

MailXstream Web Services

Version 3.0 Date: 2026-02-01

Background

MailXstream is a simple and powerful online print-to-mail service that allows you to outsource routine printing and mailing processes. It accepts digital PDF or PostScript documents via an Internet web service. These documents are processed, validated, verified, sorted, printed, inserted into envelopes, and mailed.

This document defines a protocol that can be used to automate the document and specification upload process, allowing end users to create a fully automated solution for print-to-mail services.

Templates

Templates define the processing and finishing of uploaded documents. These are created by eVanguard when a customer registers for MailXstream services. They define:

- Processing specifications (paper, ink, envelopes, etc.)
- Data extraction parameters (address, identifier, etc.)

Documents

- Documents must be in **PDF 1.4** format or later.
- Fonts should be embedded as full fonts (not subsetted) except for the standard 14 PDF fonts.
- If a font is not embedded, MailXstream will try to match the font name with their current set; otherwise the document may be rejected.

Document Collections

Multiple documents may be combined and uploaded in a number of "collection" formats:

Single PDF Document File - An individual PDF document uploaded one at a time. The file may be compressed with "zip" or "gzip". A custom template is required to define the document identifier location, the delivery address location, any other extraction parameters, and processing specifications. Additional "soft inserts" as PDF files may be included.

Single PDF File with Multiple Documents - The documents are uploaded as a single PDF file with many documents. This is typical of what would be sent to a local printer as the output of a mail-merge or the printing of many invoices or statements from an accounting package such as QuickBooks. The file may be compressed with "zip" or "gzip" to reduce the required network bandwidth. A custom template is required to define the "start of document" identifier location and value, the document identifier location, the delivery address location, any other extraction parameters, and processing specifications. Additional "soft inserts" as PDF files may be included.

Single File with Multiple PDF Document Files - The individual PDF documents are combined into a "zip" file or a "tar" file with or without "gzip" compression. A custom template is required to define the document identifier location, the delivery address location, any other extraction parameters, and processing specifications. Additional "soft inserts" as PDF files may be included.

Single File with Multiple PDF Document and Meta Data Files - The individual PDF documents along with a matching meta data file are combined into a "zip" file or a "tar" file with or without "gzip" compression. The metadata file contains the data fields (identifier, to address, etc) for each document. The meta data file name must be the same as the PDF file with a "meta" extension. A custom template is required to define the processing specifications. Additional "soft inserts" as PDF files may be included.

Single Document File with Metadata - An individual PDF document uploaded one at a time. The file may be compressed with "zip" or "gzip". Additional meta data is included with the upload to define the data fields (delivery address, identifier, etc) and processing specifications (paper, ink, etc). A standard template is required to define default data fields and processing parameters. Additional "soft inserts" as PDF files may be included.

Single PDF Document File with Address List (Mass Mailing) - The individual PDF document and an address list CSV file are uploaded. The document will be sent to each address in the address list file. A standard template is required to define default data fields and processing parameters. See "Address_File_Specification" for details. Additional "soft inserts" as PDF files may be included.

Protocol

- **Transport:** REST over HTTPS (port 443). HTTP (port 80) is not supported.
- **Test URL:** `https://test.mailxstream.com/api/v3/`
- **Production URL:** `https://prod.mailxstream.com/api/v3/`

Authentication

Authentication is provided through access tokens. An access token can be created on the user's profile page and must be provided as a header:

```
Authorization: Bearer <access-token>
```

Responses

All responses are JSON objects.

- Successful responses will have a status code in the 200 family. See each endpoint for the structure of the response.
- On error, the response will have a status code in the 400 or 500 family depending on whether the error is client-side or server-side.

Error Response Schema:

```
{
  "message": "The overall error message",
  "fieldName": "field_name",
```

```

    "itemId": "item_id",
    "errors": [
      {
        "message": "Specific error message",
        "fieldName": "field_name"
      }
    ]
  }

```

Field	Type	Description
message	string	The overall error message
fieldName	string	Only present if the error relates to a specific field
itemId	string	Only present if the error relates to a specific item
errors	array	Present when multiple errors are returned for a single call

Service Methods

The MailXstream Web Services exposes the following methods:

