



# Cheetah Loyalty Point of Sale (POS) APIs

## *Development Guide*

June 2019



**Cheetah Digital, Inc., 72 W Adams St 8th floor, Chicago, IL 60603**

**Copyright © 2019 Cheetah Digital, Inc.**

**All rights reserved.**

Printed in the United States of America

No part of this publication may be stored in a retrieval system, transmitted, or reproduced in any way, including but not limited to photocopy, photographic, magnetic, or other record, without the prior agreement and written permission of Cheetah Digital, Inc.

Cheetah Digital, the Cheetah Digital logo, and other Cheetah names referenced herein are trademarks of Cheetah Digital, Inc., and may be registered in certain jurisdictions.

Other product names, designations, logos, and symbols may be trademarks or registered trademarks of their respective owners.

**PRODUCT MODULES AND OPTIONS.** This guide contains descriptions of modules that are optional and for which you may not have purchased a license. As a result, your software implementation may differ from descriptions in this guide. To find out more about the modules your organization has purchased, see your corporate purchasing agent or your Cheetah Digital sales representative.

**U.S. GOVERNMENT RESTRICTED RIGHTS.** Programs, Ancillary Programs and Documentation, delivered subject to the Department of Defense Federal Acquisition Regulation Supplement, are “commercial computer software” as set forth in DFARS 227.7202, Commercial Computer Software and Commercial Computer Software Documentation, and as such, any use, duplication and disclosure of the Programs, Ancillary Programs and Documentation shall be subject to the restrictions contained in the applicable Cheetah Digital license agreement. All other use, duplication and disclosure of the Programs, Ancillary Programs and Documentation by the U.S. Government shall be subject to the applicable Cheetah Digital license agreement and the restrictions contained in subsection (c) of FAR 52.227-19, Commercial Computer Software - Restricted Rights (June 1987), or FAR 52.227-14, Rights in Data—General, including Alternate III (June 1987), as applicable. Contractor/licensor is Cheetah Digital, Inc., 72 W Adams St 8th floor, Chicago, IL 60603.

**PROPRIETARY INFORMATION NOTICE**

Stellar Loyalty, Inc. considers information included in this documentation and all Stellar Loyalty documents to be Confidential Information. Your access to and use of this Confidential Information are subject to the terms and conditions of: (1) the applicable Stellar Loyalty software license agreement, which has been executed and with which you agree to comply; and (2) the proprietary and restricted rights notices included in this documentation.



# Contents

<b>Introduction</b>	<b>3</b>
<b>Audience</b>	<b>3</b>
<b>Use Cases</b>	<b>3</b>
<b>API Details</b>	<b>4</b>
Assign Member (aka the “Get Offers Transaction”)	4
Unassign Member	7
Finalize Transaction (aka “Submit Transaction”)	9
Get Member Summary	12
Activate Card	13
Replace Card	17



## Introduction

This document describes the Cheetah Loyalty APIs for third party point-of-sale (POS), such as MICROS RES, RICS, and Squirrel POS, to communicate with Cheetah Loyalty's servers.

