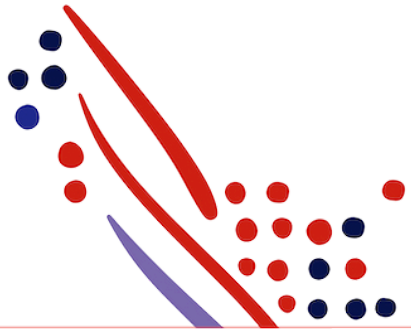
Guide

Data Collection APIs for ADP Workforce Manager (WFMgr)

Published on
Oct 15, 2024, 01:57 PM

Last modified
Mar 06, 2025, 08:36 PM

Always Designing for People™



ADP Copyright Information

ADP, the ADP logo, and Always Designing for People are trademarks of ADP, Inc.

Windows is a registered trademark of the Microsoft Corporation.

All other trademarks are the property of their respective owners.

Copyright © 2025 ADP, Inc. ADP Proprietary and Confidential - All Rights Reserved. These materials may not be reproduced in any format without the express written permission of ADP, Inc.

These materials may not be reproduced in any format without the express written permission of ADP, Inc. ADP provides this publication "as is" without warranty of any kind, either expressed or implied, including, but not limited to, the implied warranties of merchantability or fitness for a particular purpose. ADP is not responsible for any technical inaccurancies or typographical errors which may be contained in this publication. Changes are periodically made to the information herein, and such changes will be incorporated in new editions of this publication. ADP may make improvements and/or changes in the product and/or the programes described in this publication.

Published on
Oct 15, 2024, 01:57 PM

Published on
Mar 06, 2025, 08:36 PM

Table of Contents

Chapter 1

Data Collection API

- API Usage
- Application Scope
- Request Header Parameters
- Sequence of Interactions
- Data Dictionary
- Responses

Chapter 2

Data Collection Entries Web Service Method - Upload - POST

- Data Collection Entries Web Service Method - Upload - POST
- Action Code/ Entry Codes Overview
  - Action and Entry Code JSON Object Locations
  - Action Code/Entry Code Description Matrix
  - Action Code/Entry Code Not Supported for WorkforceNow Next Gen
- General Comments and Notes
- Request Header Parameters
- Multiple Punches Sample
- Scan/Punch
- Lunch Out
- Meal Return
- Clock In
- Clock Out
- Change Work/Transfer
- Data Collection Entries API Element Glossary
  - Events[]
  - dataCollectionEntries[]
  - workerDataCollectionEntries[]

Chapter 3

Data Collection Entries Web Service Method - Status - GET

- Request
- Request Header Parameters
- Possible Responses

201 - Succesful Import Status Response

207– Partial Failures

400 – Transaction Failure Response

404 – Resource Not Found Response

Chapter 4

Message Code Dictionary

Errors

Chapter 5

Frequently asked questions

Chapter 6

Known Issues and Limitations

Chapter 1

Data Collection API

Overview

The API is used to list the clock entries for a worker.
Below is a summary of the services documented in this API:

API Usage

Method	URI	Description
POST	/events/time/v1/clock.punch	Clock punch event instance
GET	/events/time/v1/clock.punch/meta	Retrieve event metadata

Application Scope

The following canonical needs to be added to your application scope to enable this use case:

/time/timeLaborManagement/timeEntryManagement/clockEntryManagement/clock.read

Request Header Parameters

Name	Description
associateoid *required **string** (header)	Associate OID of the requestor as provisioned by the ADP Registry Default value:G4073G9Z62SL2NFM
orgoid *required **string** (header)	Organization OID of the requestor as provisioned by the ADP Registry
SOR *required **string** (header)	SOR where the call is routed to - usuall injected by the API proxy, with few exceptions

