

FICO® Marketplace: OpenAPI (OAS) Requirements

Overview

To streamline integrations for FICO Platform clients, each API listed on FICO Marketplace is accompanied by a corresponding FICO Platform Connector and related artifacts. These connectors are generated from the API specifications that you give us for each of your APIs. These connectors ensure seamless setup, connectivity, and maintenance for FICO Platform clients calling your APIs.

FICO has standardized on the **OpenAPI Specification** as the preferred format for you to describe your JSON-based RESTful APIs. This standard simplifies human and machine readability, facilitating interaction without requiring access to source code or supplementary API documentation.

This document outlines the areas in your OAS that FICO will be reviewing and validating during marketplace onboarding. This validation is to ensure that FICO clients can easily interact with your API from the FICO Platform.

If you have any questions or require assistance around your API and using the OAS standard, please contact us at marketplace-ops@fico.com.

General information

Your specifications must conform to **OAS 3.0 or higher** and be submitted to FICO in JSON format.

Field Name	Req*	Description	Example
OpenAPI	M	Specifies the Open API version used.	"3.0.3"
info	M		
title	M	Title of the API.	"API Title"
version	M	Version of the API.	"1.0.0"
.description	S	Description of the API.	"API for managing users"

M = Mandatory S = Suggested O = Optional

Servers

Your specifications must include at least one nonproduction server. Ideally, providing both a nonproduction and production server would be optimal.

Field Name	Req*	Description	Example
servers []	M	Required at least 1 entry, usually for development environment.	
.URL	M	URL of the API server.	"https://api.example.com"
.description	S	Description of the server.	"Sandbox environment"

M = Mandatory S = Suggested O = Optional

Tags

Tags are optional in your specifications, but if you're using them, please note the mandatory elements below.

Field Name	Req*	Description	Example
.tags []	O		
name	M	Is used for grouping operations.	"User Management"
.description	S	Description of the tag.	"API for managing users"

M = Mandatory S = Suggested O = Optional

Endpoints

Your specifications must include at least one endpoint. The endpoint must specify at least one media type (currently supported application/json) and at least one example 200 response for each HTTP method accepted for that endpoint. If you require a request, to invoke your API, you'll need to include the corresponding request(s) to the above 200 response(s).

Field Name	Req*	Description	Example
paths	M	Requires at least 1 path.	
{path}	M	Path of the API endpoint.	"/users/{userId}"
{method}	M	HTTP method for the operation.	"post"
.tags	O	Tags assigned to the operation.	["User Management", ..]
.description	S	Description of the endpoint.	"Creates a new user"
.operation	S	Unique identifier for the operation.	"createUser"

M = Mandatory S = Suggested O = Optional

Parameters

Parameters are optional, but if you do include them, please avoid duplicate parameter names within an operation. Also please note the mandatory elements below.

Field Name	Req*	Description	Example
parameters[]	O		
.name	M	Name of the parameter.	"userId"
.in	M	Location of the parameter (query, header, path, cookie).	"query", "header", "path"
.required	S	Indicates if the parameter is required.	true
.schema.type	M	Data type of the parameter.	"string", "integer", "boolean"
.style .explode	M	Mandatory for schema type array.	<pre> "parameters": [{ "name": "someArray", "in": "query", "style": "form", "explode": true, "schema": { "type": "array", "items": { "type": "string" } } }] </pre>

.default	O	Default value.	"defaultValue"
.example	S	Example value for the parameter.	"12345", "true", "exampleValue"
.description	S	Description of the parameter.	"The ID of the user"

M = Mandatory S = Suggested O = Optional

Request Body

If your API is invoked without a request body (through a query parameter or another method) than this is optional for you. But if your API requires a request to be invoked, please note the requirements below.

It's very important that your specifications include request examples so FICO® Platform clients can easily understand common scenarios for calling your API. Furthermore, it's critical that you include a schema that covers all examples provided per endpoint and HTTP method and that matches declared types and structural properties.

