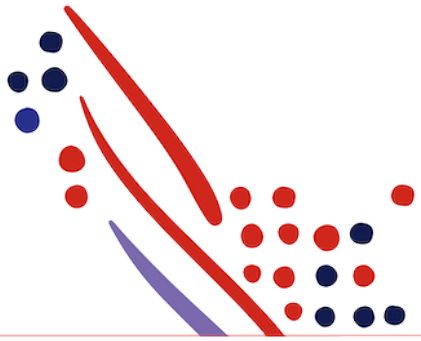
Guide

Payroll Data Input API Guide for RUN Powered by ADP

Published on
Mar 03, 2020 12:35PM

Last modified
Aug 25, 2022 5:51AM

Always Designing for People™



ADP Copyright Information

Published on
Mar 03, 2020 12:35PM

Last modified
Aug 25, 2022 5:51AM

Table of Contents

Chapter 1

About the API

Summary

The Payroll Data Input Application Programming Interface (API) enables users to create a pay data batch for a payroll cycle. These APIs are used to view, add, or replace worker pay data for a payroll cycle.

What's New in this Guide?

August 2022

- Added the pay data input API's 504 error description in the Chapter 7 - known issues and limitations section.
- Added the steps to check the API GET call from UI prospective in the Chapter 6 - Frequently Asked Questions.

March 2020

The Payroll Data Input API now accepts negative values in the **rateValue** resource for an earning code with the type of **Amount**.

April 2020

There was a new question added for the CARE Act in response to COVID-19. See Chapter 5 - Frequently Asked Questions for more information.

July 2020

Added **Required Setup Steps – Marketplace Connector for PayData Input API** to this chapter.

October 2020

Added a new section **Use Case: Retrieving Code Lists** in Chapter 5.

Supported Product Version and Customer Base

The Payroll Data Input API is supported for the following RUN Powered by ADP bundles:

- ADP Essential Payroll
- ADP Enhanced Payroll
- ADP Complete Payroll and HR Plus

Process Overview

The following table shows an illustration of how your data connector application would be used by a client.

	Actor	Task Description
1	Payroll Practitioner	Purchases Time-related application from Marketplace and provides consent.
2	Marketplace	Sends application purchase notification to RUN Powered by ADP.
3	RUN Powered by ADP	Enable Marketplace Connector for PayData Input API for the Client.
4	Your application	Uses Workers and Code list APIs to synchronize the following between RUN Powered by ADP and your application: • Workers • Earnings • Departments • Pay rates • Pay cycles data
5	Your application	Captures time data for employees and exports to RUN Powered by ADP for an open pay frequency.
6	Payroll Practitioner	Reviews and approves the time data imported and processed in a payroll run.

Required Setup Steps – Marketplace Connector for PayData Input API

Marketplace Connector for Pay Data Input API is a feature that allows you to import data from your time and attendance system into RUN Powered by ADP for each payroll. To use the Pay Data Input API, the Marketplace Connector must be selected on the **Features Overview** page, which is accessed in the RUN Powered by ADP user interface (UI) by selecting **Company > Features Overview**. This feature gets enabled by the default on the Partner Application Subscription (if the application has a Pay Data Input API scope setup on the ADP Marketplace). In case of any failure, the API returns the error as **Marketplace Connector for PayData Input feature not set up in payroll** then Partner/Client can contact the ADP Service to get this enabled.

Note: Currently, the Pay Data Input API's supports multiple partners.

Postman Collection

Postman allows you to import a collection of APIs, created by others, so you can try them out. For more information on Postman, see Making Your First API Call Using Postman.

To download API collections for the Payroll Data Input API from the ADP GitHub library and import them to Postman, go to Payroll Data Input API Postman Collection.

Chapter 2

Use Case: Retrieving the Pay Data Input Details

Use Case Description

This use case requests the list of all available pay data input that the requester is authorized to view. The Payroll Data Input API exposes values found in the RUN Powered by ADP UI by selecting **Home > Time Sheet > Time Sheet Import**.