## Audience

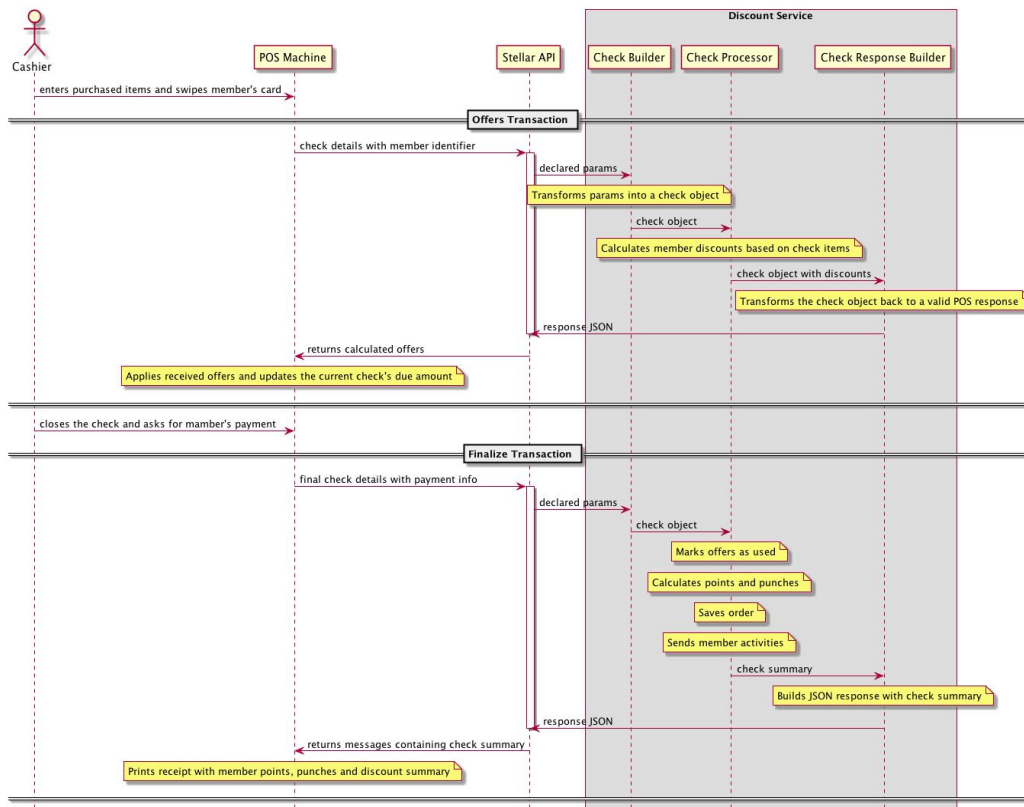
The intended audience is developers integrating 3rd party point-of-sale systems with Cheetah Loyalty. These developers require technical knowledge for invoking and debugging API calls using OAuth, REST, and JSON.

## Use Cases

The Cheetah Loyalty POS APIs supports these use cases:

1. Assign a member to a transaction
2. Unassign a member from a transaction
3. Submit transaction to Cheetah Loyalty
4. Get member summary / balance
5. Activate a physical card
6. Replace a physical card

The sequence diagram below describes the flow of API calls between the POS system and Cheetah Loyalty's servers.





## API Details

This section describes the API endpoints, with their corresponding parameter and response structures.

These endpoints can be accessed either via Cheetah Loyalty's Member API or Cheetah Loyalty's Program API.

### 1. Program API

Base URL: /program/api/pos/stellar

### 2. Member API

Base URL: /api/pos/stellar

- The owner of the `access_token` is set as the primary account in each API request

## Assign Member (aka the "Get Offers Transaction")

Use this API to validate a member, apply qualifying discounts earned from previous transactions. The POS is responsible for applying the discounts to the open transaction.

This API utilizes Cheetah Loyalty's Discount Service to calculate the available discounts. The POS submits the transaction line items, then this api validates the member account and returns all possible rewards. The POS should call this API every time the transaction details changes, such as after adding or removing of items, comps, or other discounts on the transaction.

## API

**PUT** <base\_url>?**access\_token**=<access\_token>

Keys in **bold** are required.

```
{
  "status": String,
  "timestamp": Timestamp,
  "store_id": String,
  "terminal_id": String,
  "transaction_id": String,
  "operator_id": String,
  "accounts": [
    {
      "identifier": String,
      "identifier_type": String
    }
  ],
  "items": [
    {
```



```
        "level": Integer,
        "line_number": Integer,
        "parent_line_number": Integer,
        "name": String,
        "description": String,
        "sku": String,
        "categories": Array,
        "quantity": Decimal,
        "unit_price": Decimal,
        "discountable": Boolean
    }
],
"coupons": [
    {
        "type": String,
        "pos_code": String,
        "stellar_code": String,
        "name": String,
        "quantity": Float,
        "value": BigDecimal,
        "items": [
            {
                "line_number": Integer
            }
        ]
    }
],
"summary": {
    "subtotal": Decimal,
    "discount": Decimal,
    "service_charge": Decimal,
    "tax": Decimal,
    "payment": Decimal
}
```

## Example

### Request:

```
{
    "status": "assign_member",
    "timestamp": "2017-09-27T00:00:00Z",
    "store_id": "1",
    "terminal_id": "1",
    "transaction_id": "1",
    "operator_id": "1",
    "accounts": [
        {
            "identifier": "6011888052563949",
```



```
        "identifier_type": "card_id"
      }
    ],
    "items": [
      {
        "level": 0,
        "line_number": 1,
        "parent_line_number": 0,
        "name": "Fries",
        "description": "Cheese Fries",
        "sku": "401301",
        "categories": [
          "102"
        ],
        "quantity": 1,
        "unit_price": 5.25,
        "discountable": true
      }
    ],
    "coupons": [],
    "summary": {
      "subtotal": 6.25,
      "discount": 0,
      "service_charge": 0,
      "tax": 1,
      "payment": 0
    }
  }
}
```

