

REST API Schemas

Ivan Kezić

Version: 1.8

Last Modified: September 17, 2025

Operations: *Mark Paid, Reject, Report, Get Fiscalization Status, Search Classification, Check Identifier, XML Validation*

Note: This document represents a draft version and is subject to ongoing review and updates.

Document Revision History

Version	Date	Author	Description of Change
1.0	2025-06-20	Ivan Kezić	initial draft version created
1.1	2025-06-26	Ivan Kezić	updated values for <i>PaymentMethod</i> (section 5.1), <i>RejectionReasonType</i> (section 5.2), and <i>InvoiceType</i> (section 5.3)
1.2	2025-06-26	Ivan Kezić	added important note and updated request structure for <code>markPaidWithoutElectronicId</code> and <code>rejectWithoutElectronicId</code> (sections 2.1.1 and 2.2.1)
1.3	2025-07-10	Ivan Kezić	changed data type for <code>ElectronicId</code> from string to int
1.4	2025-07-28	Ivan Kezić	removed <code>oib</code> from <code>rejectWithoutElectronicId</code> request (section 2.2.1)
1.5	2025-09-02	Ivan Kezić	- added minor styling changes and cross references for better readability - added url for Validation System in section 7 - added <code>successTimestamp</code> and <code>xmlBytes</code> to all success responses in sections 2.1 and 2.2 - added error response in section 1 when username or password is incorrect - added API endpoint for checking if identifier is registered or not (section 3) - added API endpoint for getting fiscalization status (section 4)
1.6	2025-09-08	Ivan Kezić	- updated <code>IdentifierType</code> values from string to int (section 5.4) - updated <code>MessageType</code> values from string to int (section 5.5) - updated <code>StatusId</code> values from string to int
1.7	2025-09-16	Borna Žaja, Ivan Kezić	- updated error response structure (4xx / 5xx Not OK) in all requests - updated <code>Get Fiscalization Status</code> endpoint from GET to POST (section 4) - removed <code>StatusId</code> enumeration and references (section 4) - renamed <code>xmlBytes</code> field to <code>encodedXml</code> - renamed <code>actionTimestamp</code> to <code>fiscalizationTimestamp</code> (section 4) - added <code>isSuccess</code> field to <code>Get Fiscalization Status</code> response - added <code>isSuccess</code> and <code>fiscalizationTimestamp</code> fields to <code>Mark Paid</code>, <code>Reject</code>, and <code>Report</code> responses (sections 2.1, 2.2, and 2.3) - added <code>isSuccess</code> and <code>reportingTimestamp</code> fields to <code>Report</code> responses (section 2.3) - added <code>reportingTimestamp</code> field to <code>Report</code> response (section 2.3)
1.8	2025-09-17	Ivan Kezić	renamed <code>rejectionDescription</code> to <code>rejectionReasonDescription</code> in <code>rejectWithoutElectronicId</code> request (section 2.2.2)

Contents

1	Authorization	4
2	Fiscalization 2.0 Operations	4
2.1	Mark Paid Operations	4
2.1.1	Mark Paid without ElectronicId	4
2.1.2	Mark Paid with ElectronicId	5
2.2	Reject Operations	6
2.2.1	Reject without ElectronicId	6
2.2.2	Reject with ElectronicId	6
2.3	Report Operation	7
3	Check Identifier	8
4	Get Fiscalization Status	9
5	Common Enumerations	10
5.1	PaymentMethod	10
5.2	RejectionReasonType	10
5.3	InvoiceType	10
5.4	IdentifierType	10
5.5	MessageType	10
6	Search Classification	10
7	Validation System	12

1 Authorization

All requests require the following authentication fields:

Field	Description
username	Authentication username
password	Authentication password
companyId	Company identifier
softwareId	Software identifier

Error Response (401 Unauthorized)

When the provided username or password is incorrect, the API will return a 401 Unauthorized error with the following response:

```
1 {
2   "traceId": "b5feb4c7-5c58-49a9-84fe-891f4f76e659",
3   "message": "Some error message",
4 }
```

2 Fiscalization 2.0 Operations

2.1 Mark Paid Operations

2.1.1 Mark Paid without ElectronicId

Endpoint: POST /api/fiscalization/markPaidWithoutElectronicId

Important Note: The value of companyId must be the same as senderIdentifierValue.

Request:

```
1 {
2   "username": "user@mail.com",
3   "password": "mystrongpass",
4   "companyId": "1111111111",
5   "softwareId": "software-123",
6   "internalMark": "INV-2025-001",
7   "issueDate": "2025-06-09",
8   "senderIdentifierValue": "1111111111",
9   "recipientIdentifierValue": "2222222222",
10  "paymentDate": "2025-06-15",
11  "paymentAmount": 1250.75,
12  "paymentMethod": "T"
13 }
```

Note: For possible values of paymentMethod, see section 5.1.