API Usage

Method	URI	Description	GitHub Sample Request Payload
GET	/payroll/v1/pay-data-input	Requests the list of all available pay data input the requester is authorized to view.	NA

Application Scope

The canonical Uniform Resource Indicator (URI) corresponding to the API needs to be added in the Consumer Application Registry (CAR) for the subscription following which a user can access this API and make successful API calls.

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

/payroll/payrollManagement/payrollProcessing/payDataInputManagement/payDataInput.read

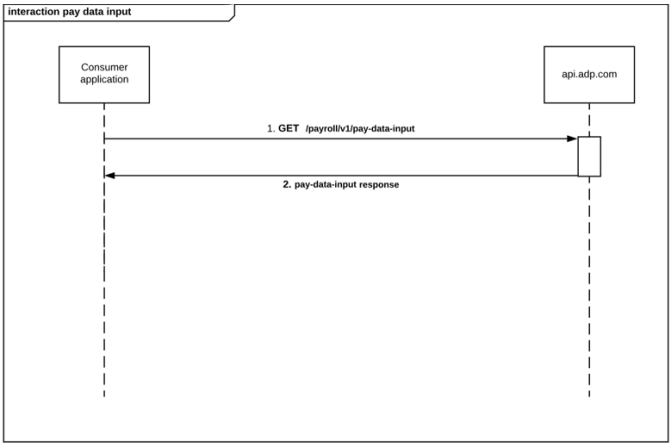
Supported Actors

Request Parameter roleCode Value	Usage
practitioner	Retrieves an employee's pay data input details. A system user is considered a practitioner.

Request Header Parameters

Apart from the ADP standard header parameters, there are no request header parameters.

Sequence of Interactions



The following are the steps shown in the previous diagram:

1. Your consumer application makes a request to the ADP API endpoint to GET the Pay Data input for a worker.
2. The ADP API endpoint responds to your consumer application with single worker Pay Data input information.

Data Dictionary

Resources listed in the following table can be accessed in the RUN Powered by ADP UI by selecting **Home > Time Sheet > Time Sheet Import > Select the Pay Frequency**.

Schema Location	Field Name in RUN Powered by ADP
Schema Location	Field Name in RUN Powered by ADP
/payDataInput/payeePayInputs/associateOID	NA
/payDataInput/payeePayInputs/workerID/idValue	NA
/payDataInput/payeePayInputs/payPeriodStartDate	**Pay period start**
/payDataInput/payeePayInputs/payPeriodEndDate	**Pay period end**
/payDataInput/payeePayInputs/payrollProfilePayInputs/payInputs/earningInputs/earningCode/codeValue	NA
/payDataInput/payeePayInputs/payrollProfilePayInputs/payInputs/earningInputs/rate/rateValue	**Rate/Salary Amount Import**

/payDataInput/payeePayInputs/payrollProfilePayInputs/payInputs/earningInputs/rate/currencyCode	NA
/payDataInput/payeePayInputs/payrollProfilePayInputs/payInputs/earningInputs/rate/baseUnitCode/codeValue	NA
/payDataInput/payeePayInputs/payrollProfilePayInputs/payInputs/earningInputs/rate/baseUnitCode/shortName	NA
/payDataInput/payeePayInputs/payrollProfilePayInputs/payInputs/earningInputs/numberOfHours	**Regular Hours Import**
/payDataInput/payeePayInputs/payrollProfilePayInputs/payInputs/payAllocation/allocationTypeCode/codeValue	NA
/payDataInput/payeePayInputs/payrollProfilePayInputs/payInputs/payAllocation/allocationID	**Department**
/payDataInput/payeePayInputs/payNumber	NA
/payrollGroupCode/codeValue	**Pay frequency**
/itemID	NA

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	Response Condition	message Txt	GitHub Sample Request Payload	GitHub Sample Response Payload
200 OK	Retrieve the pay data input batch details when there are	NA	NA	No pay data batch entries Response

	no entries.			
200 OK	Retrieve the pay data batch details.	NA	NA	pay data batch details Response

Chapter 3

Use Case: Adding New Entries in the Pay Data Input Batch

Use Case Description

This use case allows your application to import time sheet data for a period and exposes values found in the RUN Powered by ADP UI by selecting **Home > Time Sheet > Time Sheet Import**.