SORContext **string** (header)	Context within an SOR that the call should be routed to
SORUri **string** (header)	Actual Uri within the SOR
realm *required **string** (header)	The identity management realm, e.g. ISI Default value:ISI
roleCode *required **string** (header)	The role the user is playing during the transaction. Possible values: employee,manager,practitioner,administrator,supervisor. The roleCode header should be passed in all calls. Available values :employee,manager,practitioner,administrator,supervisor Default value:employee
sm_serversessionid **string** (header)	SiteMinder session ID. Required for requests where conversation was initiated from a service consumer, external to ADP, and that consumer was authenticated by SMS. Example - sm_serversessionid: SoFpy8UMyhtXY6YsEumvXJbznFA=
sm_transactionid **string** (header)	SiteMinder transaction ID. Required for requests where conversation was initiated from a service consumer, external to ADP, and that consumer was authenticated by SMS. Example - sm_transactionid: 1aaf14ac-7083-54c0186f-4680c940-82b33ca3
sm_auth_level **string** (header)	SiteMinder autorization level. Required for requests where conversation was initiated from a service consumer, external to ADP, and that consumer was authenticated by SMS. Example - sm_auth_level: 10
sm_user **string** (header)	SiteMinder user ID. Required for requests where conversation was initiated from a service consumer, external to ADP, and that consumer was authenticated by SMS. Example - sm_user: JJohns@Abc
sm_userdn **string**	SiteMinder user LDAP distinguished name. Required for requests where conversation was initiated from a service consumer, external to ADP, and that consumer was authenticated by SMS. Example - sm_userdn: uid=JJohns@Abc,ou=Users,o=Abc

(header)	
Accept-Language **string** (header)	Accept-Language' ': ' (language | '*') [';' 'q' '='qvalue]. Specifies the language(s) that are acceptable for the response. The optional qvalue represents a quality level for acceptable languages. Default value:en-US
ConsumerAppOID **string** (header)	Identifier for the service consumer application. This header is required, if the actor (Service Consumer) is an application or the actor is internal to ADP
ADP-Act-As-OrgOID **string** (header)	Org OID for organization of the associate that is the actor in 'Acting As' as provisioned by the ADP Registry
ADP-Act-As-AssociateOID **string** (header)	Associate OID of the associate that is the actor in 'Acting As' as provisioned by the ADP Registry
ADP-On-Behalf-Of-OrgOID **string** (header)	Org OID for organization of the associate that is the actor in ' On Behalf Of' as provisioned by the ADP Registry
ADP-On-Behalf-Of-AssociateOID **string** (header)	Associate OID for the associate that is the actor in ' On Behalf Of' as provisioned by the ADP Registry
ADP-Acting-SessionID **string** (header)	Session ID for the acting session in 'Act-As' and act 'On-Behalf-Of' scenarios

X-Forwarded-For **string** (header)	Standard header for identifying the originating IP address of a client connecting to a web server through a proxy server Default value:en-US
ADP-Content-Capture-Mode **string** (header)	The ADP-Content-Capture-Mode header MUST be used to contain a list of content capture modes. Note: The term 'captured' indicates the the message content was stored for some period time at some interim location before reaching the destined SOR; the mode implies the reason that the content was captured (i.e. 'server-down', 'client-offline') and may also imply where it was captured. 'ADP-Content-Capture-Mode': 'mode' The mode field value MUST be limited to an element of the value domain: 'server-down', 'client-offline'.
Authorization **string** (header)	Provides the client's authorization token to access the resource representation at a URI
ADP-Context-ExpressionID **string** (header)	Geopolitical context, typically country, region, e.g. country=CA;subdivision=Quebec
ADP-Consumer-ApplicationID **string** (header)	Identifier of the application sending a request - not intended to be used as a correlation id for messages across systems participating in a collaboration

Sequence of Interactions

1. Your consumer application makes a request for **/events/time/v1/clock.punch/meta** to the ADP API endpoint.
2. The ADP API endpoint responds to your consumer application with the meta payload (see the table in the following Data Dictionary section).
3. Your consumer application processes the meta payload to validate data included in the payload of step 4 and prompts the user to fix any validation issues. Otherwise, the payload for the **/events/time/v1/clock.punch** request is composed.
4. Your consumer application makes a **/events/time/v1/clock.punch** request the ADP API endpoint.
5. The ADP API endpoint responds to the consumer application concerning the details of **/events/time/v1/clock.punch**.

Data Dictionary

Resources listed in the following table can be accessed in the ADP Workforce Now UI by selecting **People > Personal Information > Talent Profile > Licenses & Certifications**.