Response:

Success (200 OK)

```
1 {
2   "isSuccess": true,
3   "fiscalizationTimestamp": "2025-06-15T14:30:25Z",
4   "encodedXml": "PD94bWwgdmVyc2lvbj0iMS4wIiB..."
5 }
```

Error (4xx / 5xx Not OK)

```
1 {
2   "traceId": "b5feb4c7-5c58-49a9-84fe-891f4f76e659",
3   "message": "Some error message",
4 }
```

2.1.2 Mark Paid with ElectronicId

Endpoint: POST /api/fiscalization/markPaid

Request:

```
1 {
2   "username": "user@mail.com",
3   "password": "mystrongpass",
4   "companyId": "123456",
5   "softwareId": "software-123",
6   "electronicId": 121111131,
7   "paymentDate": "2025-06-15",
8   "paymentAmount": 1250.75,
9   "paymentMethod": "T"
10 }
```

Note: For possible values of paymentMethod, see section 5.1.

Response:

Success (200 OK)

```
1 {
2   "isSuccess": true,
3   "fiscalizationTimestamp": "2025-06-15T14:30:25Z",
4   "encodedXml": "PD94bWwgdmVyc2lvdj0iMS4wIiB..."
5 }
```

Error (4xx / 5xx Not OK)

```
1 {
2   "traceId": "b5feb4c7-5c58-49a9-84fe-891f4f76e659",
3   "message": "Some error message",
4 }
```

2.2 Reject Operations

2.2.1 Reject without ElectronicId

Endpoint: POST /api/fiscalization/rejectWithoutElectronicId

Important Note: The value of companyId must be the same as recipientIdentifierValue.

Request:

```
1 {
2   "username": "user@mail.com",
3   "password": "mystrongpass",
4   "companyId": "2222222222",
5   "softwareId": "software-123",
6   "internalMark": "INV-2025-001",
7   "issueDate": "2025-06-15",
8   "senderIdentifierValue": "1111111111",
9   "recipientIdentifierValue": "2222222222",
10  "rejectionDate": "2025-06-20",
11  "rejectionReasonType": "U",
12  "rejectionReasonDescription": "Invoice contains incorrect tax calculation."
13 }
```

Note: For possible values of rejectionReasonType, see section 5.2.

Response:

Success (200 OK)

```
1 {
2   "isSuccess": true,
3   "fiscalizationTimestamp": "2025-06-15T14:30:25Z",
4   "encodedXml": "PD94bWgdmVyc2lrbj0iMS4wIiB..."
5 }
```

Error (4xx / 5xx Not OK)

```
1 {
2   "traceId": "b5feb4c7-5c58-49a9-84fe-891f4f76e659",
3   "message": "Some error message",
4 }
```

2.2.2 Reject with ElectronicId

Endpoint: POST /api/fiscalization/reject

Request:

```
1 {
2   "username": "user@mail.com",
3   "password": "mystrongpass",
4   "companyId": "123456",
5   "softwareId": "software-123",
6   "electronicId": "121111131",
7   "rejectionDate": "2025-06-20",
8   "rejectionReasonType": "U",
9   "rejectionReasonDescription": "Invoice contains incorrect tax calculation."
10 }
```

Note: For possible values of rejectionReasonType, see section 5.2.

Response:
Success (200 OK)

```
1 {  
2   "isSuccess": true,  
3   "fiscalizationTimestamp": "2025-06-15T14:30:25Z",  
4   "encodedXml": "PD94bWgdmVyc2lvbj0iMS4wIiB..."  
5 }
```

Error (4xx / 5xx Not OK)

```
1 {  
2   "traceId": "b5feb4c7-5c58-49a9-84fe-891f4f76e659",  
3   "message": "Some error message",  
4 }
```

2.3 Report Operation

Endpoint: POST /api/fiscalization/eReporting

Request:

```
1 {  
2   "username": "user@mail.com",  
3   "password": "mystrongpass",  
4   "companyId": "123456",  
5   "softwareId": "software-123",  
6   "xmlInvoice": "<?xml version='1.0' encoding='Utf-8'...",  
7   "deliveryDate": "2025-06-20",  
8   "isCopy": false,  
9   "invoiceType": "U"  
10 }
```

Note: For possible values of invoiceType, see section 5.3.

Response:
Success (200 OK)

```
1 {  
2   "isSuccess": true,  
3   "reportingTimestamp": "2025-06-15T14:30:25Z",  
4   "encodedXml": "PD94bWgdmVyc2lvbj0iMS4wIiB..."  
5 }
```

Error (4xx / 5xx Not OK)

```
1 {  
2   "traceId": "b5feb4c7-5c58-49a9-84fe-891f4f76e659",  
3   "message": "Some error message",  
4 }
```

3 Check Identifier

Endpoint: POST /api/mps/check

Description: This endpoint checks if a given identifier is registered in the AMS system and returns its registration status. If the identifier is found, it returns a 200 OK status; otherwise, it returns an appropriate error status.

Response statuses:

- **200 OK** — The identifier is registered in the system.
- **4xx / 5xx NOT OK** — Error occurred (e.g., identifier not found, invalid input, server error).