**Response:**

```
{
  "messages": {
    "display": [
      {
        "message": "Adding guest - Michael Jordan.\n\nApplied the following rewards:\nFree Fries"
      }
    ],
    "receipt": []
  },
  "accounts": [
    {
      "identifier": "michael@jordan.com",
      "valid": true,
      "name": "Michael Jordan",
      "member_id": "S-000105990"
    }
  ]
}
```



```
],
"add_coupons": [
  {
    "type": "stellar",
    "pos_code": "904",
    "stellar_code": "OfferResponse-1",
    "name": "Free Fries",
    "quantity": 1,
    "value": 5.25,
    "items": [
      {
        "line_number": "1"
      }
    ]
  }
]
```

## Unassign Member

Use this API to unassign a previously assigned member from an open transaction. For example, after swiping a wrong card or entering an incorrect phone number on the POS. This API supports instructs Cheetah Loyalty to remove the member from the current transaction and make the member's discounts available for another transaction.

### API

**PUT** <base\_url>?access\_token=<access\_token>

Keys in **bold** are required.

```
{
  "status": "unassign_member",
  "timestamp": Timestamp,
  "store_id": String,
  "terminal_id": String,
  "transaction_id": String,
  "operator_id": String,

  "accounts": [
    {
      "identifier": String,
      "identifier_type": String
    }
  ]
}
```

### Example

Request:





```
{
  "status": "unassign_member",
  "timestamp": "2017-09-27T00:00:00Z",
  "store_id": "1",
  "terminal_id": "1",
  "transaction_id": "1",
  "operator_id": "1",
  "accounts": [
    {
      "identifier": "6011888052563949",
      "identifier_type": "card_id"
    }
  ]
}
```

**Response:**

```
{
  "messages": {
    "display": [
      {
        "message": "Unassigned guest - Michael Jordan."
      }
    ],
    "receipt": []
  },
  "accounts": [
    {
      "identifier": "6011888052563949",
      "valid": true,
      "name": "Michael Jordan",
      "member_id": "S-000105990"
    }
  ],
  "remove_coupons": [
    {
      "type": "stellar",
      "pos_code": "904",
      "stellar_code": "OfferResponse-1",
      "name": "Free Fries",
      "quantity": 1,
      "value": 5.25,
      "items": [
        {
          "line_number": "1"
        }
      ]
    }
  ]
}
```



```
    ]
  }
```

## Finalize Transaction (aka “Submit Transaction”)

After all items are inputted/scanned into the transaction, use this API to finalize/close the transaction with Cheetah Loyalty. This API logs the transaction with the previously assigned member, and performs all possible calculations for the member to accrue points, earn discounts, trigger emails, and so on. This API marks discounts that were used in the transaction as ‘Used’. This API responds with messages that the POS could display or print on a receipt.

## Shell account

In the case when the member does not exist in the Cheetah Loyalty database based on identifier, Cheetah Loyalty will create a shell account with profile attributes specified in the POST request. After creating a shell account, Cheetah Loyalty assigns the transaction to the new shell account. In the case where the profile attributes are in conflict with an existing member account, the shell account will be created without the attributes, so that the card issued continues to work. In the case when the primary identifier is in conflict (e.g. card ID where used or phone number or email - just the primary) then the account should not be created. However, if the conflict is in the secondary attribute, for example, the account is identified by phone number in the identifier field, and email is passed in the profile section, and email has an issue but phone number is valid, Cheetah Loyalty creates the new account but leave the email not updated. However, in the same example, if the phone number already exists, then Cheetah Loyalty will not create the new account

## Update profile

This API will try to update profile attribute for existing member based on identifier. In case profile update fails then transaction will be assigned to member without updating profile.