Schema Location	Field Name	Is Required (Y/N)
/data/eventContext		
/data/eventContext/associateOID		
/data/transforms/associateCertification/certificationID	**License/Certification ID**	
/data/transforms/associateCertification/certificationID/idValue		
/data/transforms/associateCertification/certificationNameCode	**License/Certification**	
/data/transforms/associateCertification/certificationNameCode/codeValue		
/data/transforms/associateCertification/categoryCode	**Category**	
/data/transforms/associateCertification/categoryCode/codeValue		
/data/transforms/associateCertification/issuingParty		
/data/transforms/associateCertification/issuingParty/nameCode	**Issued By**	
/data/transforms/associateCertification/firstIssueDate	**Effective Date**	
/data/transforms/associateCertification/expirationDate	**Expiration Date**	
/data/transforms/associateCertification/comments	**Comments**	
/data/transforms/associateCertification/employerPaidAmount/amountValue	**Company Paid Amount**	
/data/transforms/associateCertification/employerPaidAmount/currencyCode		
/data/transforms/associateCertification/employerPaidAmount/currencyCode/codeValue	**Currency**	
/data/transforms/associateCertification/renewalComments	**Renewal Requirement**	

Responses

You may encounter exceptions outside your common success scenarios. You must account for these exceptions during your initial development.

For more information, see API Common Exceptions and Tips for Handling.

Response Code - Clock.Punch	messageTxt	Tips to Handle

400 Bad Request	"userMessage": {
	"messageTxt": "A required HTTP header was not specified."
	}	

403 Forbidden	"userMessage": {
	"messageTxt": "Server failed to authenticate the request"
	}	

405 Method Not Allowed	"userMessage": {
	"messageTxt": "The request method is not allowed for the resource identified by the request URI."
	}	

406 Not Acceptable	"userMessage": {
	"messageTxt": "The API is not able to generate the any of the client's preferred content characteristics according to the request's accept headers."
	}	

412 Precondition Failed	"userMessage": {
	"messageTxt": "The precondition given in one or more of the request-header fields evaluated to false when it was tested on the server."
	}	

429 Too Many Requests	"userMessage": {
	"messageTxt": "The rate limit has been exceeded. Try again later."
	}	

500 Internal Server Error	"userMessage": {
	"messageTxt": "The server encountered an internal error. Please retry the request."
	}	

503 Service Unavailable	"userMessage": {
	"messageTxt": "The server is currently unable to receive requests. Please retry your request."
	}	

Response Code- Clock.Punch/Meta	messageTxt	Tips Hand
304 Not Modified		

400 Bad Request	"userMessage": {
	"messageTxt": "A required HTTP header was not specified."
	}	

Response Code-Clock.Punch/Meta	messageTxt	Tips Hand

403 Forbidden	"userMessage": {
	"messageTxt": "Server failed to authenticate the request"
	}	

404 Not Found	"userMessage": {
	"messageTxt": "The specified resource does not exist."
	}	

405 Method Not Allowed	"userMessage": {
	"messageTxt": "The request method is not allowed for the resource identified by the request URI."
	}	

406 Not Acceptable	"userMessage": {
	"messageTxt": "The API is not able to generate the any of the client's preferred content characteristics according to the request's accept headers."
	}	

412 Precondition Failed	"userMessage": {
	"messageTxt": "The precondition given in one or more of the request-header fields evaluated to false when it was tested on the server."
	}	

413 Request Entity Too Large	"userMessage": {
	"messageTxt": "The precondition given in one or more of the request-header fields evaluated to false when it was tested on the server."
	}	

429 Too Many Requests	"userMessage": {
	"messageTxt": "The rate limit has been exceded. Try again later."
	}	

500 Internal Server Error	"userMessage": {
	"messageTxt": "The server encountered an internal error. Please retry the request."
	}	

503 Service Unavailable	"userMessage": {
	"messageTxt": "The server is currently unable to receive requests. Please retry your request."
	}	

Chapter 2

Data Collection Entries Web Service Method - Upload - POST

Data Collection Entries Web Service Method - Upload - POST

The data-collection-entries.process HTTP POST method is used to upload a payload of punches to the ADP TLM servers for processing.

Action Code/ Entry Codes Overview

This section describes the validate Action code and Entry Code to be used in the API.

Action and Entry Code JSON Object Locations

Element Name	JSON Object Location
Action Code	/events/data/transform/dataCollectionEntries/workerDataCollectionEntries/actionCode/codeValue
Entry Code	/events/data/transform/dataCollectionEntries/workerDataCollectionEntries/entryCode/codeValue

Action Code/Entry Code Description Matrix

Action Code	Entry Code	Description
punch	NA	Scan operation (first scan represents an "IN"; second scan represents an "OUT").
lunchout	NA	Clock out for lunch operation.
MealReturn	NA	Clock in for lunch operation
clockin	NA	Clock-in operation
clockout	NA	Clock-out operation
transfer	NA	Simultaneous clock-out and clock-in for work change