Method	Endpoint	Description
FindTemplates	GET /v3/templates	Retrieve a list of valid template names
SubmitOrder	POST /v3/order/submit	Submit a new print order
GetCurrentUser	GET /v3/me	Retrieve information about the authenticated user
FindOrders	GET /v3/order	Retrieve order IDs, names, and templates
GetOrderStatus	GET /v3/order/:orderId/status	Retrieve the current status of an order
GetOrderDetails	GET /v3/order/:orderId	Retrieve detailed information about an order
		Retrieve detailed

GetDocumentDetails	GET /v3/document/:documentId	information about a document
GetOrderCOM	GET /v3/order/:orderId/com	Retrieve the Certificate of Mailing (PDF) for an order
CancelOrder	DELETE /v3/order/:orderId/cancel	Cancel a specified order
CancelDocument	DELETE /v3/document/:documentId/cancel	Cancel a specified document

Endpoint Documentation

FindTemplates

Purpose: Retrieve a list of valid template names for the authenticated user's account.

URL: GET /v3/templates

Query Parameters

Parameter	Type	Required	Description
templateName	string	No	Filter templates by name pattern
includeFields	boolean	No	When true, returns template fields and override options

Response

200 OK (without includeFields)

```
{
  "templates": ["InvoiceTemplate", "StatementTemplate", "NoticeTemplate"]
}
```

200 OK (with includeFields=true)

```
[
  {
    "name": "InvoiceTemplate",
    "fields": [
      {
        "FIELD_NAME": "INK",
        "FIELD_TYPE": "SELECT",
        "FIELD_VALUES": ["BLACK", "COLOR"],
        "MAIN_DISPLAY": "TRUE"
      }
    ]
  },
]
```

```

      "defaultReturnAddress": "ACME Inc.\n123 Main St\nSpringfield, MO 65801"
    }
  ]
}

```

400 Bad Request

```

{
  "message": "No client context"
}

```

Example

```

curl -X GET "https://prod.mailxstream.com/api/v3/templates" \
-H "Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..."

```

```

curl -X GET "https://prod.mailxstream.com/api/v3/templates?
includeFields=true&templateName=Invoice" \
-H "Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..."

```

SubmitOrder

Purpose: Submit a new order for processing.

URL: `POST /v3/order/submit`

Note: This must be a multipart/form-data POST request.

Request Body (multipart/form-data)

Field	Type	Required	Description
name	string	Yes	Unique name for this order. Use "auto" to generate automatically.
template	string	Yes	Template name or template object ID (e.g., <code>ORT_123456</code>). For shared templates, use format <code>clientId:templateName</code> .
input	file	Yes	Main document file (PDF, may be compressed with zip or gzip)
testFlag	string	No	Set to "true" for a test upload (only when directed by support)
insert1...insertN	file	No	Additional PDF files to include as soft inserts
RET_ADDRESS	string	No	Return address override (overrides template return address)
ADDFILE	file	No	CSV address file (required for mass-mail uploads)
DEL_ADDRESS	string	No	Delivery address lines (required for single-document-with-metadata uploads)
INK	string	No	Ink override for templates that support it. Allowed values: <code>BLACK</code> , <code>COLOR</code> .

PLEX	string	No	Plex (sided printing) override for templates that support it. Allowed values: <code>SIMPLEX</code> , <code>DUPLEX</code> .
------	--------	----	--

Additional override fields may be available depending on the template configuration. Contact support for possible values for a specific template.

Supplying Addresses

There are three ways to provide address information on a submit, depending on the upload mode. They can be combined — for example, a mass-mailing upload can supply both `RET_ADDRESS` (one shared return address) and `ADDFILE` (per-recipient delivery addresses).

1. Return address (`RET_ADDRESS`)

A single multi-line string that overrides the return address baked into the template. Lines are separated by `\n`. Applies to every document in the order.

```
-F "RET_ADDRESS=ACME Corp
123 Main St
Springfield, IL 62701"
```

2. Delivery address (`DEL_ADDRESS`)

A single multi-line string for the delivery address. Used with the "single document with metadata" upload mode, where the document itself does not carry an embedded delivery address. Lines are separated by `\n`.

```
-F "DEL_ADDRESS=Brian Ziems
53555 865 Rd
Plainview, NE 68769"
```

3. Address list file (`ADDFILE`)

A CSV file containing one delivery address per row. Used with the mass-mailing upload mode: the input document is sent once to each address in the file. The file is attached as a multipart upload, not a text field. See **Address File Specification** for the column layout.

```
-F "ADDFILE=@/path/to/recipients.csv"
```

