TDH022 – TECHNICAL INTEROPERABILITY GUIDELINES AND API MANAGEMENT

Operative Document

Implementation Recommendations

Versione: 0.2

Data: 22/02/2022

Version	Release Date	Release Type
0.1	15/12/2021	First Release – Italian
0.2	22/02/2022	Second Release – English

Contents Index

CHAPTER 1 – INTRODUCTION

This Operative Document contains guidelines that providers *(in this sense, they are considered Public Institutions such as, for instance, Regions and Provinces, as well as Public Entities or similar and Private Entities, including Second and Third Parties that make services and functions available to the TDH)* certified within the Tourism Digital Hub must be considered in the implementation of APIs in order to promote interoperability with users, who are also certified within the Tourism Digital Hub *(in this sense we consider all subjects that use the digital services made available by providers within the ecosystem).*

The recommendations are applied by providers according to specific application needs and/or in relation to the nature of the users.

This document, whose application is related to the specific context of the Tourism Digital Hub (TDH), was drafted on the basis of the Operative Document "Implementation Recommendations"[1] issued by AgID and linked to the document "Guidelines on Technical Interoperability of Public Administrations"[2] also issued by AgID; in this sense, please refer to the two documents mentioned above for the general principles.

---

[1] Online reference: https://www.agid.gov.it/sites/default/files/repository_files/04_raccomandazioni_di_implementazione.pdf
[2] Online reference: https://www.agid.gov.it/sites/default/files/repository_files/linee_guida_interoperabilit_tecnica_pa.pdf

CHAPTER 2 – APPLICATION SCOPE

This Operative Document is intended as an Operative Document related to the Technical Interoperability Guidelines.

2.1 Recipients of this document

This Operative Document is intended for all providers who provide users with services and functions within the Tourism Digital Hub (TDH), as well as for the users themselves, in order to enable the fruition of the desired services and functions; therefore, these provisions can be used as a basis for the implementation of new functions in case they have to be developed from scratch or as a basis for the integration of existing functions.

The following is a list of Public and Private Parties to whom the Operational Document is addressed, both as providers and users of services and functions within the Tourism Digital Hub (TDH).

Public Parties

- Central Public Administration (e.g., Ministry of Tourism),
- Local Public Administration (e.g., Regions, Provinces...),
- National and Local Authorities (e.g., ENIT),
- Non-Profit Organizations,
- Public Enterprises related to tourism (e.g., ski lifts...).

Private Parties

- Hospitality enterprises, catering enterprises, etc.,
- Tour Operators/Travel Agencies,
- Unions,
- Private Enterprises related to tourism (e.g., ski lifts…).

CHAPTER 3 – REFERENCES AND ABBREVIATIONS

3.1 Document Reading Notes

In accordance with ISO/IEC Directives, Part 3 for drafting technical documents this Operational Document will use the keywords "MUST", "MUST NOT", "SHOULD", "SHALL NOT", "MAY" and "OPTIONAL", the interpretation of which is described below:

- **MUST**, specify a mandatory requirement to comply with Guidelines;

- **MUST NOT**, indicate an absolute no-go on specifications;

- **SHOULD or SHOULD NOT**, mean that the implications must be understood and carefully weighed before choosing alternative approaches;

- **MAY or OPTIONAL**, signifies that the reader may choose to apply or not apply the specification without any kind of implication or restriction.

3.2 Terms and Definitions[3]

For an easier reading, a glossary of terms and definitions contained in this document is given below.

[AgID]	Digital Agency for Italy
[API]	Application Programming Interface
[Array]	Complex, static and heterogeneous data structure
[CAD]	Legislative Decree 7 March 2005, n. 82 - "Digital Administration Code" (also known as "CAD"), updated with amendments by Legislative Decree 76 of 16 July 2020 and converted into law with Law 120 of 11 September 2020
[E2E-Key]	Unique transaction key

---

[3] Some terms and definitions explained in this paragraph are also available in the Guidelines on Technical Interoperability for Public Administrations issued by AgID (see the section "Reference Bibliography and Sitography" for the redirect links to the cited contents).

[Provider]	One of the subjects referred in Article 2, paragraph 2 of the CAD that makes e-services available to other organizations, for the use of data in its possession or the integration of the processes it has carried out
[User]	Organization that uses the e-services made available by one of the subjects referred in Article 2, paragraph 2 of the CAD
[HTTP]	Hypertext Transfer Protocol
[IDL]	Interface Description Language
[JSON]	JavaScript Object Notation
[Open API]	Specification for managing RESTful web services
[REST]	Representational State Transfer
[Req-Timestamp]	Time stamp for a request
[SOAP]	Simple Object Access Protocol
[TDH]	Tourism Digital Hub
[TDH022]	TDH022 - Interoperability interface of the Tourism Digital Hub
[UTF-8]	Unicode Transformation Format, 8 bit
[WSDL]	Web Services Description Language

CHAPTER 4 – TECHNICAL IMPLEMENTATION RECOMMENDATIONS

The content of this chapter contains technical recommendations that providers should keep in mind for the purposes of building APIs to be displayed within the Tourism Digital Hub (TDH).

[RAC_GEN_001] APIs' Description

APIs MUST be represented using a standard Interface Description Language (IDL). Specifically:

- **REST:** Swagger 2.0, RAML 1.0, OpenAPI 3.0 and following;
- **SOAP:** WSDL 1.1 and following.

[RAC_GEN_002] APIs' Endpoint

The version number MUST NOT be listed within the API name.

