wire" level.

The specification of RESTful APIs plays its role at stage 3: given an interface between two components in the system architecture, a RESTful API provides the implementation details for the communication between these two entities, which are then identified as the API Producer (or Server) and the API Consumer (or Client).

In this respect, the RESTful approach lets the designer of the standard focus on the entities and data exchanged or manipulated, while providing a framework where (typically) the underlying protocols and serialization techniques are already specified elsewhere.

As an example, the HTTP protocol over TCP/IP and XML serialization may be selected (among many others). Once these choices are made, the designer of the API needs only to focus on the entities manipulated by the interface.

A fourth stage in this process may be identified: the development of testing specifications, for interoperability or for conformance. As for any communication technology, sound test specifications are required to validate the standards and to certify the implementations. Given the specific characteristics of RESTful APIs, the generic ETSI test development methodology will be tailored and documented in subsequent clauses of the present document.

4.2 Introduction on RESTful interfaces

4.2.1 Introduction

In computer communications, the term REST, coined by Roy Fielding in the year 2000, indicates the Representational State Transfer architectural style, defining a set of constraints and agreements based on the concept of Resource Representations. The REST approach does not enforce rules for implementations at a lower level, rather, it draws high-level design guidelines for interactions among different communicating entities.

An Application Programming Interface (API) is a set of libraries or specifications which allows interaction with an external artefact or agent from a third party. APIs which comply with the REST constraints are said to be *RESTful* and refer to the description of a communication interface that allows interacting with a system based on the REST architecture style.

Before introducing a methodology for specification of RESTful APIs in standardization, in the next clause the main principles of REST are presented, to introduce the required terminology and set the background.

4.2.2 Main Principles of the RESTful paradigm

It is recommended for the implementation of APIs to be technology or protocol independent. RESTful APIs take all aspects of HTTP/1.1 [i.5] including its request methods, response codes, and HTTP headers. A RESTful API specification comprises of the following information:

- Purpose of the API;
- URIs of resources;
- HTTP methods for a given resource [i.5];
- Supported representations (e.g. JSON, see [i.6]);
- Request body schema(s) (where applicable);
- Response body schema(s) (where applicable);
- HTTP response status codes.

To abide by certain principles, the use of OpenAPI™ specifications is recommended to design the APIs first.