API Usage

Method	URI	Description	GitHub Sample Request Payload	GitHub Sample Response Payload
POST	/events/payroll/v1/pay-data-input.add	Adds pay data input information for an associate.	Pay-data-input.add Request	Pay-data-input.add Response
GET	/events/payroll/v1/pay-data-input.add/meta	Returns an event metadata.	NA	pay-data-input.add-meta Response
GET	/codelists/payroll/v1/payroll-processing/pay-cycles/run	Returns the pay cycles.	NA	Codelist pay cycles Response
GET	/codelists/payroll/v1/payroll-processing/earnings/run	Returns the earnings code.	NA	Codelist Earnings Response
GET	/codelists/payroll/v1/payroll-processing/pay-rates/run	Returns the pay rates.	NA	Codelists Payrates Response

In the given payload (Pay-data-input.add Request), the following is the purpose of the **eventContext** and **transform** sections:

- **eventContext**: A set of keys, identifying the subject. In the payload, the **associateID** field is present under **eventContext**. The **associateID** identifies the subject.
- **transform**: Provides the values added or changed with respect to the subject keys defined in the **eventContext** section.

Application Scope

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

/payroll/payrollManagement/payrollProcessing/payDataInputManagement/pay-data-input.add

/core/tableManagement/codeListManagement/codeListViewing/codelist.read

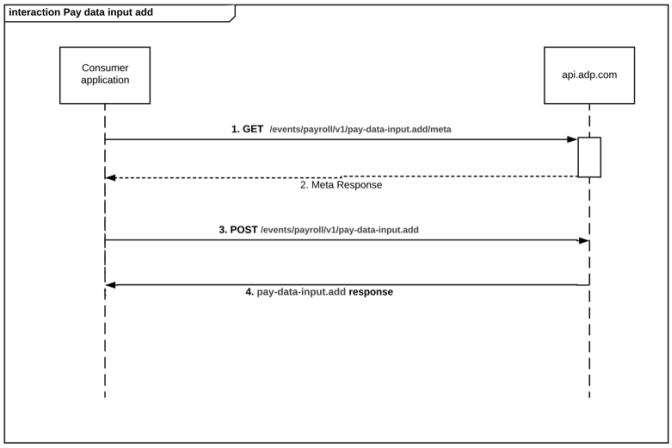
Supported Actors

Request Parameter roleCode Value	Usage
practitioner	Adds pay data input information for an associate. A system user is considered a practitioner.

Request Header Parameters

Apart from the ADP standard header parameters, there are no request header parameters.

Sequence of Interactions



The following are the steps shown in the previous diagram:

1. Your consumer application makes a request for /events/payroll/v1/pay-data-input.add/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. Your consumer application processes the meta payload to validate data included in the payload in step 3 and prompts the user to fix any validation issues. Otherwise, compose the payload for the /events/payroll/v1/pay-data-input.add request.
3. Your consumer application makes a /events/payroll/v1/pay-data-input.add request the ADP API endpoint.
4. The ADP API endpoint responds to the consumer application concerning the details of /events/payroll/v1/pay-data-input.add.

Data Dictionary

Resources listed in the following table can be accessed in the RUN Powered by ADP UI by selecting Home > Time Sheet > Time Sheet Import > Select the Pay Frequency.

Schema Location	Field Name in RUN Powered by ADP	Note
/payrollGroupCode/codeValue	**Pay frequency**	For example: Biweekly, Weekly, Monthly, Quarterly, and so on.
/payDataInput/payee	NA	