Action Code/Entry Code Not Supported for WorkforceNow Next Gen

Action Code	Entry Code	Description
earnings	**{required}**	**Earning are not supported for WorkforceNow Next Gen**
suppEarnings	**{required}**	**Earning are not supported f**or WorkforceNow Next Gen

General Comments and Notes

- The target clocking employees are always specified in the body by using the Badge value
- For all HTTP Status responses of 202, The "Location" and "Retry-After" response headers indicate that a subsequent request can be made to the data-collection-entries/{eventid} HTTP GET method to obtain an import and/or timecard posting status of the uploaded punches
- The data-collection-entries.process web service is designed to be used asynchronously:
    - Honoring the "Retry-After" response header returned by the data-collection-entries.process and data-collection-entries/{eventid} web service pertains only to the subsequent requests to obtain a status for a particular Event ID (using data-collection-entries/{eventid}).
    - Subsequent requests can be made to the data-collection-entries.process web service to upload additional punches while asynchronously polling/waiting for previous bulk upload events to complete
    - To limit inadvertent DOS (Denial of Service) incidences, ADP asks consuming applications to honor a wait interval of at least one minute between each subsequent bulk upload request made to data-collection-entries.process.

- Refer to the latest "Marketplace Integration Guides" for the following:
    - Creating your Personal Information Exchange (PFX) Certificate.
    - Installing Your PFX Certificate.
    - Installing and Preparing Google Postman
    - Requesting a Bearer Token

Request Header Parameters

For API related information please refer Data Collection Entries in API explorer

Header Name	Sample value	Comments
Authorization	Bearer {accessToken}	The Token Type (accessToken—i.e. "Bearer") plus the Access Token (access_token—i.e. "edc3c2-6b5d-4251-b258-d1208cbf6339") returned from ADP Marketplace's Token web service.

Multiple Punches Sample

This is an example of a request that contains multiple time punch entries in the uploading payload

API:	/events/time/v1/data-collection-entries.process
Method:	POST
Condition:	The Payload has been accepted and stored for further processing by ADP Marketplace. Subsequent requests to the datacollection-entries.process, Status service are required by providing the **eventID** value returned in the **confirmMessage** object within the response to confirm the processing status of the submitted bulk upload. The Response Code applies specifically to the POST Method request.

Scan/Punch

Performs a scan punch operation where the first scan represents an "IN" punch and the next scan represents an "OUT" punch.

API:	/events/time/v1/data-collection-entries.process
Meth	POST

od:	
Condition:	The Payload has been accepted and stored for further processing by ADP Marketplace. Subsequent requests to the datacollection entries.process, Status service are required by providing the **eventID** value returned in the **confirmMessage** object within the response to confirm the processing status of the submitted bulk upload. The Response Code applies specifically to the POST Method request.

Lunch Out

Performs a lunch out operation.

API:	events/time/v1/clock.punch
Method:	POST
Condition:	The Payload has been accepted and stored for further processing by ADP Marketplace. Subsequent requests to the datacollection entries.process, Status service are required by providing the **eventID** value returned in the **confirmMessage** object within the response to confirm the processing status of the submitted bulk upload. The Response Code applies specifically to the POST Method request.

Meal Return

Performs a meal return operation.

API:	/events/time/v1/data-collection-entries.process
Method:	POST
Condition:	The Payload has been accepted and stored for further processing by ADP Marketplace. Subsequent requests to the datacollection entries.process, Status service are required by providing the **eventID** value returned in the **confirmMessage** object within the response to confirm the processing status of the submitted bulk upload. The Response Code applies specifically to the POST Method request.

Clock In

Performs a clock-in operation.

API:	/events/time/v1/data-collection-entries.process
Method:	POST
Condition:	The Payload has been accepted and stored for further processing by ADP Marketplace. Subsequent requests to the datacollection entries.process, Status service are required by providing the **eventID** value returned in the **confirmMessage** object within the response to confirm the processing status of the submitted bulk upload.

	The Response Code applies specifically to the POST Method request.

Clock Out

API:	/events/time/v1/data-collection-entries.process
Method:	POST
Condition:	The Payload has been accepted and stored for further processing by ADP Marketplace. Subsequent requests to the datacollection entries.process, Status service are required by providing the **eventID** value returned in the **confirmMessage** object within the response to confirm the processing status of the submitted bulk upload. The Response Code applies specifically to the POST Method request.

