

Deem Offers API Documentation

Introduction

[Overview](#)

[V3 Authentication](#)

[V1 Authentication](#)

[Errors](#)

Getting Started

[Quick Start](#)

V3 Core Calls

[Get Deals](#)

V1 Core Calls

[Get Deals\(Deprecated\)](#)

[Execute Purchases →](#)

V1 Additional Calls

[Ad Units](#)

[Deals](#)

[Purchases](#)

[Regions](#)

[Region Deals](#)

[V3 Region Deals](#)

[Users](#)

[User Access Tokens](#)

[User Card Links](#)

[User Coupons](#)

[User Credit Cards](#)

[User Purchases](#)

[User Saved Deals](#)

[User Shipping Addresses](#)

[User Subscriptions](#)

[Targeted Emails](#)

[Stats](#)

V3 Additional Calls

[Reseller Purchases](#)

[Reseller Coupons](#)

Execute Purchases (Create a Full Purchase)

You can use the all-in-one Execute Purchases call to create a purchase, create a new user, and store a purchase card.

This single operation will create a new user if the user doesn't already exist, and execute a single purchase transaction. It will also associate the purchase card information with the user. It doesn't matter if a user or purchase card resource already exists in the system. The backend software is intelligent enough to match existing resources to fulfill the purchase properly.

↔ Request

```
POST /v1/purchases/full.json?api_key={api_key}
```

↔ Request Parameters

You provide the required and optional parameters as a JSON file. The required fields are: * **purchase parameters** (deal_id, quantity) * **user parameters** (user id or user resource) * **payment card parameters** (credit card id or credit card resource) * **shipping address parameters** (shipping address id or shipping address resource) if the deal needs to be shipped

Purchase parameters

deal_id

String ID of the deal for which you are purchasing inventory

quantity

Integer Number of deals to be purchased

User parameters

Provide a stored ID or a complete user resource with the associated user fields (email, password, and password_confirmation).

User id

user_id

String ID of the user making the purchase

User resource

email

String Email address of the user making the purchase

pass

String Password of user making the purchase

pass_confirmation

String Exact duplicate of user password

first_name

Optional String First name of user

last_name

Optional String Last name of user

Payment card parameters

Provide a stored ID or a complete payment card resource with the associated payment card fields (number, verification_value, month, and year).

Payment card id

credit_card_id

String ID of the stored payment card used for purchasing

Payment card resource

number

String Payment card number used for purchasing

verification_value

String Verification value (CVV) for the payment card when a payment card number is provided

month

String Two-digit month for the payment card expiration date when a payment card number is provided

year

String Four-digit year for the payment card expiration date when a payment card number is provided

zip_code

String The zip code for the billing address for the payment card when a payment card number is provided.

Shipping address parameters

Provide an existing ID or a complete shipping address resource with the associated shipping address fields (address_one, postal_code, city, state, and country). This is required only for deals with a “shipped” fulfillment type.

Shipping address id**shipping_address_id**

String ID of the stored shipping address used for purchasing

Shipping address resource**name**

String Name associated with this shipping address (Home, Work, and so on)

address_one

String First line of the shipping address

address_two

Optional String Second line of the shipping address

postal_code

String Postal/Zip code of the shipping address

city

String City name of the shipping address

state

String Two-character state abbreviation of the shipping address

country

String Two-character country abbreviation of shipping address

↔ Request Example

New user account, new payment card, and a shipping address:

```
POST /v1/purchases/full.json?api_key={api_key}
```

```
{
  "deal_id": "a-50-for-100-worth-of-dining-1",
  "quantity": 1,
  "user": {
    "email": "help@developer.offerengine.com",
    "password": "pass123",
    "password_confirmation": "pass123",
    "first_name": "Hank",
    "last_name": "Rearden"
  },
  "credit_card": {
    "number": "424242424242424",
    "verification_value": 123,
    "month": 1,
    "year": 2014,
    "zip_code": "90210"
  },
  "shipping_address": {
    "address_one": "501 Howard St",
    "address_two": "#2000",
    "postal_code": "94105",
    "city": "San Francisco",
    "state": "CA",
    "country": "US"
  }
}
```

Existing user account, stored payment card, and no shipping address:

```
POST /v1/purchases/full.json?api_key={api_key}
```

```
{  
  "deal_id": "a-50-for-100-worth-of-dining-1",  
  "quantity": 1,  
  "user_id": "ABC123",  
  "credit_card_id": "2342f8073e"  
}
```

↔ Response

The request response includes an HTTP status of the request along with the associated response data provided as a JSON file.

↔ Response Attributes

The response JSON file includes an overall purchase status and purchase resources for each purchase in the request.

Status attributes

status

String State of this purchase (success or error)

Purchases attributes

The response includes a purchase attribute, deal attributes, a user attribute, a payment card attribute, coupon attributes, and errors for each attribute that failed during the request.