PayInputs/associateOID		
/payDataInput/payeePayInputs/payPeriodStartDate	**Pay period start**	Indicates the pay frequency start date.
/payDataInput/payeePayInputs/payPeriodEndDate	**Pay period end**	Indicates the pay frequency end date.
/payDataInput/payeePayInputs/payrollProfilePayInputs/payInputs/earningInputs/earningCode/codeValue	NA	Indicates the earning code type.
/payDataInput/payeePayInputs/payrollProfilePayInputs/payInputs/earningInputs/rateCode/codeValue	**Rate**	Indicates the pay rate.
/payDataInput/payeePayInputs/payrollProfilePayInputs/payInputs/earningInputs/numberOfHours	**Regular Hours Import**	Indicates the hours for an earning code. Depending on the type of earning code, the value is reflected for a column on the RUN Powered by ADP user interface (UI). For example, the earning code might be **Regular**.
/payDataInput/payeePayInputs/payrollProfilePayInputs/payInputs/earningInputs/rate/rateValue	Salary Amount Import	Indicates the amount or salary for an earning code. Depending on the type of earning code, the amount is reflected for column on the RUN Powered by ADP UI. For example, earning code might be **Regular**.
/payDataInput/payeePayInputs/payrollProfilePayInputs/payInputs/payAllocation/allocationTypeCode/codeValue	**Department**	The **payAllocation** parameter is tied to **Department**, which is tied to a payrate, accessed in the RUN Powered by ADP UI by selecting **Employee profile > Payroll Info page**.
/payDataInput/payeePayInputs/payrollProfilePayInputs/payInputs/payAllocation/allocationID	**Allocation ID**	Specifies the numeric ID of **Department** in **Payroll Info** relating to **payrate**.
/payDataInput/payeePayInputs/payrollProfilePayInputs/payInputs/earningInputs/rate/baseUnitCode/codeValue	**Rate**	Indicates the custom rate.
/payDataInput/payeePayInputs/payNumber	NA	The **payNumber** parameter is optional for each /meta call. It represents a line item for the payroll import. If you are entering three different pays for one employee, you can enter: • "payNumber":"1" • "payNumber":"2" • "payNumber":"3"

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.

ResponseCode	Request Condition	message Txt	GitHub Sample Request Payload	GitHub Sample Response Payload	Tips to Handle
ResponseCode	Request Condition	message Txt	GitHub Sample Request Payload	GitHub Sample Response Payload	Tips to Handle
200 OK	Add a pay data input batch for an associate with **Pay Frequency** set at Weekly.	NA	Weekly Pay Frequency Request	Weekly Pay Frequency Response	
200 OK	Add a pay data input batch for an associate with **Pay Frequency** set at BiWeekly.	NA	BiWeekly Pay Frequency Request	BiWeekly Pay Frequency Response	
200 OK	Add a pay data input batch for an associate with **Pay Frequency** set at Semimonthly.	NA	Semimonthly Pay Frequency Request	Semimonthly Pay Frequency Response	
200 OK	Add a pay data input batch for an associate with **Pay Frequency** set at Monthly.	NA	Monthly Pay Frequency Request	Monthly Pay Frequency Response	
200 OK	Add a pay data input batch for an associate with **Pay Frequency** set at Quarterly.	NA	Quarterly Pay Frequency Request	Quarterly Pay Frequency Response	
200 OK	Add a pay data batch with different earning codes for the same associate.	NA	Different Earning Codes Request	Different Earning Codes Response	
200 OK	Add a pay data batch with the **Regular** earning code passing hours and amount.	NA	Regular Earning Code Request	Regular Earning Code Response	
200 OK	Add a pay data batch with the **Overtime** earning code passing hours and Amount.	NA	Overtime Earning Code Request	Overtime Earning Code Response	