Change Work/Transfer

API:	/events/time/v1/data-collection-entries.process
Method:	POST
Condition:	The Payload has been accepted and stored for further processing by ADP Marketplace. Subsequent requests to the datacollection entries.process, Status service are required by providing the **eventID** value returned in the **confirmMessage** object within the response to confirm the processing status of the submitted bulk upload. The Response Code applies specifically to the POST Method request.

Data Collection Entries API Element Glossary

Selected important elements of the payload API are defined below for user reference and information. Any elements not documented below are assumed to be non-applicable and can be ignored for purposes of the data-collection-entries HTTP POST web service method. Elements that have a "Required?" value of "N/A" are data elements that are typically found in the response of an upload request (and would not be needed in the upload request itself but are documented to indicate their importance in subsequent requests that might be made in order to obtain an upload status).

Events[]

Element Name	Description	Required?
events[]	An array of events. Typically, only one event (events[0]) need ever be supplied. The events[] array element is the parent object of the dataCollectionEntries[] array element.	Yes
events[].eventID	The eventID is the value that can be used to reference an upload event in a subsequent request to obtain upload status. It typically is found in the response API of the initial data-collection-entries.process upload request made.	N/A
events[].serviceCatogoryCode.codeValue	Should always have a value of "core" (constant value).	Yes

dataCollectionEntries[]

The table below describes the elements of the events[].data.transform.dataCollectionEntries[] array objects.

Element Name	Description	Required?
Element Name	Description	Required?
itemID	The itemID is used to identify the dataCollectionEntries array element. It is typically an integer value that represents the indexed position of the array element being described. For example, dataCollectionEntries[0] would have a value of "0", events[1].eventID would have a value of 1, etc.	Yes
terminalName	Represents the ID of the terminal associated with the punches described by the workerDataCollectionEntries[] array (clock name).	NO

workerDataCollectionEntries[]

The table below describes selected elements of the events[].data.transform.dataCollectionEntries[].workerDataCollectionEntries array objects. All other elements of the workerDataCollectionEntries array objects not defined here can be ignored and are not required.

Element Name	Description	Required?
entryID	The entryID is used to identify the workerDataCollectionEntries array element. It is typically an integer value that represents the indexed position of the array element being described. For example, dataCollectionEntries[0] would have a value of "0", events[1].eventID would have a value of 1, etc.	Yes
entryCode.codeValue	Used to assign the Earning or Supplemental earning code assocated with an "earnings" or "suppEarnings" action code. This is a required value whenever an "earnings" or "suppEarnings" action code is specified. Not required for all other cases. See "Standard ADP Available Entry Code Values" for more information about the codes that can be used for this assignment.	Required when used with the "earnings" and "suppEarnings" action codes
badgeID	The badge value associated with the time punch being recorded.	Yes
actorBadgeID	The badge value associated with a "punch edit" made at the clock. Typically, it represents the badge value of the supervisor adding a missed punch or recording an allocation on behalf of an employee.	No
deviceDateTime	The timestamp of the data collection device for the punch being recorded. It has the following format: YYYY-MM-DDTHH:MI:SS-HH:MI For example: 2016-11-11T11:30:15-05:00 Where "-05:00" is the hour/minutes offset from UTC.	
entryDateTime	The timestamp of the Punch being recorded for the employee. When the punch is not an edit (either a Supervisor or Employee Edit operation at the clock), the entryDateTime will typically match the deviceDateTime. Uses the same date format as deviceDateTime.	Yes
actionCode	The action code associated with the punch being recorded. See "Action Code/Entry Code Description Matrix" for the set	Yes

Element Name	Description	Required?
	of action codes that can be used for this assignment.	
alternateWokerIDs	Used to identify the Employee Position belonging to the punch by alternative elements, such as ASSOCIATEOID. To date, only ASSOCIATEOID is presently supported. See "Alternate Worker ID's (alternateWokerIDs and alternateActorWokerIDs)" for property information.	No
alternateActorWokerIDs	Used to identify the Employee Position belonging to the Punch Edit being made (Supervisor Edit) that might belong to a Punch using an alternative element, such as an ASSOCIATEOID. See "Alternate Worker ID's (alternateWokerIDs and alternateActorWokerIDs)" for property information.	No

For API related information please refer Data Collection Entries in API explorer

Chapter 3

Data Collection Entries Web Service Method - Status - GET

Data Collection Entries Web Service Method – Status - GET

The data-collection-entries.process/{eventid} HTTP GET method reports on the status of a previously submitted data-collection-entries.process HTTP POST punch uploaded payload method request.