## API

**POST** <base\_url>?access\_token=<access\_token>

Keys in **bold** are required.

```
{
  "status": String,
  "timestamp": Timestamp,
  "store_id": String,
  "terminal_id": String,
  "transaction_id": String,
  "operator_id": String,
  "accounts": [
    {
      "identifier": String,
      "identifier_type": String
```



```
        "profile": {
            "member_attribute_name": "<value>",
            "member_attribute_name": "<value>",
            "member_attribute_name": "<value>"
        }
    },
    "items": [
        {
            "level": Integer,
            "line_number": Integer,
            "parent_line_number": Integer,
            "name": String,
            "description": String,
            "sku": String,
            "categories": Array,
            "quantity": Decimal,
            "unit_price": Decimal,
            "discountable": Boolean
        }
    ],
    "coupons": [
        {
            "type": String,
            "pos_code": String,
            "stellar_code": String,
            "name": String,
            "quantity": Float,
            "value": BigDecimal,
            "items": [
                {
                    "line_number": Integer
                }
            ]
        }
    ],
    "summary": {
        "subtotal": Decimal,
        "discount": Decimal,
        "service_charge": Decimal,
        "tax": Decimal,
        "payment": Decimal
    }
}
```

## Example

Request:



```
{
  "status": "submit_transaction",
  "timestamp": "2017-09-27T00:00:00Z",
  "store_id": "1",
  "terminal_id": "1",
  "transaction_id": "1",
  "operator_id": "1",
  "accounts": [
    {
      "identifier": "6011888052563949",
      "identifier_type": "card_id"
    }
  ],
  "items": [
    {
      "level": 0,
      "line_number": 1,
      "parent_line_number": 0,
      "name": "Fries",
      "description": "Cheese Fries",
      "sku": "401301",
      "categories": [
        "102"
      ],
      "quantity": 1,
      "unit_price": 5.25,
      "discountable": true
    }
  ],
  "coupons": [
    {
      "type": "stellar",
      "pos_code": "904",
      "stellar_code": "OfferResponse-1",
      "name": "Free Fries",
      "quantity": 1,
      "value": 5.25,
      "items": [
        {
          "line_number": "1"
        }
      ]
    }
  ],
  "summary": {
    "subtotal": 1.00,
    "discount": 5.25,
  }
}
```



```
        "service_charge": 0.00,  
        "tax": 1.00,  
        "payment": 0.00  
    }  
}
```

### Response:

```
{  
  "messages": {  
    "display": [  
      {  
        "message": "Thank you and come again soon!"  
      }  
    ],  
    "receipt": []  
  },  
  "accounts": [  
    {  
      "identifier": "michael@jordan.com",  
      "valid": true,  
      "name": "Michael Jordan",  
      "member_id": "S-000105990"  
    }  
  ]  
}
```

## Get Member Summary

This API allows the cashier to get a member's current balance and available discounts.

### API

**PUT** <base\_url>?access\_token=<access\_token>

Keys in **bold** are required.

```
{  
  "status": "member_summary",  
  "timestamp": Timestamp,  
  "store_id": String,  
  "terminal_id": String,  
  "transaction_id": String,  
  "operator_id": String,  
  "accounts": [  
    {  
      "identifier": String,  
      "identifier_type": String  
    }  
  ]  
}
```



```
    }  
  ]  
}
```

## Example

### Request:

```
{  
  "status": "member_summary",  
  "timestamp": "2017-09-27T00:00:00Z",  
  "store_id": "1",  
  "terminal_id": "1",  
  "transaction_id": "1",  
  "operator_id": "1",  
  "accounts": [  
    {  
      "identifier": "370615108474724",  
      "identifier_type": "card_id"  
    }  
  ]  
}
```

### Response:

```
{  
  "messages": {  
    "display": [  
      {  
        "message": "Name: Michael Jordan\nMember ID: S-000105990\nBalance: 100 points\nRewards:\n(2) Free Fries\nFree Pizza\n(2) $5-Gift Cards with $10.00-value"  
      }  
    ],  
    "receipt": []  
  },  
  "accounts": [  
    {  
      "identifier": "370615108474724",  
      "valid": true,  
      "name": "Michael Jordan",  
      "member_id": "S-000105990"  
    }  
  ]  
}
```

## Activate Card

This API allows the cashier to activate a new loyalty card. In this process, Cheetah Loyalty creates a shell account (visitor: true) that is eligible to earn loyalty points and other rewards on



every transaction. After this process, the customer is expected to register online and update this shell account to complete his registration process. Only from that point that the customer will be able to use and redeem any of his previously earned points and rewards.

Card activation could be done in two ways; with or without any pending transaction.

