



# PURPLE MANAGER API DOCUMENTATION UPLOAD PACKAGES



## CONTENTS

<b>1</b>	<b>INTRODUCTION .....</b>	<b>4</b>
1.1	DATA MODEL.....	4
1.2	WORKFLOW .....	5
<b>2</b>	<b>API DESCRIPTION .....</b>	<b>6</b>
2.1	BASE URL.....	6
2.2	SECURITY HEAER .....	6
2.3	CONFIGURATION .....	6
2.3.1	<i>Call</i> .....	6
2.3.2	<i>Result</i> .....	6
2.4	LOGIN .....	7
2.4.1	<i>Call</i> .....	7
2.4.2	<i>Result</i> .....	7
2.5	LOGOUT .....	7
2.5.1	<i>Call</i> .....	7
2.5.2	<i>Result</i> .....	8
2.6	PUBLICATION LIST.....	8
2.6.1	<i>Call</i> .....	8
2.6.2	<i>Result</i> .....	8
2.7	CREATE ISSUE .....	9
2.7.1	<i>Call</i> .....	9
2.7.2	<i>Result</i> .....	9
2.8	ISSUE LIST .....	10
2.8.1	<i>Call</i> .....	10
2.8.2	<i>Result</i> .....	10
2.9	CREATE ISSUE VERSION.....	11
2.9.1	<i>Call</i> .....	11
2.9.2	<i>Result</i> .....	11
2.10	ISSUE VERSION LIST .....	11
2.10.1	<i>Call</i> .....	11
2.10.2	<i>Result</i> .....	12
2.11	CHANGE THE STATUS OF AN ISSUE VERSION .....	12
2.11.1	<i>Preconditions</i> .....	12
2.11.2	<i>Notes</i> .....	13
2.11.3	<i>Call</i> .....	13
2.11.4	<i>Result</i> .....	13
2.12	UPLOAD PACKAGE .....	14
2.12.1	<i>Preconditions</i> .....	14
2.12.2	<i>Call</i> .....	14
2.12.3	<i>Result</i> .....	14
2.13	ASSOCIATE PACKAGE WITH PLATFORMS OR DEVICE CLASSES .....	15
2.13.1	<i>Preconditions</i> .....	15
2.13.2	<i>Call</i> .....	15



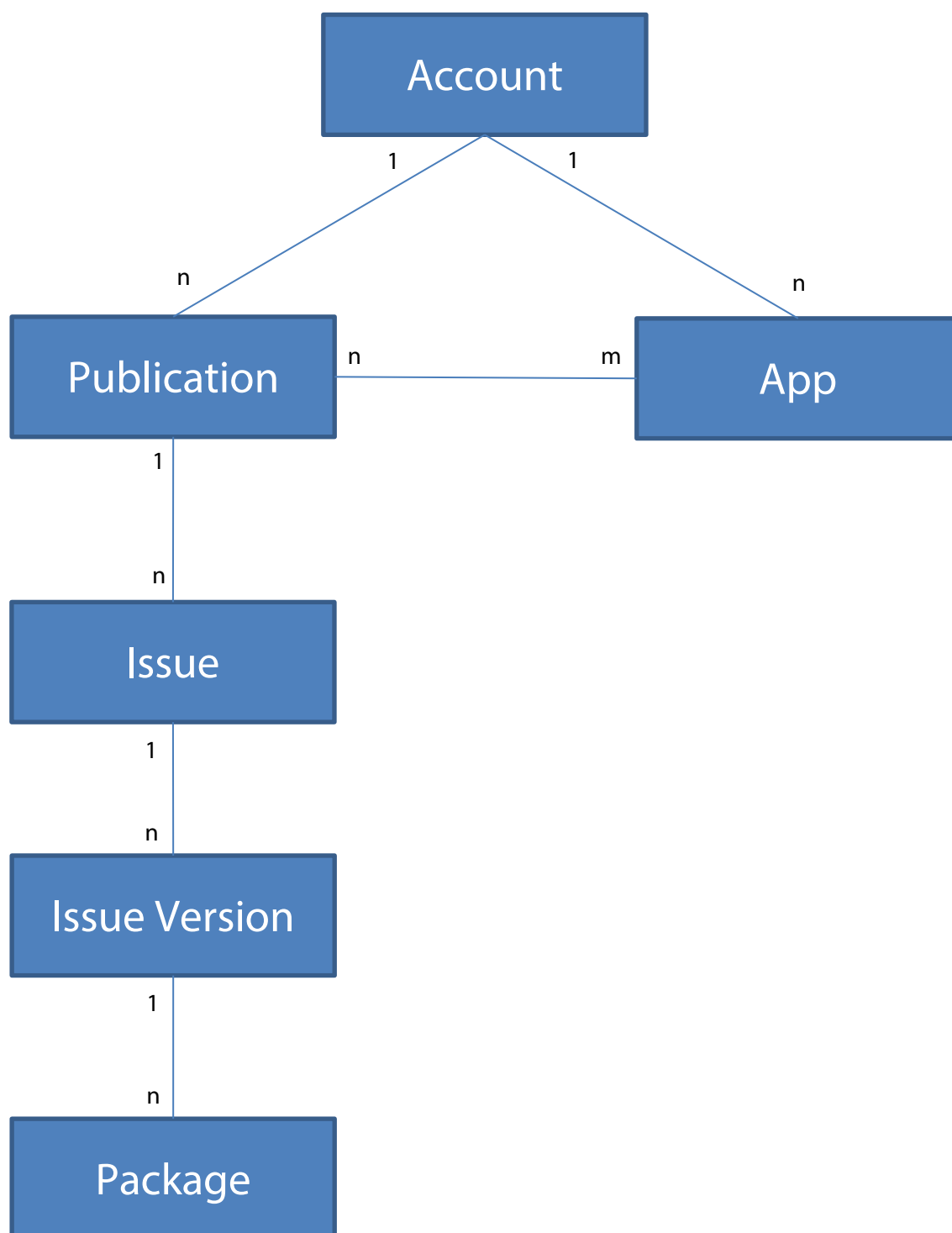
2.13.3	Result .....	16
2.14	DELETING A PACKAGE.....	17
2.14.1	Call .....	17
2.14.2	Result .....	17
2.15	DELETE ALL ISSUE VERSION OF AN ISSUE VERSION.....	17
2.15.1	Call .....	17
2.15.2	Result .....	18



