



PADN

A Publishers Data Network

API Documentation

11th January 2024
Phillip Darrall
Papermule Ltd
Version 19

Contents

1.Introduction	3
2.Document Change Log	3
3.API Overview	4
3.1.API Access Security	4
3.2.API Responses	4
4.API Calls	5
4.1.Get Publishers	5
4.2.Get Ads	6
4.3.Confirm Sender	8
4.4.Reject Sender	9
4.5.Get Record	10
Appendix (Service Level Agreement)	11
Service Description	12
Hours of Cover	13
Service Response Priorities Matrix	14
Impact of Service	14
Severity	15

1. Introduction

This document provides details about the APIs exposed to intermediary third parties. The Phase I project scope is extremely constrained and as such the initial API offering is equally limited.

The aims behind Phase I:

- Expose advertising production data to through intermediary third parties to ad senders in a manner controlled by the publisher.
- Allow restricted use of that data in any way the intermediary sees fit in improving the chasing and supply of advertising content to a publisher
- Enable the intermediary to mark individual ad records at a publisher with copy contact details and free text note.
- Enable the intermediary to mark individual ad records at a publisher as NOT being supplied by a given person

2. Document Change Log

Doc Version	Revision Date	Who	Comment
1.5	2017-06-06	Mike Hoy	Intro, Change Log added
7	2018-07-02	Phillip Darrall	Changes to ads method
8	2018-09-13	Phillip Darrall	Additional fields returned from ads method
9	2018-10-04	Phillip Darrall	Added id restrictor to ads method
10	2018-10-05	Phillip Darrall	Additional fields returned from ads method
11	2019-03-01	Phillip Darrall	Added new record method
12	2019-11-19	Phillip Darrall	Additional fields in ads and record methods
13	2020-01-24	Phillip Darrall	Added SLA
14	2020-10-08	Phillip Darrall	Addition fields / restrictors for ads method
15	2020-10-12	Phillip Darrall	Updated ads restrictors
16	2022-02-14	Phillip Darrall	Additional fields in ads / record methods
17	2022-09-21	Phillip Darrall	Additional fields in ads / record methods
18	2022-10-20	Phillip Darrall	Clarify /ads parameters
19	2024-01-11	Phillip Darrall	Additional /ads parameters

3. API Overview

3.1. API Access Security

The individual API key (auth token) provided by Papermule to each intermediary will be required as a parameter for each request and must be kept secret. As such all requests for the API must be server side (cannot be made from a browser) so that the key is not exposed publicly.

3.2. API Responses

The API calls will all respond with a JSON object along with a status in the response headers.

There will be a header "X-Papermule-Status" with value "OK" or "ERROR" depending on outcome.

If the status is OK, and a result set is returned, then there will be an additional header "X-Papermule-Rows" containing the number of rows in the result set.

The JSON response will be either:

```
{"status":"OK","data":[...]}
```

or

```
{"status":"ERROR","data":"reason"}
```

The base URL for all of the following API calls is:

<https://api.mediamule.co.uk/muletrain/publicapi>

4. API Calls

4.1. Get Publishers

Description: Gets a list of Publishers that may be queried giving their Name and Reference.

```
GET /publishers?key={auth_token}
```

parameters = key (required)

Example output:

```
{
  "status": "OK",
  "data":
  [
    {
      "publisher": "MULE"
      "publisherName": "Mediamule",
    }
  ]
}
```

4.2. Get Ads

Description: Gets a list of ads from a given Publisher for the provided Sender that the given publisher wishes to share. It is possible to restrict returned results via the status, days, id, urn, tpref, publication, edition and productionStatus parameters.

```
GET /ads?key={auth_token}&publisher={publisher_ref}&sender={sender_email}
```

```
GET
/ads?key={auth_token}&publisher={publisher_ref}&sender={sender_email}&status={P
EDICTED | CONFIRMED |
REJECTED}&days={number_of_days}&id={id_value}[&urn={urn}&publication={publicatio
n}&edition={edition}][&tpref={3rd_party_ref}&publication={publication}&edition={
edition}&productionStatus={required_production_status_list}]
```

parameters = key (required), publisher (required), sender (required), status (optional), days (optional), id (optional), urn (optional), tpref (optional), publication (required if urn or tpref provided), edition (required if urn or tpref provided), productionStatus (a comma-separated list of production statuses that should be included in the results - e.g AwaitingCopy,Finished)