Response

200 OK

```
{
  "orderId": "ORD_17563068339779DEV",
  "jobName": "Invoice-March-2026"
}
```

Field	Type	Description
orderId	string	Unique order identifier
jobName	string	The assigned order name

400 Bad Request

```
{
  "message": "Name and template are required."
}
```

```
{
  "message": "Template not found."
}
```

```
{
  "message": "Order name is already used."
}
```

```
{
  "message": "Missing document file"
}
```

500 Internal Server Error

```
{
  "message": "Unable to submit order"
}
```

Example

```
curl -X POST "https://prod.mailxstream.com/api/v3/order/submit" \
-H "Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..." \
-F "name=Invoice-March-2026" \
-F "template=InvoiceTemplate" \
-F "input=@/path/to/invoices.pdf" \
-F "INK=BLACK" \
-F "PLEX=DUPLEX"
```

```
curl -X POST "https://prod.mailxstream.com/api/v3/order/submit" \
-H "Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..." \
-F "name=auto" \
-F "template=ACME:SharedInvoiceTemplate" \
-F "input=@/path/to/invoices.pdf" \
-F "insert1=@/path/to/terms.pdf"
```

FindOrders

Purpose: Retrieve order IDs, names, and template names for matching orders.

URL: GET /v3/order

Query Parameters

Parameter	Type	Required	Description
orderName	string	No	Order name or pattern. Wildcards: * or %
templateName	string	No	Template name or pattern to filter by
			Start of search date range (ISO 8601

startDate	string	No	format). Defaults to epoch.
endDate	string	No	End of search date range (ISO 8601 format). Defaults to now.
state	string	No	Filter by processing state (INIT, SEGMENT, SORT, PRINT, MAIL, COMPLETE)
status	string	No	Filter by status within the state (WAIT, RUN, ERROR, COMPLETE)
actionRequired	boolean	No	When <code>true</code> , return only orders awaiting manual approval (overrides state/status)
limit	number	No	Maximum number of orders to return, between 1 and 10000 (default 10000).

Response

200 OK

```
{
  "orders": [
    {
      "templateName": "InvoiceTemplate",
      "orderName": "Invoice-March-2026",
      "orderId": "ORD_17563068339779DEV",
      "state": "SEGMENT",
      "status": "WAIT",
      "date": 1757451399,
      "errorFlag": false,
      "actionRequired": true,
      "testFlag": false
    }
  ]
}
```

Orders are returned most-recently-updated first.

Field	Type	Description
orders	array	Array of order summary objects
orders[].templateName	string	Name of the template used
orders[].orderName	string	Name of the order
orders[].orderId	string	Unique order identifier
orders[].state	string	Current processing state (INIT, SEGMENT, SORT, PRINT, MAIL, COMPLETE)
orders[].status	string	Status within the state (WAIT, RUN, ERROR, COMPLETE)
orders[].date	number	Unix timestamp of the last state change
orders[].errorFlag	boolean	True if the order has errors
orders[].actionRequired	boolean	True if the order is waiting for manual approval

orders[].testFlag	boolean	True if this is a test order
-------------------	---------	------------------------------

400 Bad Request

```
{
  "message": "No client context"
}
```

Example

```
curl -X GET "https://prod.mailxstream.com/api/v3/order?
orderName=Invoice*&startDate=2026-01-01" \
-H "Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..."
```

GetOrderStatus

Purpose: Retrieve the current state and status of a specific order.

URL: GET /v3/order/:orderId/status

Path Parameters

Parameter	Type	Required	Description
orderId	string	Yes	Unique order identifier

Response

200 OK

```
{
  "orderId": "ORD_17563068339779DEV",
  "orderName": "Invoice-March-2026",
  "state": "SEGMENT",
  "status": "WAIT",
  "date": 1757451399,
  "errorFlag": false,
  "testFlag": false
}
```

Field	Type	Description
orderId	string	Unique order identifier
orderName	string	Name of the order
state	string	Current processing state (INIT, SEGMENT, SORT, PRINT, MAIL, COMPLETE)
status	string	Status within the state (WAIT, RUN, ERROR, COMPLETE)
date	number	Unix timestamp of the last state change
errorFlag	boolean	True if the order has errors
testFlag	boolean	True if this is a test order

--	--	--

404 Not Found

```
{
  "message": "Order not found"
}
```

Example

```
curl -X GET
"https://prod.mailxstream.com/api/v3/order/ORD_17563068339779DEV/status" \
-H "Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..."
```

GetOrderDetails

Purpose: Retrieve detailed information about a specific order, optionally including all documents.