## 1 INTRODUCTION

Before starting with specific APIs we give an overview of the basic data model and the complete workflow from login to upload of packages.

### 1.1 DATA MODEL





## 1.2 WORKFLOW

At first you have to log into the Purple manager. That generates a token used for all other API calls.

The second step is to retrieve an ID of a publication for which an issue will be edited. Therefore a list all publications for a specific account may be requested. From the result the publication name may be used to identify a specific publication.

Using this data (for example the ID of a publication and connected apps) either a new issue is created or the list of all issues of that publication is requested. Again the issue name may be used to identify a specific issue and the ID is used for further operations.

Now in a similar way an issue version is created or requested from the list of versions of this issue. The version ID is used in further APIs to set the state of a version (e.g. published, preview or coming soon).

Using the version ID new packages for this issue can be uploaded. In a first step the package is connected to an issue version. With separate API calls constraints (like platform or device type) may be added to the package. In order to have all parameter values for these constraints the global configuration - containing the lists of platforms and device types - has to be queried once.



## 2 API DESCRIPTION

This chapter describes all APIs to fulfill the package management workflow explained before.

### 2.1 BASE URL

Base URL for all API calls is

- Production system: <https://purplemanager.com/purple-manager-backend>
- Staging system: <https://staging.purplemanager.com/purple-manager-backend>

### 2.2 SECURITY HEAER

Each request to the backend must have a request header named "xr" and an addition query parameter (even if it's a POST request) names "xr", both containing the same random number (integer).

### 2.3 CONFIGURATION

For later usage during upload of packages the global configuration needs to be queried. It returns all platforms and device classes which need to be passed using their identifiers.

#### 2.3.1 CALL

**Path:** */configuration*

**Method:** GET

**Parameter:** none

#### 2.3.2 RESULT

Result is a JSON object with the following relevant entries

Field	Type	Description
platforms	Array	All known platforms as JSON objects containing an id and name field
deviceClasses	Array	All known device classes as JSON objects containing an id and name field and an optional field "subclasses" containing sub-device classes.



## 2.4 LOGIN

After successful login a session token is returned, which is used for each subsequent API call.

### 2.4.1 Call

**Path:** */auth/login*

**Method:** POST

**Parameter:**

Parameter name	Type	Description
email	String	Email of a valid Purple Manager account
password	String	Password of a valid Purple Manager account

### 2.4.2 RESULT

Result is a JSON object from which only the property „sessionID“ is used for further API usage.

Field	Type	Description
sessionID	String	Session token used for subsequent API calls

## 2.5 LOGOUT

Logout invalidated the current session.