Example output:

```
{
  "status": "OK",
  "data":
  [
    {
      "id": "747452",
      "urn": "12345",
      "publication": "PE",
      "publicationName": "Papermule Express",
      "edition": "2017-02-22",
      "deadline": "2017-01-01 17:00:00",
      "width": "225.0",
      "depth": "102.0",
      "sizeUnits": "mm",
      "size": "Half Page Horizontal (102.0mm x 225.0mm)",
      "columns": "4",
      "colour": "Colour",
      "brand": "Test Brand",
      "sectionType": "Display",
      "section": "ROP",
      "sectionName": "ROP",
      "advertiser": "43",
      "advertiserName": "Test Company",
      "padnAdvertiser": "42",
      "padnAdvertiserName": "Test Advertiser",
      "productionStatus": "AwaitingCopy",
      "sender": "test@test.co.uk",
      "senderStatus": "Predicted",
      "description": "Additional advert information",
      "isOption": "N",
      "created": "2016-12-01 12:34:56",
      "tpref": "3763402",
      "productionName": "174267",
      "bleed":
      {
        "bleedMargin": "3.0",
        "topMargin": "9.0",
        "leftMargin": "15.0",
        "rightMargin": "15.0",
```

```
        "bottomMargin": "12.0"  
      }  
    }  
  ]  
}
```

The size field in the results varies depending on the available data from the publisher. It will be either a named size (e.g. "Full Page") followed by the depth x width in mm, a "column centimetre" value (e.g. "5x10") followed by the depth x width in mm or just the depth x width in mm.

The optional status parameter will restrict the results to ads that are either confirmed to be the provided sender (status=CONFIRMED), confirmed not to be the provided sender (status=REJECTED) or predicted to be the provided sender (status=PREDICTED) depending on the given value.

The optional days parameter will restrict the results to ads with a deadline within the provided number of days.

The optional urn parameter will restrict the results to ads with the value provided

The optional tpref parameter will restrict the results to ads with either a 3rd part ref (purchase order) or URN matching the given value.

Note: if the urn or tpref parameter is provided, then publication and edition parameters must also be provided. In this case, sender is still a required parameter, but is not used to restrict the results.

4.3. Confirm Sender

Description: Sets the 'confirmed Sender details' for one advert at one publisher to the provided Senders details.

```
POST /confirmsender?key={auth_token}&publisher={publisher_ref}
```

```
parameters = key (required), publisher (required)
```

```
payload = {"id":"ad_id","sender":{"name":"sender_name", "email":"sender_email",  
"phone":"sender_phone_no", "company":"sender_company"}, "note":"optional_note"}
```

Example output:

```
{  
  "status":"OK",  
  "data":  
  {  
    "message":"CONFIRMED"  
  }  
}
```


4.4. Reject Sender

Description: Sets the 'Sender Details' for one advert at one publishers as 'contact not sending this advert'. Can be used to reject a previous confirmation or on its own.

```
POST /rejectsender?key={auth_token}&publisher={publisher_ref}
```

```
parameters = key (required), publisher (required)
```

```
payload = {"id":"ad_id", "sender":"sender_email", "note":"optional_note"}
```

Example output:

```
{
  "status":"OK",
  "data":
  {
    "message":"REJECTED"
  }
}
```

4.5. Get Record

Description: Gets details of a particular record within a given publisher's AdDesk. The record is identified by means of the publisher and record id.

```
GET /record?key={auth_token}&publisher={publisher_ref}&id={record_id}
```

```
parameters = key (required), publisher (required), id (required)
```

Example output:

```
{
  "status":"OK",
  "data":
  {
    "id":"747452",
    "urn":"12345",
    "publication":"PE",
    "publicationName":"Papermule Express",
    "edition":"2017-02-22",
    "deadline":"2017-01-01 17:00:00",
    "width":"225.0",
    "depth":"102.0",
    "sizeUnits":"mm",
    "size":"Half Page Horizontal (102.0mm x 225.0mm)",
    "columns":"4",
    "colour":"Colour",
    "brand":"Test Brand",
    "sectionType":"Display",
    "section":"ROP",
    "sectionName":"ROP",
    "advertiser":"43",
    "advertiserName":"Test Company",
    "padnAdvertiser":"42",
    "padnAdvertiserName":"Test Advertiser",
    "productionStatus":"AwaitingCopy",
    "sender":"test@test.co.uk",
    "senderStatus":"Predicted",
    "description":"Additional advert information",
    "isOption":"N",
  }
}
```

```
"created":"2016-12-01 12:34:56",
"tpref":"3763402",
"productionName":"174267",
"bleed":
{
  "bleedMargin":"3.0",
  "topMargin":"9.0",
  "leftMargin":"15.0",
  "rightMargin":"15.0",
  "bottomMargin":"12.0"
}
}
```