URL: GET /v3/order/:orderId

Path Parameters

Parameter	Type	Required	Description
orderId	string	Yes	Unique order identifier

Query Parameters

Parameter	Type	Required	Description
includeDocs	string	No	Set to "1" to include document details in the response

Response

200 OK (Base Response)

```
{
  "orderId": "ORD_17563068339779DEV",
  "orderName": "Invoice-March-2026",
  "state": "SEGMENT",
  "status": "WAIT",
  "date": 1757451399,
  "errorFlag": false,
  "testFlag": false,
  "softInserts": ["terms.pdf", "promo.pdf"],
  "errorMsgs": [],
  "stateArray": [
    {
      "state": "CREATE",
      "status": "COMPLETE",
      "date": 1756306784
    },
    {
      "state": "SEGMENT",
```

```

        "status": "WAIT",
        "date": 1757451399
    }
],
"pieces": 150,
"impressions": 300,
"prodPieces": 150,
"prodImpressions": 300
}

```

Field	Type	Description
orderId	string	Unique order identifier
orderName	string	Name of the order
state	string	Current processing state
status	string	Status within the state
date	number	Unix timestamp of the last state change
errorFlag	boolean	True if the order has errors
testFlag	boolean	True if this is a test order
softInserts	array	List of soft insert filenames
errorMsgs	array	List of error messages
stateArray	array	History of state transitions
pieces	number	Total number of mail pieces
impressions	number	Total number of impressions
prodPieces	number	Number of pieces in production
prodImpressions	number	Number of impressions in production

Extended Response (when includeDocs=1)

```

{
  "orderId": "ORD_17563068339779DEV",
  "orderName": "Invoice-March-2026",
  "state": "MAIL",
  "status": "COMPLETE",
  "date": 1757451399,
  "errorFlag": false,
  "testFlag": false,
  "softInserts": [],
  "errorMsgs": [],
  "stateArray": [
    {
      "state": "CREATE",
      "status": "COMPLETE",
      "date": 1756306784
    }
  ],
  "pieces": 2,
  "impressions": 4,
  "prodPieces": 2,

```

```

    "prodImpressions": 4,
    "documents": [
      {
        "trackingId": "INV-2026-0001",
        "documentId": "DOC_17563068080089DEF",
        "impressions": 2,
        "errorFlag": false
      },
      {
        "trackingId": "INV-2026-0002",
        "documentId": "DOC_17563068080090DEF",
        "impressions": 2,
        "errorFlag": false,
        "carrierTracking": {
          "carrier": "USPS",
          "trackingId": "9400111899223456789012",
          "link": "https://tools.usps.com/go/TrackConfirmAction?tLabels=9400111899223456789012"
        }
      }
    ]
  }
}

```

Field	Type	Description
documents	array	Array of document objects (only when includeDocs=1)
documents[].trackingId	string	Document tracking identifier
documents[].documentId	string	Unique document identifier
documents[].impressions	number	Number of impressions for this document
documents[].errorFlag	boolean	True if the document has errors
documents[].carrierTracking	object	Carrier tracking info (only for certified/priority mail)

404 Not Found

```

{
  "message": "Order not found"
}

```

Example

```

curl -X GET "https://prod.mailxstream.com/api/v3/order/ORD_17563068339779DEV" \
-H "Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..."

```

```

curl -X GET "https://prod.mailxstream.com/api/v3/order/ORD_17563068339779DEV?includeDocs=1" \
-H "Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..."

```

GetDocumentDetails

Purpose: Retrieve detailed information about a specific document.