200 OK	Add a pay data batch with the Vacation earning code.	NA	Vacation Earning Code Request	Vacation Earning Code Response	
200 OK	Add a pay data batch with the **Sick** earning code.	NA	Sick Earning Code Request	Sick Earning Code Response	
200 OK	Add a pay data batch with a different check number and different earning code for an associate.	NA	Different Check Number Request	Different Check Number Response	
200 OK	Add a pay data batch when using **Department**.	NA	Department Request	Department Respons e	
200 OK	Add a pay data batch when using **Custom rate**.	NA	Custom Rate Request	Custom Rate Response	
200 OK	Add a pay data batch for different **PayRates**.	NA	Different Pay rates Request	Different Pay rates Response	
200 OK	Add a pay data batch for multiple associates.	NA	Multiple Associates Request	Multiple Associates Response	
200 OK	Add a pay data batch when earning amount (**rateValue**) is negative for the **BASE Payrate**.	NA	Negative_rateValue_Re q.json	Negative_rateValue_Re s.json	
200 OK	Add a pay data batch when earning amount (**rateValue**) is negative for **RATE_2**.	NA	Negative_Rate_Value_R ate_2_Req.json	Negative_rate_value_R ate_2_Res.json	
400 Bad Request	Add a pay data batch when there is an existing pay data file.	"userMessage": {"messageTxt": "An Open Timesheet already exists for this Company and Pay Frequency"}	Existing batch Request	Existing batch Response	User to check if there is an existing pay data batch.
400 Bad Request	Add a pay data batch when **payrollGroupC ode** code is blank.	"userMessage": {"messageTxt": "Events[0]. Data.EventContext.Pa yload. CodeValue: The CodeValue field is required."}	pay-data-input.add-payrollgroupcode-blank-http-400-request.json	pay-data-input.add-payrollgroupcode-blank-http-400-response.json	User to check that **payrollGroupCo de** is not empty.
400 Bad Request	Add a pay data batch when **payrollGroupC ode** is not valid.	"userMessage": {"messageTxt": "It looks like the pay frequency (%%1) in your pay data import file doesn't match the pay frequency of this payroll.	Invalid payrollGroup Code Request	Invalid payrollGroup Code Response	User to check that a valid **payrollGroupCo de** is passed.

		Make sure you selected the correct time sheet file for this pay frequency. (34611)\n"}			
400 Bad Request	Add a pay data batch when hours for earning is a negative value.	"userMessage": {"messageTxt": "Events[0].Data.Transform.Payload.PayeePayInputs[0].PayrollProfilePayInputs[0].PayInputs[0].EarningInputs[0].numberOfHours: The field numberOfHours must be between 0 and 9999.99."}	pay-data-input.add-earninghours-invalid-http-400-request.json	pay-data-input.add-earninghours-invalid-http-400-response.json	User to check hours for earning code, which should be within the range **0 and 9999.99**.
400 Bad Request	Add a pay data batch when earning amount (**rateValue**) is negative lower than -9999999.99.	"Events[0].Data.Transform.Payload.PayeePayInputs[0].PayrollProfilePayInputs[0].PayInputs[0].EarningInputs[0].rate.RateValue: The field RateValue must be between -9999999.99 and 9999999.99."	Negative_rateValue_beyondRange_Req.json	Negative_rateValue_beyondRange_Res.json	User to check amount for earning code should be within the range **0 and 9999.99**.
400 Bad Request	Add a pay data batch when check number is not in the following range (0-9).	"userMessage": {"messageTxt": "Events[0].Data.Transform.Payload.PayeePayInputs[0].PayNumber:The field PayNumber must match the regular expression '^[0-9]$'."}	pay-data-input.add-PayNumber-invalid-http-400-request.json	pay-data-input.add-PayNumber-invalid-http-400-response.json	User to check **PayNumber** should match within the range of **0 and 9**.

Chapter 4

Replacing the Entries in the Pay Data Input Patch

Use Case Description

This use case replaces pay data input information with new data and exposes values found in the RUN Powered by ADP UI by selecting **Home > Time Sheet > Time Sheet Import**.

API Usage

Method	URI	Description	GitHub SampleRequest Payload	GitHub SampleResponse Payload

POST	/events/payroll/v1/pay-data-input.replace	Replaces pay data input information.	pay-data-input.replace Request	pay-data-input.replace Response
GET	/events/payroll/v1/pay-data-input.replace/meta	Returns an event metadata.		pay-data-input.replace-meta Response
GET	/codelists/payroll/v1/payroll-processing/earnings/run	Returns the earnings code.		Codelist Earnings Response
GET	/codelists/payroll/v1/payroll-processing/pay-rates/run	Returns the pay rates.		Codelist Pay rates Response

In the given payload (pay-data-input.replace Request), the following is the purpose of the **eventContext** and **transform** sections:

- **eventContext**: A set of keys, identifying the subject. In the payload, the **associateID** field is present under **eventContext**. The **associateID** identifies the subject.
- **transform**: Provides the values added or changed with respect to the subject keys defined in the **eventContext** section.

