Swagger 3.0.3

Definitions

OpenAPI Document

A document (or set of documents) that defines or describes an API. An OpenAPI definition uses and conforms to the OpenAPI Specification.

Path Templating

Path templating refers to the usage of template expressions, delimited by curly braces ({}), to mark a section of a URL path as replaceable using path parameters. Each template expression in the path MUST correspond to a path parameter that is included in the Path Item itself and/or in each of the Path Item's Operations.

Media Types

Media type definitions are spread across several resources. The media type definitions SHOULD be in compliance with RFC6838.

HTTP Status Codes

The HTTP Status Codes are used to indicate the status of the executed operation. The available status codes are defined by RFC7231 and registered status codes are listed in the IANA Status Code Registry.

Specifications

The OpenAPI Specification is versioned using Semantic Versioning 2.0.0 (semver) and follows the semver specification.

Format

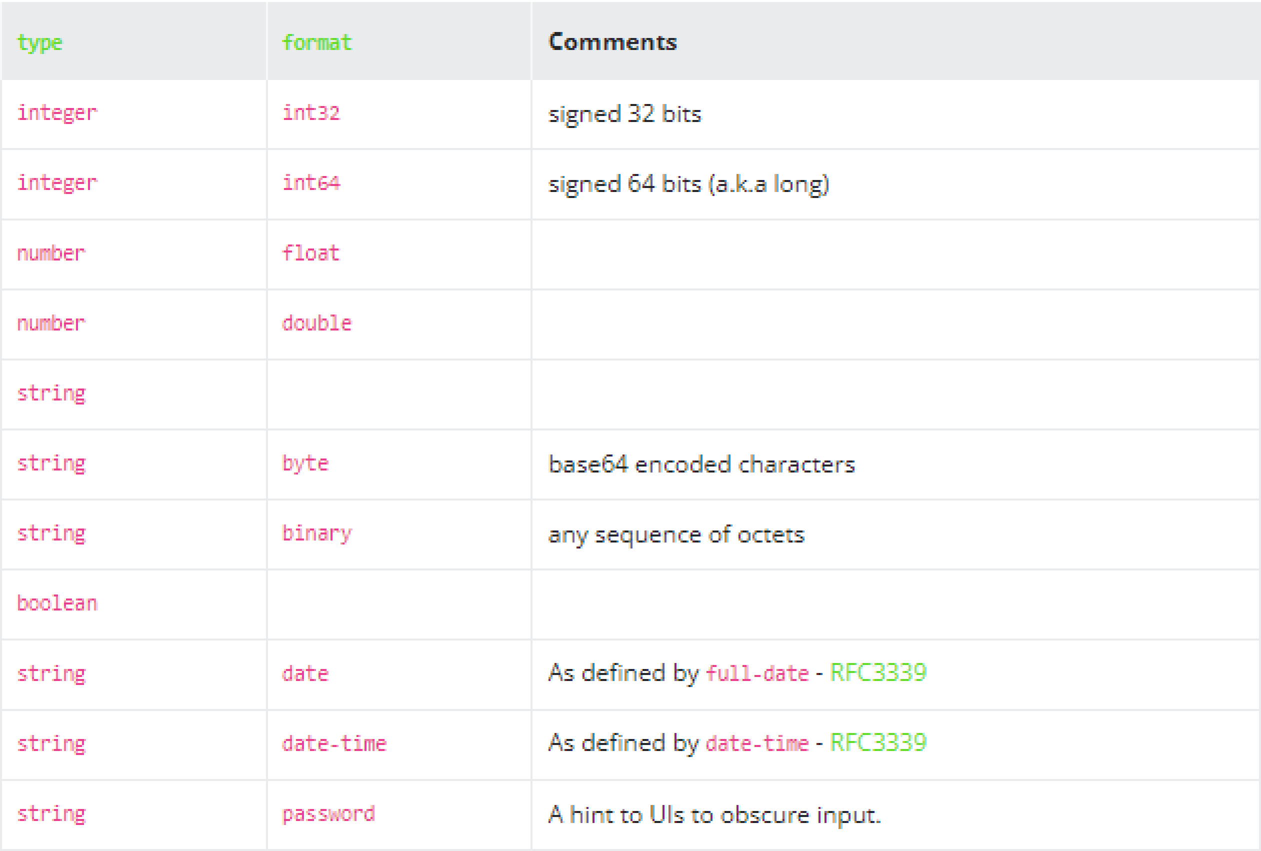
An OpenAPI document that conforms to the OpenAPI Specification is itself a JSON object, which may be represented either in JSON or YAML format. While APIs may be defined by OpenAPI documents in either YAML or JSON format, the API request and response bodies and other content are not required to be JSON or YAML.

Document Structure

An OpenAPI document MAY be made up of a single document or be divided into multiple, connected parts at the discretion of the user. In the latter case, $ref fields MUST be used in the specification to reference those parts as follows from the JSON Schema definitions.

Data Types

The formats defined by the OAS are:



type	format	Comments
integer	int32	signed 32 bits
integer	int64	signed 64 bits (a.k.a long)
number	float	
number	double	
string		
string	byte	base64 encoded characters
string	binary	any sequence of octets
boolean		
string	date	As defined by full-date - RFC3339
string	date-time	As defined by date-time - RFC3339
string	password	A hint to UIs to obscure input.

Relative References in URLs

Unless specified otherwise, all properties that are URLs MAY be relative references as defined by RFC3986. Relative references are resolved using the URLs defined in the Server Object as a Base URI.

Relative references used in $ref are processed as per JSON Reference, using the URL of the current document as the base URI. See also the Reference Object.

Specification Extensions

While the OpenAPI Specification tries to accommodate most use cases, additional data can be added to extend the specification at certain points.

The extensions properties are implemented as patterned fields that are always prefixed by "x-".

Security Filtering

Some objects in the OpenAPI Specification MAY be declared and remain empty, or be completely removed, even though they are inherently the core of the API documentation.

The reasoning is to allow an additional layer of access control over the documentation. While not part of the specification itself, certain libraries MAY choose to allow access to parts of the documentation based on some form of authentication/authorization.

Two examples of this:

- The Paths Object MAY be empty. It may be counterintuitive, but this may tell the viewer that they got to the right place, but can't access any documentation. They'd still have access to the Info Object which may contain additional information regarding authentication.
- The Path Item Object MAY be empty. In this case, the viewer will be aware that the path exists, but will not be able to see any of its operations or parameters. This is different from hiding the path itself from the Paths Object, because the user will be aware of its existence. This allows the documentation provider to finely control what the viewer can see.