URL: GET /v3/document/:documentId

Path Parameters

Parameter	Type	Required	Description
documentId	string	Yes	Unique document identifier

Response

200 OK (Base Response)

```
{
  "trackingId": "INV-2026-0001",
  "documentId": "DOC_17563067990078DEF",
  "impressions": 2,
  "errorFlag": false,
  "errorMsgs": [],
  "pages": 2,
  "duplex": true,
  "deliverAddress": [
    "JOHN DOE",
    "123 MAIN ST",
    "SPRINGFIELD MO 65801-9745"
  ],
  "returnAddress": [
    "ACME INC",
    "9999 CORPORATE BLVD",
    "STERLING VA 20165-5675"
  ],
  "emailAddress": "john.doe@example.com",
  "hardInserts": [],
  "softInserts": ["terms.pdf"],
  "paper": "LETTER_20LB",
  "ink": "BLACK",
  "envelope": "10S-2",
  "testFlag": false
}
```

Field	Type	Description
trackingId	string	Document tracking identifier
documentId	string	Unique document identifier
impressions	number	Number of impressions
errorFlag	boolean	True if the document has errors
errorMsgs	array	List of error messages
pages	number	Number of pages
duplex	boolean	True if printed duplex
deliverAddress	array	Delivery address lines
returnAddress	array	Return address lines
emailAddress	string	Associated email address (if any)

hardInserts	array	List of hard insert names
softInserts	array	List of soft insert filenames
paper	string	Paper type
ink	string	Ink type
envelope	string	Envelope type
testFlag	boolean	True if this is a test document

Extended Response (with carrier tracking)

For documents sent via USPS Certified Mail, USPS Priority Mail, or UPS:

```
{
  "trackingId": "INV-2026-0001",
  "documentId": "DOC_17563067990078DEF",
  "impressions": 2,
  "errorFlag": false,
  "errorMsgs": [],
  "pages": 2,
  "duplex": true,
  "deliverAddress": [
    "JOHN DOE",
    "123 MAIN ST",
    "SPRINGFIELD MO 65801-9745"
  ],
  "returnAddress": [
    "ACME INC",
    "9999 CORPORATE BLVD",
    "STERLING VA 20165-5675"
  ],
  "emailAddress": "",
  "hardInserts": [],
  "softInserts": [],
  "paper": "LETTER_20LB",
  "ink": "BLACK",
  "envelope": "10S-2",
  "testFlag": false,
  "carrierTracking": {
    "carrier": "USPS",
    "trackingId": "9400111899223456789012",
    "link": "https://tools.usps.com/go/TrackConfirmAction?tLabels=9400111899223456789012"
  }
}
```

Field	Type	Description
carrierTracking	object	Present only for certified/priority mail
carrierTracking.carrier	string	Carrier name (USPS, UPS)
carrierTracking.trackingId	string	Carrier tracking number
carrierTracking.link	string	URL to track the shipment

404 Not Found



```
{
  "message": "Document not found"
}
```

Example

```
curl -X GET "https://prod.mailxstream.com/api/v3/document/DOC_17563067990078DEF
\
-H "Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..."
```

GetOrderCOM

Purpose: Retrieve the MailXstream Certificate of Mailing (PDF) for a given order.

URL: GET /v3/order/:orderId/com

Path Parameters

Parameter	Type	Required	Description
orderId	string	Yes	Unique order identifier

Response

200 OK

Returns a PDF file with content type `application/pdf`.

Note: The Certificate of Mailing is generated on first request and may take time for large orders. If the request times out, retry after a delay.

404 Not Found

```
{
  "message": "Order not found"
}
```

400 Bad Request

```
{
  "message": "Certificate of Mailing not available for this order"
}
```

Example

```
curl -X GET "https://prod.mailxstream.com/api/v3/order/ORD_17563068339779DEV/co
\
-H "Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..." \
-o certificate-of-mailing.pdf
```

CancelOrder

Purpose: Cancel a pending order. The order must be in `INIT`, `SEGMENT`, or `SORT` state with `WAIT` or `ERROR` status.

URL: `DELETE /v3/order/:orderId/cancel`

Path Parameters

Parameter	Type	Required	Description
orderId	string	Yes	Unique order identifier

Response

202 Accepted

```
{
  "orderId": "ORD_17563068339779DEV"
}
```

400 Bad Request

```
{
  "message": "Order is not in a cancelable state - PRINT/RUN"
}
```

404 Not Found

```
{
  "message": "Order not found"
}
```

500 Internal Server Error

```
{
  "message": "Failed to cancel order"
}
```

Example

```
curl -X DELETE
"https://prod.mailxstream.com/api/v3/order/ORD_17563068339779DEV/cancel" \
-H "Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..."
```

CancelDocument

Purpose: Cancel a single document within an order. The order must be in `INIT`, `SEGMENT`, or `SORT` state with `WAIT` or `ERROR` status.