Service Level Agreement



26th November 2019
Mike Hoy
Papermule Ltd
Version 1.0

Service Description

PADN is a distributed data distribution network based on a three-tier architectural model.

Tier One comprises of a number of Publishers individual AdDesk platforms.
Tier Two is the PADN hub facilitating security and communication routing.
Tier Three is the customer (vendor) infrastructure making the API calls.

Tier One

Each tier one platform providing PADN with source data is covered by Publisher specific service level agreements and these underpin each publisher's ability to share data via the PADN network. Depending on the Publisher the tier one provision may be covered by a combination of SLA's covering infrastructure, communications and applications. Where the PADN service is partially unavailable due to a tier one issue the issue will be resolved under the appropriate authorities SLA.

Tier Two

This Service Level Agreement covers the provision and availability of the PADN Hub.

Our PADN Hub Infrastructure consists of a cluster of autoscaling AWS EC2 servers and Multi AZ RDS database instance sheltered behind an AWS elastic load balancer. This infrastructure has been created to provide a highly resilient, distributed solution delivering a 24/7/365 service.

Our underlying infrastructure is based on the Amazon AWS SLA's for Amazons EC2, Elastic Load Balancers and Multi-AZ RDS:

- <https://aws.amazon.com/compute/sla/>
- <https://aws.amazon.com/elasticloadbalancing/sla/>
- <https://aws.amazon.com/rds/sla/>

The only service provided solely by the PADN Hub is the GET API call /publishers
This service should consistently deliver sub second responses.

All other API calls require PADN to make further calls to individual publishers' platforms each encountering varied performance of the communications, server and databases. We expect these channels to consistently provide results within seconds. Issues around performance to these API calls are dealt with under the individual Publishers SLA agreements.

Hours of Cover

Papermule provides support 24 hours a day, 365 days a year for Critical and Essential Service Response Priority issues with Significant, Regular and Normal Service Response Priority issues dealt with in normal Working Hours

Service Response Priorities Matrix

<u>Severity -></u> <u>Impact</u>	<u>Critical</u>	<u>Major</u>	<u>Minor</u>
<u>High</u>	Critical	Essential	Significant
<u>Medium</u>	Essential	Significant	Regular
<u>Low</u>	Significant	Regular	Normal

Papermule response levels;

Critical - Aim to respond and investigate immediately with a resolution within the hour.

Essential - Aim to respond and investigate within 30 minutes with a resolution within 2 hours.

Significant - Aim to respond and investigate within the hour with a resolution within 6 hours.

Regular - Aim to respond and investigate within a day with a resolution within 3 Business Days

Normal - Aim to respond and investigate within a day with a resolution within 10 Business Days.

Impact of Service

Papermule categorises the Impact of Service as a measure of how disruptive the incident is, e.g.

5. **Low** - Affecting a minority of users, minor functionality or service performance issues
6. **Medium** – many users suffering loss of functionality across multiple publishers
7. **High** – unavailability or severely degraded performance of the service rendering it unusable.

Severity

The Vendor consider Severity of the issue as a combination of available functionality and degree of records/assets being affected.

The levels of Severity are:

Minor

Implies work is still possible but occasional components / records / integrations suffer but not to the extent that has an impact on immediate business. An intermittent browser crash, connectivity issue, size or publication mapping missing or incorrect would be 'minor' as would occasional integration processing issues. In all minor cases a manual intervention should be able to rectify the issue.

Major

Would affect multiple unrelated records or an entire integration, user interface or key aspect of the service where a service that can be temporarily circumnavigated but if left unresolved would lead to the severity becoming critical.

Critical

If one or more aspects of the service have failed to a point where general production is no longer possible and manual circumnavigation of the issue is not possible.