### 2.5.1 Call

**Path:** */auth/logout*

**Method:** POST

**Parameter:**



Parameter name	Type	Description
sessionID	String	Session token that is invalidated

### 2.5.2 RESULT

No result data.

## 2.6 PUBLICATION LIST

A publication list is associated with one or more accounts (teams) the logged in user is associated with. Using this list a publication may be selected by name (or other criteria).

### 2.6.1 CALL

**Path:** */publication/list*

**Method:** GET

**Parameter:**

Parameter name	Type	Description
sessionID	String	Valid session token created by <a href="#">Login</a> .

### 2.6.2 RESULT

The result is a JSON array containing all publications accessible for the logged in user. Each publication is a JSON object containing the following relevant properties:

Field	Type	Description
id	String	Publication id used for subsequent API calls
name	String	Name of the publication





## 2.7 CREATE ISSUE

In case the issue for which the package shall be uploaded does not exist yet, it may be created using this API call.

### 2.7.1 CALL

**Path:** */issue*

**Method:** POST

**Parameter:**

Parameter name	Type	Description
sessionID	String	Valid session token created by <a href="#">Login</a> .
publicationId	String	Id of the publication returned by <a href="#">Publication list</a> .
name	String	Name of the new issue.
description	String	A description of the new issue (optional).
published	yyyy-MM-dd	Date of publication of the new issue (may be in the past or future).
file	File	issue thumbnail file (jpg or png)
issuelid	String	Set to "-1"

### 2.7.2 RESULT

Id of the new issue



## 2.8 ISSUE LIST

Querying an issue list of a publication helps to select an issue by name.

### 2.8.1 CALL

**Path:** */publication/listissues*

**Method:** GET

**Parameter:**

Parameter name	Type	Description
sessionID	String	Valid session token created by <a href="#">Login</a> .
publicationId	String	Id of a publication returned by <a href="#">Publication list</a> .

### 2.8.2 RESULT

The result is a JSON array containing all issue of the given publication. Each issue is a JSON object with the following relevant properties:

Field	Type	Description
id	String	Issue id used for subsequent API calls.
name	String	Name of the issue



## 2.9 CREATE ISSUE VERSION

Shall a package be uploaded into a new version of an issue, it has to be created beforehand. Every issue has a first version (number 1) created automatically when the issue is created. Besides the “full version” of an issue there may be “customer preview versions” used to present part of an issue to the customer for free. For these “customer preview versions” the first version has to be created by this API call explicitly.

### 2.9.1 CALL

**Path:** */version*

**Method:** POST

**Parameter:**

Parameter name	Type	Description
sessionID	String	Valid session token created by <a href="#">Login</a> .
issuelid	String	Id of an issue returned by <a href="#">Issue list</a> or by <a href="#">Create issue</a> .
customerPreview	boolean	Optional, if true a new „customer preview version“ is created, otherwise a full version is created.

### 2.9.2 RESULT

The result is the id of the new issue version.

## 2.10 ISSUE VERSION LIST

Having the list of all versions of an issue one may pick a version by its version number or the status (published, preview or “coming soon”) and use the id of this version in subsequent calls.

### 2.10.1 CALL

**Path:** */issue/listversions*

**Method:** GET

**Parameter:**



Parameter name	Type	Description
sessionID	String	Valid session token created by <a href="#">Login</a> .
issuelid	String	Id of an issue returned by <a href="#">Issue list</a> or by <a href="#">Create issue</a> .
customerPreview	boolean	Optional, if true „customer preview versions“ are returned, otherwise the list of full versions is returned.

### 2.10.2 RESULT

The result is a JSON array containing all versions of the given issue. Each version is a JSON object with the following relevant properties:

Field	Type	Description
id	String	Version id that may be used in subsequent API calls
number	Integer	Version number
active	Boolean	Flag indicating whether this version published in release app.
preview	Boolean	Flag indicating whether this version published in preview app.
comingSoon	Boolean	Flag indicating whether this version is “coming soon” (restricted to most recent version).

## 2.11 CHANGE THE STATUS OF AN ISSUE VERSION

In order to change the status of an issue version there are activating and deactivating API calls for each of the three status flags (published/active, preview, “coming soon”). These calls are distinguished by their path:

### 2.11.1 PRECONDITIONS

In order to change the status of an issue version to be published into the release app, there needs to be a bundle uploaded.



### 2.11.2 NOTES

If a version is published to release or preview app, the same status of another version is automatically deleted, so there is always exactly one issue version published to release or preview app. A version published to release app cannot be modified afterwards, a new version has to be created.