URL: `DELETE /v3/document/:documentId/cancel`

Path Parameters

Parameter	Type	Required	Description
documentId	string	Yes	Unique document identifier

Response

202 Accepted

```
{
  "documentId": "DOC_17563067990078DEF"
}
```

400 Bad Request

```
{
  "message": "Order containing document is not in a cancelable state - PRINT/RU"
}
```

404 Not Found

```
{
  "message": "Document not found"
}
```

500 Internal Server Error

```
{
  "message": "Failed to cancel document"
}
```

Example

```
curl -X DELETE
"https://prod.mailxstream.com/api/v3/document/DOC_17563067990078DEF/cancel" \
-H "Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..."
```

GetCurrentUser

Purpose: Retrieve information about the authenticated user. Useful for verifying an access token and identifying the connected account.

URL: GET /v3/me

Response

200 OK

```
{
  "userId": "jane",
  "name": "Jane Doe",
  "client": {
    "name": "ACME Corp"
  }
}
```

Field	Type	Description
userId	string	The login id of the authenticated user
name	string	The user's full name
client	object	The user's client (null when the account is not a client user)

Example

```
curl -X GET "https://prod.mailxstream.com/api/v3/me" \
-H "Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..."
```

Common Response Codes

Code	Status	Description
200	OK	Request succeeded
202	Accepted	Request accepted for processing
400	Bad Request	Invalid request parameters or business rule violation
401	Unauthorized	Missing or invalid authentication token
403	Forbidden	Authenticated but not authorized for this resource
404	Not Found	Requested resource does not exist
500	Internal Server Error	Server-side error occurred

Schema Reference

ErrorResponse

```
{
  "message": "string",
  "fieldName": "string (optional)",
  "itemId": "string (optional)",
  "errors": [
    {
      "message": "string",
      "fieldName": "string (optional)"
    }
  ]
}
```

OrderSummary

```
{
  "templateName": "string",
  "orderName": "string",
  "orderId": "string",
  "state": "string",
  "status": "string",
  "date": "number (unix timestamp)",
  "errorFlag": "boolean",
  "actionRequired": "boolean",
  "testFlag": "boolean"
}
```

OrderStatus

```
{
  "orderId": "string",
  "orderName": "string",
  "state": "string",
  "status": "string",
  "date": "number (unix timestamp)",
  "errorFlag": "boolean",
  "testFlag": "boolean"
}
```

OrderDetails

```
{
  "orderId": "string",
  "orderName": "string",
  "state": "string",
  "status": "string",
  "date": "number (unix timestamp)",
  "errorFlag": "boolean",
  "testFlag": "boolean",
  "softInserts": ["string"],
  "errorMsgs": ["string"],
  "stateArray": [
    {
      "state": "string",
      "status": "string",
      "date": "number (unix timestamp)"
    }
  ],
  "pieces": "number",
  "impressions": "number",
  "prodPieces": "number",
  "prodImpressions": "number"
}
```

OrderDetailsExtended

Extends OrderDetails with:

```
{
  "documents": [
    {
      "trackingId": "string",
      "documentId": "string",
      "impressions": "number",
      "errorFlag": "boolean",
      "carrierTracking": "CarrierTracking (optional)"
    }
  ]
}
```

DocumentDetails

```
{
  "trackingId": "string",
  "documentId": "string",
  "impressions": "number",
  "errorFlag": "boolean",
}
```

```
"errorMsgs": ["string"],
"pages": "number",
"duplex": "boolean",
"deliverAddress": ["string"],
"returnAddress": ["string"],
"emailAddress": "string",
"hardInserts": ["string"],
"softInserts": ["string"],
"paper": "string",
"ink": "string",
"envelope": "string",
"testFlag": "boolean"
}
```

DocumentDetailsExtended

Extends DocumentDetails with:

```
{
  "carrierTracking": {
    "carrier": "string",
    "trackingId": "string",
    "link": "string"
  }
}
```

CarrierTracking

```
{
  "carrier": "string",
  "trackingId": "string",
  "link": "string"
}
```

TemplateInfo

```
{
  "name": "string",
  "fields": [
    {
      "FIELD_NAME": "string",
      "FIELD_TYPE": "string",
      "FIELD_VALUES": ["string"],
      "MAIN_DISPLAY": "string"
    }
  ],
  "defaultReturnAddress": "string"
}
```