You SHOULD indicate the version number and technology in the API endpoint.

Example:

```
http://<dominioOrganizzativo>-<NomeAPI>-
[rest|soap]/<DominioApplicativo>/v<major>[.<minor>[.<patch>]]/<risorsa>
```

dove:

- **<dominioOrganizzativo>** refers to the organization providing the service;
- **[rest|soap]** represents the API technology;
- **<DominioApplicativo>** means the area within the organization;
- **v<major>[.<minor>[.<patch>]]** shows the version number consistent with Semantic Versioning 2.0.01;
- **<NomeAPI>** is the name of the API specification;
- **<risorsa>** it shows the logical path, also composed, to access the resource to which it is referred.

[RAC_GEN_003] Default encoding

You SHOULD use UTF-8 as the default encoding for your data.

[RAC_GEN_005]

In the Soap Header or in the http Header (in case of REST call) the following minimum data must be sent:

- **Source**: uniquely indicates the source of the message. This field is a string type, and will be composed as follows: <ragionesociale>_<progressivo> where:
    - <ragionesociale> indicates the business name of the calling party,
    - <progressivo> unique id shared between the calling party and the TDH022.
- **Req-Timestamp:** indicates the time when the request was made. This field is of date type according to [RAC_GEN_FORMAT_003];
- **e2e-Key**: represents the unique key of the transaction. The field is of string type and will have a maximum length of 20 characters.

[RAC_REST_FORMAT_003] Representational Conventions

The following representation conventions MUST be used:

- Booleans MUST not be null;
- Empty arrays MUST not be null, but empty lists, e.g. [];
- Enumerations MUST be represented by non-null strings;
- "Data" types MUST not be null or empty.

[RAC_REST_NAME_007] Using the Problem JSON schema for error responses

In the case of errors handled by the application they MUST be returned:

- A payload of type Problem defined in RFC 7807;
- the media type application/problem+json;
- an explanatory status code;
- the object, possibly extended.

When returning an error, it is important not to expose internal application data. To prevent the risk of user-enumeration, the authentication error messages must not provide information about the existence or not of the user.

After validating the content of requests, you MUST return an error http code according to RFC 2616.

CHAPTER 5 – ROBUSTNESS[4]

In order to guarantee the responsiveness of an API, it is necessary to prevent individual users from exhausting the computational and bandwidth capacity of the provider. The technique commonly used in these cases is rate limiting (also known as throttling). The rate limit provides a specific user with a maximum number of requests that can be satisfied within a specific time frame (e.g., 1000 requests per minute). A number of requests that exceeds the imposed limit causes the refusal of further requests by a specific user for a predetermined time interval.

The policies regarding the maximum number of requests and the relative time window, and those regarding the waiting time for new requests (which can be increased in the case of repeated requests, e.g., with a policy of exponential increase) are left to the implementers after an analysis of the maximum load that the provider can bear.

[RAC_ROBUSTNESS_001] Report usage limits

REST service interface providers MUST report any limits reached with HTTP status 429 Too Many Requests.

APIs return global throttling values in each response via the following headers:

- **X-RateLimit-Limit:** maximum limit of requests for an endpoint;
- **X-RateLimit-Remaining:** number of requests remaining until the next reset;
- **X-RateLimit-Reset:** number of seconds remaining until the next reset.

If the quotas are exceeded, the API SHOULD also return the header:

- HTTP header Retry-After: minimum number of seconds after which the client is invited to retry.

In the case of SOAP, there are no standard guiding rules for handling rate limiting and throttling. You MAY use the same HTTP headers and status code as in the REST case.

Users MUST:

---

[4] The content of the introduction to this paragraph is in line with the requirements of the Operative Document "Implementation Recommendations" linked to the "Guidelines on Technical Interoperability of Public Administrations" published by AgID, referred to in Chapter 8 of the Operative Document cited above (see the section "Reference Bibliography and Sitography" for links to redirect content).

- respect the throttling headers;

- respect the X-RateLimit-Reset header when it returns the number of seconds to the next reset, and eventually manage the unix timestamp indication;

- respect the HTTP header Retry-After both in the variant that exposes the number of seconds after which to retry, and in the variant that exposes the date on which to retry.

[RAC_ROBUSTNESS_002] Report system overload or service unavailability

Providers MUST define and display a business continuity plan by reporting system overload or service unavailability with HTTP status 503 Service Unavailable.

In the event of an overload or unavailability, the dispenser SHOULD also return:

- HTTP header Retry-After with the minimum number of seconds after which the client is invited to retry.

Users MUST:

- comply with HTTP header Retry-After in both the variant that exposes the number of seconds after which to retry, and the variant that exposes the date on which to retry.

Rate limit management should be external to the interface descriptor, through an API Management component. If the subject should lack such a component, it MUST provide in the API description the indication of the headers related to the rate limiting. The use of HTTP headers in SOAP is outside the scope of WSDL as Interface Definition Language.

REFERENCE BIBLIOGRAPHY AND SITOGRAPHY

Guidelines on Technical Interoperability of Public Administrations

Author: AgID – First Release: 27/04/2021

Online reference: https://www.agid.gov.it/sites/default/files/repository_files/linee_guida_interoperabilit_tecnica_pa.pdf

Operative Document - Implementation Recommendations

Author: AgID – First Release: 27/04/2021

Online reference: https://www.agid.gov.it/sites/default/files/repository_files/04_raccomandazioni_di_implementazione.pdf