Field Name	Req [*]	Description	Example
requestBody	O		
.required	O	Indicates if the request body is required.	true
.content	M	Object defining content type and schema.	
.{mediaType}	M	Media type of the request body. Currently supported	"application/json"

		application/json	
.schema	M	Request schema definition.	<pre>{ "type": "object", "properties": { "name": { "type": "string" }, ... }, "required": ["name"] }</pre>
.schema.required	O	List of required fields in the schema.	"required": ["name"]
.schema.\$ref	O	Reference to the request schema definition within components section. Using inline defined or referenced request schema is mutually exclusive.	"#/components/schemas/CreateUserRequest"
.examples	M	Request examples. At least 1 example is required.	

M = Mandatory S = Suggested O = Optional

Response Body

Response body details are mandatory. It's very important that your specifications include examples so FICO® Platform clients can easily understand common scenarios for getting data back from your API. Furthermore, it's critical that you include a schema that covers all examples provided per endpoint and HTTP method and that matches declared types and structural properties.

Finally, please include at least one 400 error response, with detailed error codes and examples. Ideally, we'd like all your 400 and 500 error messages, as FICO Platform clients trigger actions based on error codes.

Field Name	Req *	Description	Example
responses	M		
{statusCode}	M	Response status code, 200, 400, ... default.	"200", "400", "default"
.description	O	Description of the response status code.	"OK"
.headers	O	Response headers.	
.content	O	Response content.	
.{mediaType}	M	Media type of the response body. Currently supported application/json	"application/json"
.schema	M	Response schema definition.	<pre>{ "type": "object", "properties": { "message": { "type": "string" } } }</pre>

.schema.\$ref	O	Reference to the response schema definition. Using inline defined or referenced response schema is mutually exclusive.	"#/components/schemas/UserResponse"
.schema.oneOf	O	Response payload matches exactly one of the specified subschemas; i.e., is of polymorphic type.	<pre>"schema": { "oneOf": [{ "\$ref": "#/components/schemas/MessageResponse" }, { "\$ref": "#/components/schemas/ErrorResponse" }] }</pre>
.examples	M	Response named examples. If content is present, at least 1 example must be provided.	

M = Mandatory S = Suggested O = Optional

Callbacks

If your API supports asynchronous operations, please include this in your specifications. For APIs that do not support asynchronous operations, this is not required.

Field Name	Req*	Description	Example
callbacks	O	Defines asynchronous nature of an API. This allows the provider to describe the callback operation that the server will invoke on the	

		client once the processing is complete.	
--	--	---	--

M = Mandatory S = Suggested O = Optional

Security Schemes

Including security scheme details in your specification is optional, but we strongly encourage it, as it helps clients understand what schemes are available for use.

Field Name	Req*	Description	Example
components .securitySchemes	S		
{schemeName}	M	Name of the security scheme.	"basicAuth"
.type	M	Type of security scheme.	"http", "apiKey", "oauth2"
.scheme	M	HTTP authentication scheme.	"basic", "bearer"
security	O	Security requirements for the API.	[{"basicAuth": []}]

M = Mandatory S = Suggested O = Optional

Components Schema

Components schemas are optional, but if you're using them, please note the mandatory elements below.

Field Name	Req*	Description	Example
components.schemas	O		
{schemaName}	M	Name of the schema.	"CreateUserRequest"
.type	M	Data type of the schema (object, string, etc.).	"object"
.properties	O	Schema properties. Mandatory and it should include at least one property entry if schema type is object.	
.{propertyName}.type	M	Data type of the schema property.	"string", "integer", "array"
.{propertyName}.description	O	Description of the schema property.	"The name of the user"
.default	O	Default value.	"defaultValue"
.example	O	Example value for the property.	"User name"
.required	O	List of required fields in the schema.	["userId", "name"]

M = Mandatory S = Suggested O = Optional