### API Without Transaction

This supports the scenario where the customer only wants to activate his card without buying anything from the store.

```
PUT <base_url>?access_token=<access_token>
```

Keys in **bold** are required.

```
{
  "status": "activate_card",
  "timestamp": Timestamp,
  "store_id": String,
  "terminal_id": String,
  "transaction_id": String,
  "operator_id": String,

  "accounts": [
    {
      "identifier": String,
      "identifier_type": String
    }
  ]
}
```

### Example Without Transaction

Request:

```
{
  "status": "activate_card",
  "timestamp": "2017-09-27T00:00:00Z",
  "store_id": "1",
  "terminal_id": "1",
  "transaction_id": "1",
  "operator_id": "1",
  "accounts": [
    {
      "identifier": "370615108474724",
      "identifier_type": "card_id"
    }
  ]
}
```

**Response:**

```
{
  "messages": {
    "display": [
      {
        "message": "Card #370615108474724 activated."
      }
    ],
    "receipt": []
  },
  "accounts": [
    {
      "identifier": "370615108474724",
      "valid": true
    }
  ]
}
```

**API With a Transaction**

Use this API to link a pending transaction to the shell account that Cheetah Loyalty will create upon closing the transaction.

**PUT** <base\_url>?access\_token=<access\_token>

Keys in **bold** are required.

```
{
  "status": String,
  "timestamp": Timestamp,
  "store_id": String,
  "terminal_id": String,
  "transaction_id": String,
  "operator_id": String,
  "accounts": [
    {
      "identifier": String,
      "identifier_type": String
    }
  ],
  "items": [
    {
      "level": Integer,
      "line_number": Integer,
      "parent_line_number": Integer,
      "name": String,
      "description": String,
      "sku": String,

```





```
        "categories": Array,  
        "quantity": Decimal,  
        "unit_price": Decimal,  
        "discountable": Boolean  
    }  
],  
"coupons": Array,  
"summary": {  
    "subtotal": Decimal,  
    "discount": Decimal,  
    "service_charge": Decimal,  
    "tax": Decimal,  
    "payment": Decimal  
}  
}
```

### Example With a Transaction

Request:

```
{  
    "status": "assign_member",  
    "timestamp": "2017-09-27T00:00:00Z",  
    "store_id": "1",  
    "terminal_id": "1",  
    "transaction_id": "1",  
    "operator_id": "1",  
    "accounts": [  
        {  
            "identifier": "370615108474724",  
            "identifier_type": "card_id"  
        }  
    ],  
    "items": [  
        {  
            "level": 0,  
            "line_number": 1,  
            "parent_line_number": 0,  
            "name": "Fries",  
            "description": "Cheese Fries",  
            "sku": "401301",  
            "categories": [  
                "102"  
            ],  
            "quantity": 1,  
            "unit_price": 5.25,  
            "discountable": true  
        }  
    ],  
}
```



```
"coupons": [],
"summary": {
  "subtotal": 6.25,
  "discount": 0.00,
  "service_charge": 0.00,
  "tax": 1.00,
  "payment": 0.00
}
}
```

### Response:

```
{
  "messages": {
    "display": [
      {
        "message": "About to activate card #370615108474724.\n\nRemind guest to register online to get\n\ncredit for their purchase."
      }
    ],
    "receipt": []
  },
  "accounts": [
    {
      "identifier": "370615108474724",
      "valid": true
    }
  ]
}
```

## Replace Card

Use this API to change the card id for an existing member account, for example, when the member loses or damages her existing card. This API expects a member identifier to retrieve the existing member account and a new card id to assign to the member account.

### API

**PUT** <base\_url>?access\_token=<access\_token>

Keys in **bold** are required.

```
{
  "status": "replace_card",
  "timestamp": Timestamp,
  "store_id": String,
  "terminal_id": String,
  "transaction_id": String,
```



```
"operator_id": String,  
  
"accounts": [  
  {  
    "identifier": String,  
    "identifier_type": String  
  },  
  {  
    "identifier": String,  
    "identifier_type": String  
  }  
]  
}
```



## Example

### Request:

```
{
  "status": "replace_card",
  "timestamp": "2017-09-27T00:00:00Z",
  "store_id": "1",
  "terminal_id": "1",
  "transaction_id": "1",
  "operator_id": "1",
  "accounts": [
    {
      "identifier": "michael@jordan.com",
      "identifier_type": "email"
    },
    {
      "identifier": "6011888052563949",
      "identifier_type": "card_id"
    }
  ]
}
```

### Response:

```
{
  "messages": {
    "display": [
      {
        "message": "Card #6011888052563949 is now associated\\n to
guest - Michael Jordan."
      }
    ],
    "receipt": []
  },
  "accounts": [
    {
      "identifier": "michael@jordan.com",
      "valid": true,
      "name": "Michael Jordan",
      "member_id": "S-000105990"
    },
    {
      "identifier": "6011888052563949",
      "valid": true
    }
  ]
}
```