Example Request:

```
POST /api/mps/check
{
  "username": "user@mail.com",
  "password": "mystrongpass",
  "companyId": "2222222222",
  "softwareId": "software-123",
  "identifierType": 0,
  "identifierValue": "1111111111"
}
```

Note: For possible values of `identifierType`, see section 5.4.

Response:

Success (200 OK)

```
1 {
2 }
```

Error (4xx / 5xx Not OK)

```
1 {
2   "traceId": "b5feb4c7-5c58-49a9-84fe-891f4f76e659",
3   "message": "Some error message",
4 }
```

4 Get Fiscalization Status

Endpoint: POST /api/fiscalization/status

Description: This endpoint retrieves the fiscalization status of a document using its ElectronicId and MessageType.

Response statuses:

- **200 OK** — Fiscalization status retrieved successfully.
- **4xx / 5xx NOT OK** — Error occurred (e.g., fiscalization status not found, invalid input, server error).

Example Request:

```
POST /api/fiscalization/status
{
  "username": "user@mail.com",
  "password": "mystrongpass",
  "companyId": "22222222222",
  "softwareId": "software-123",
  "electronicId": 121111131,
  "messageType": 0
}
```

Note: For possible values of messageType, see section 5.5.

Response:

Success (200 OK)

```
1 {
2   "isSuccess": true,
3   "fiscalizationTimestamp": "2025-06-20T16:45:30Z",
4   "encodedXml": "PD94bWwgdmVyc2lvbj0iMS4wIiB..."
5 }
```

Error (4xx / 5xx Not OK)

```
1 {
2   "traceId": "b5feb4c7-5c58-49a9-84fe-891f4f76e659",
3   "message": "Some error message",
4 }
```

Error (401 Unauthorized) Cases:

1. Invalid username or password
2. Invoice SENDER trying to get fiscalization status as RECIPIENT

```
1 {
2   "traceId": "b5feb4c7-5c58-49a9-84fe-891f4f76e659",
3   "message": "Some error message",
4 }
```

5 Common Enumerations

5.1 PaymentMethod

Value	HR Description
"T"	Transakcijski racun
"O"	Obracunsko placanje
"Z"	Ostalo

5.2 RejectionReasonType

Value	Description
"N"	Data discrepancy that does not affect tax calculation
"U"	Data discrepancy that affects tax calculation
"O"	Other

5.3 InvoiceType

Value	Description
"U"	An incoming invoice
"I"	An outgoing invoice
"IR"	An outgoing invoice for which an eInvoice was not issued

5.4 IdentifierType

Value	Description
0	OIB - Croatian personal identification number
1	GLN Global Location Number

5.5 MessageType

Value	Description
0	As SENDER get the status of the fiscalization message
1	As RECIPIENT get the status of the fiscalization message
2	Get the rejection status
3	Get the payment status

6 Search Classification

Description: This endpoint allows you to search classification data by either a specific **code** or a descriptive keyword.

- **Search by code** — e.g., 01.45.1
- **Search by keyword** — e.g., ovce

Authentication

All requests must include the **X-API-KEY** header with a valid API key.

- **Header:** X-API-KEY — required for authentication
- **Query Parameter:** searchString — the value to search for

Example Requests

Search by Code

```
GET api/Fiscalization/GetKpdData?searchString=01.45.1
X-API-KEY: <your-api-key>
```

Response

```
1 {
2   "status": "success",
3   "results": [
4     {
5       "code": "01.45.1",
6       "name": "Ovce i koze, žive"
7     },
8     {
9       "code": "01.45.11",
10      "name": "Ovce, žive"
11    },
12    {
13      "code": "01.45.12",
14      "name": "Koze, žive"
15    }
16  ]
17 }
```

Search by Keyword

```
GET api/Fiscalization/GetKpdData?searchString=ovce
X-API-KEY: <your-api-key>
```

Response

```
1 [
2   {
3     "code": "01.45",
4     "name": "Ovce i koze, žive; sirovo ovčje i kozje mlijeko i šišana ovčja i
        kozja vuna"
5   },
6   {
7     "code": "01.45.1",
8     "name": "Ovce i koze, žive"
9   },
10  {
11    "code": "01.45.11",
12    "name": "Ovce, žive"
13  }
14 ]
```

Possible Errors

Error (4xx / 5xx Not OK)

```
1 {
2   "traceId": "b5feb4c7-5c58-49a9-84fe-891f4f76e659",
3   "message": "Some error message",
4 }
```

7 Validation System

Website: <https://validate-demo.moj-eracun.hr/>

Purpose & Functionality

This demo web application allows users to validate XML-formatted electronic invoices before submission. It ensures that invoices follow the correct structure and standards, providing clear feedback on errors.

User Flow

1. Go to the home page.
2. Upload an XML file and select the validation type.
3. Click **“Validate XML Document”**.

The system then analyzes the file and displays the readability of the document, giving comprehensive validation results with details error messages and line-by-line feedback.

Key Features

- Schema validation for electronic invoices.
- Error reporting with detailed, line-by-line diagnostics.
- Demo environment for testing invoice XMLs before real submission.