Request

URI Format	/events/time/v1/data-collection-entries.process/{eventid} note eventID need to be fetch from POST API response body
METHOD	GET
HTTP Headers	**Authorization:** Bearer {accessToken}

Request Header Parameters

Header Name	Sample value	Comments
Header Name	Sample value	Comments
Authorization	Bearer {accessToken}	The Token Type (accessToken—i.e. "Bearer") plus the Access Token (access_token—i.e. "edc3c2-6b5d-4251-b258-d1208cbf6339") returned from ADP Marketplace's Token web service.

Possible Responses

201 - Succesful Import Status Response

This response indicates all the punch records in the payload completed, imported and posted successfully.

For API related information please refer Data Collection Entries in API explorer.

API:	/events/time/v1/data-collection-entries.process/{event-id}
Method:	Get
Condition:	This is get call with eventid which is part of confirm message for post call made earlier.
GitHub Sample Request/Sample Response Payload:	**Response:** success-response.json

207– Partial Failures

This response occurs when some punch records fail to import or post to the timecard.

For API related information please refer Data Collection Entries in API explorer.

API:	/events/time/v1/data-collection-entries.process/{event-id}
Method:	Get
Condition:	This is get call with eventid which is part of confirm message for post call made earlier.
GitHub Sample Request/Sample Response Payload:	**Response:** Get api response Partialfailure.json

400 – Transaction Failure Response

This response occurs when all the punches failed to import or post to the timecard.

For API related information please refer Data Collection Entries in API explorer.

API:	/events/time/v1/data-collection-entries.process/{event-id}
Method:	Get
Condition:	This is get call with eventid which is part of confirm message for post call made earlier.
GitHub Sample Request/Sample Response Payload:	**Response:** Get api failure response.txt

404 – Resource Not Found Response

This response occurs when the requested upload event is not found in the database.

For API related information please refer Data Collection Entries in API explorer.

API:	/events/time/v1/data-collection-entries.process/{event-id}
Method:	Get
Condition:	This is get call with eventid which is part of confirm message for post call made earlier.
GitHub Sample Request/Sample Response Payload:	**Response:** get api 404 response.txt

For API related information please refer Data Collection Entries in API explorer

Chapter 4

Message Code Dictionary

Message Code Dictionary – userMessage.CodeValue

The example below includes a subset of possible error messages that be assigned to the "userMessage.codeValue" property that can be included in a response. The full list of error codes are defined in the table that follows this example response.

HTTP Headers	Content-Type: application/json;charset=UTF-8 Status Code: 400
Sample Response Body	For full response details, view specs in API Explorer

Errors

Message ID	Description
err_DCE_IMP_DEVICEDATEMISSING	Invalid or missing deviceDateTime.
err_DCE_IMP_ENTRYDATEMISSINGORINVALID	Invalid or missing EntryDateTime
err_DCE_IMP_ACTIONCODEINVALID	Invalid action code.

Message ID	Description
err_EmptyBadge	Missing badge value.
err_InvalidBadge	Invalid Badge ID
info_DCE_IMP_TOTALCOUNT	Bulk Punch Statistic Import Statistics (total record count found in the payload). Typically, only one message of this type occurs in the response.
info_DCE_IMP_FAILEDCOUNT	Bulk Punch Statistic Import Statistics (number of punches failing to import). Typically, only one message of this type occurs in the response.
info_DCE_IMP_INPROCESSCOUNT	Bulk Punch Statistic Import Statistics (number still in process). Typically, only one message of this type occurs in the response.
info_DCE_PST_TOTALCOUNT	Bulk Punch Statistic Posting Statistics (total record count considered for Posting). Typically, only one message of this type occurs in the response. This message may not exist at all if all punch records failed to import.
info_DCE_PST_FAILEDCOUNT	Bulk Punch Statistic Posting Statistics (number of punch records that failed to post to the timecard). Typically, only one message of this type occurs in the response. This message may not exist at all if all punch records imported into the system without any errors.
err_DCE_PST_COMMON	Indicates that the error is associated with the attempt to post the successfully imported time punch record to the timecard (i.e. while the import of the punch record succeeded the punch record failed to reach the employee's timecard).

For API related information please refer Data Collection Entries in API explorer

Chapter 5

Frequently asked questions

Chapter 6

Known Issues and Limitations