### 2.11.3 CALL

#### Paths:

Task	Pfad
Publish to release app	/version/activate
Unpublish from release app	/version/deactivate
Publish to preview app	/version/preview
Unpublish from preview app	/version/notpreview
Activate "coming soon"	/version/comingsoon
Deactivate "coming soon"	/version/notcomingsoon

#### Method: POST

#### Parameter:

Parameter name	Type	Description
sessionID	String	Valid session token created by <a href="#">Login</a> .
versionId	String	Id of an issue version for which the status is to be changed. It may be retrieved by a call to <a href="#">Issue version list</a> or by <a href="#">Create issue version</a> .

### 2.11.4 RESULT

No result data.



## 2.12 UPLOAD PACKAGE

Using this call a Purple DS archive (.pkar) is uploaded to an issue version. It may also be used to upload large asset files (e.g. videos) that are not bundled into the Purple DS archive.

### 2.12.1 PRECONDITIONS

The version must not be flagged as published to a release app.

### 2.12.2 CALL

**Path:** */package/upload*

**Method:** POST

**Parameter:**

Parameter name	Type	Description
sessionID	String	Valid session token created by <a href="#">Login</a> .
versionId	String	Id of an issue version for which the status is to be changed. It may be retrieved by a call to <a href="#">Issue version list</a> or by <a href="#">Create issue version</a> .
file	File	File to be uploaded

### 2.12.3 RESULT

Result is the id of the uploaded package.



## 2.13 ASSOCIATE PACKAGE WITH PLATFORMS OR DEVICE CLASSES

After uploading a file it is not yet associated with a platform or device class. This has to be done by a subsequent API call. If a packages is not associated with platforms and/or device classes it is valid for all platforms and/or device classes.

### 2.13.1 PRECONDITIONS

The version must not been flagged as published to a release app.

### 2.13.2 CALL

**Path:** */package/editRelations*

**Method:** POST

**Parameter:**

These parameters need to be passed within the URL:

Parameter name	Type	Description
sessionID	String	Valid session token created by <a href="#">Login</a> .
packageId	String	Id of a package uploaded beforehand using <a href="#">Upload package</a> .

**Body:**

For defining associations a request body containing a JSON object needs to be sent with the request. It has the following properties:

Field	Type	Description
t	String	Always "par"
deviceClasses	Array	An array of JSON object containing the device classes.
platforms	Array	An array of JSON objects containing the platforms
tags	String	A comma separated string of tags, may be an empty string.



### 2.13.2.1 DEVICE CLASSES OBJECT

Objects within the deviceClasses array containing the following properties:

Feld	Type	Description
t	String	Must contain "pdc"
deviceClassId	String	Id of the device class that shall be selected or unselected for the package. Use the <a href="#">Configuration</a> call to query available device classes once.
deviceClassName	String	Name of the device class that shall be selected or unselected for the package. Use the <a href="#">Configuration</a> call to query available device classes once.
selected	Boolean	Indicates the change to the device class association, i.e. selects or unselects the device class for the package

### 2.13.2.2 PKATFORMS OBJECT

Objects within the platforms array containing the following properties:

Feld	Type	Description
t	String	Must contain "pap"
platformId	String	Id of the platform that shall be selected or unselected for the package. Use the <a href="#">Configuration</a> call to query available platforms once.
platformName	String	Name of the platform that shall be selected or unselected for the package. Use the <a href="#">Configuration</a> call to query available platforms once.
selected	Boolean	Indicates the change to the platform association, i.e. selects or unselects the platform for the package

### 2.13.3 RESULT

No result data.





## 2.14 DELETING A PACKAGE

Using this call a package is deleted.

### 2.14.1 CALL

**Path:** */package/delete*

**Method:** POST

**Parameter:**

Parameter name	Type	Description
sessionID	String	Valid session token created by <a href="#">Login</a> .
packageId	String	Id of a package uploaded beforehand using <a href="#">Upload package</a> .

### 2.14.2 RESULT

No result data.

## 2.15 DELETE ALL ISSUE VERSION OF AN ISSUE VERSION

Using this call all packages of an issue version are deleted.

### 2.15.1 CALL

**Path:** */package/deleteAll*

**Method:** POST

**Parameter:**

Parameter name	Type	Description
sessionID	String	Valid session token created by <a href="#">Login</a> .



versionId	String	Id of an issue version for which the status is to be changed. It may be retrieved by a call to <a href="#">Issue version list</a> or by <a href="#">Create issue version</a> .
packageType	String	Indicates which type of files shall be deleted. Possible values: "content_bundle" for Purple DS archives and "asset" for all other assets.

---

### 2.15.2 RESULT

No result data.