Purchase attributes

id

String Unique identifier for this purchase

purchase_id

String Unique identifier for this purchase

payment_state

String Current state of the payment process for this purchase

fulfillment_state

String Current state of the fulfillment process for this purchase

created_at

String Date and time that the purchase record was created

deal_id

String Unique identifier for the deal purchased

deal_type

String Type of the deal purchased

price

Integer Price of the deal in cents

amount

Integer Total amount paid for the purchase in cents

number_bought

Integer Quantity of the deals bought

User attribute

user_id

String Unique identifier for the user making the purchase

Payment card attribute

credit_card_id

String Unique identifier for the payment card used to make this purchase

Deal attributes

id

String Unique identifier for the purchased deal

deal_id

String Unique identifier for the purchased deal

title

String Title of the purchased deal

type

String Type of the purchased deal

image_url
String Relative URL for an image associated with the purchased deal

image_url_abs
String Absolute URL for an image associated with the purchased deal

Coupon attributes

id
String Unique identifier for this purchase voucher

slug
String Unique identifier for this purchase voucher

deal_id
String Unique identifier for purchased deal

deal_type
String Type of purchased deal ("daily-deal", "on-going", "card-linked")

barcode
String Unique barcode used for purchase redemption

redemption_code
String Unique code used for purchase redemption

expires_at
String Date and time of the purchase voucher expiration

created_at
String Date and time that the purchase was made

state
String State of the purchase ("valid", "cancelled", "expired", "redeemed", "reissued")

redeemed_at
Optional String Date and time when the purchase voucher was redeemed

qrcode_url
String URL for the qrcode image associated with the coupon

qrcode_base64
String Base64 encoded qrcode image associated with the coupon

↔ Response Example

Status: 200 OK

```
{
  "status": "success",
  "purchase": {
    "id": "34063ec2",
    "purchase_id": "34063ec2",
    "deal_type": "daily-deal",
    "deal_id": "la-1024-wsc-hair-studio-01-2",
    "price": 3900,
    "amount": 0,
    "number_bought": 0,
    "payment_state": "charged",
    "fulfillment_state": "fulfilled",
    "created_at": "Fri Dec 16 19:11:15 UTC 2011",
    "user_id": "ABC123",
    "credit_card_id": "2342f8073e",
    "deal": {
      "id": "a-50-for-100-worth-of-dining-1",
      "deal_id": "a-50-for-100-worth-of-dining-1",
      "title": "$50 for $100 worth of Dining",
      "type": "daily-deal",
      "image_url": "deals/generic_medium.gif",
      "image_url_abs": "http://d2x9dz1etb1m98.cloudfront.net/deals/generic_medium.gif"
    },
  },
  "coupons": [
    {
      "id": "a21c5452d8",
      "expires_at": null,
      "created_at": "Fri Dec 16 19:14:55 UTC 2011",
      "qr_code_url": "http://domain/coupon/qr_code/ab231c9e",
      "qr_code": "apaosidnqw4tdsdpqi45jf091835ukdfq3-40695.kfmdg-q09i6m'ldfi8nqk6",
      "state": "valid",
      "redemption_code": null,
      "redeemed_at": null,
      "print_url": "http://domain/coupons/a21c5452d8/print"
    }
  ]
}
```

```
}
]
}
}
```

↔ Error Responses

Responses are provided for each failed portion of the purchase request. If the user, payment card, or shipping address sections fail, the purchase does not occur. You don't have to worry that purchases might be made with invalid data.

409

error_type = "already_charging"
Existing purchase still processing

422

error_type = "cannot_fulfill"
Not able to fulfill the purchase

error_type = "declined"
Purchase declined

error_type = "validations_failed"
Validation failed, which can be five failure types ("missing_field", "missing", "invalid", "not_owner", and "not_found"). Each type also has the associated resource that caused the error ("user", "credit_card", "purchase", "shipping address") and may also have a field narrowing down the specific attribute if applicable.

↔ Error Response Example

Validation errors for purchase only

Status: 422 Unprocessable Entity

```
{
  "status": "error",
  "error_type": "validation_failed",
  "error_msg": "validation failed",
  "user_id": "ABC123",
  "credit_card_id": "2342f8073e",
  "errors": [
    {
      "type": "invalid",
      "message": "cannot purchase that many",
      "field": "quantity",
      "resource": "purchase"
    }
  ]
}
```

Validation errors for user, payment card, purchase, and shipping address

Status: 422 Unprocessable Entity

```
{
  "status": "error",
  "error_type": "validation_failed",
  "error_msg": "validation failed",
  "errors": [
    {
      "type": "missing_field",
      "message": "is missing",
      "field": "email",
      "resource": "user"
    },
    {
      "type": "missing",
      "message": "missing credit card info",
      "resource": "credit_card"
    },
    {
      "type": "invalid",
      "message": "cannot purchase that many",
      "field": "quantity"
    }
  ]
}
```

```
field : quantity ,
"resource": "purchase"
},
{
"type": "not_found",
"message": "doesn not exist",
"resource": "shipping_address"
}
]
}
```

deem

[Products](#)

[Discover](#)

[Developers](#)

[About Us](#)

[Terms](#)

[Privacy](#)

[Feedback](#)

© 1999-2014 Deem, Inc. All rights reserved.