Application Scope

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

/payroll/payrollManagement/payrollProcessing/payDataInputManagement/pay-data-input.replace

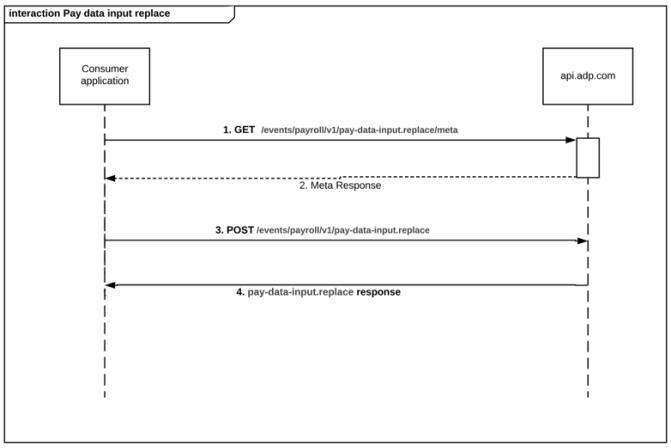
Request Header Parameters

Apart from the ADP standard header parameters, there are no request header parameters.

Supported Actors

Request Parameter roleCode Value	Usage
practitioner	Replaces pay data input information for an associate. A system user is considered a practitioner.

Sequence of Interactions



The following are the steps shown in the previous diagram:

1. Your consumer application makes a request for /events/payroll/v1/pay-data-input.replace/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. Your consumer application processes the meta payload to validate data included in the payload in step 3 and prompts the user to fix any validation issues. Otherwise, compose the payload for the /events/payroll/v1/pay-data-input.replace request.
3. Your consumer application makes a /events/payroll/v1/pay-data-input.replace request the ADP API endpoint.
4. The ADP API endpoint responds to the consumer application concerning the details of /events/payroll/v1/pay-data-input.replace.

Data Dictionary

Resources listed in the following table can be accessed in the RUN Powered by ADP UI by selecting **Home > Time Sheet > Time Sheet Import >Select the Pay Frequency**.

Schema Location	Field Name in RUN Powered by ADP	Note
/itemId	NA	Indicates the item ID of the existing pay data batch. The itemID can be retrieved using GET /payroll/v1/pay-data-input API.
/payDataInput/payeePayInputs/associateOID	NA	
/payDataInput/payeePayInputs/payPeriodStartDate	**Pay period start**	Indicates the pay frequency start date.
/payDataInput/payeePayInputs/payPeriodEndDate	**Pay period end**	Indicates the pay frequency end date.
/payDataInput/payeePayInputs/payrollProfilePayInputs/payInputs/earningInputs/earningCode/codeValue	NA	Indicates the type of earning code.
/payDataInput/payeePayInputs/payrollProfilePayInputs/payInputs	**Rate**	Indicates the pay rate.

/earningInputs/rateCode/codeValue		
/payDataInput/payeePayInputs/payrollProfilePayInputs/payInputs/earningInputs/numberOfHours	**Regular Hours Import**	Indicates the hours for an earning code. Depending on the type of earning code, the value is reflected in the RUN Powered by ADP UI. For example, the earning code might be **Regular**.
/payDataInput/payeePayInputs/payrollProfilePayInputs/payInputs/earningInputs/rate/rateValue	**Salary Amount Import**	Indicates the amount/salary for an earning code. Depending on the type of earning code the amount is reflected in the ADP RUN UI. For example, earning code might be **Regular**.
/payDataInput/payeePayInputs/payrollProfilePayInputs/payInputs/payAllocation/allocationTypeCode/codeValue	NA	Indicates the **Department**.
/payDataInput/payeePayInputs/payrollProfilePayInputs/payInputs/payAllocation/allocationID	**Department**	Indicates the allocation ID for a **Department**.
/payDataInput/payeePayInputs/payrollProfilePayInputs/payInputs/earningInputs/rate/baseUnitCode/codeValue	**Rate**	Indicates the custom rate.

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.

Condition	**message txt**	**GitHub SampleRequest Payload**	**GitHub Sample Response Payload**	**Tips to Handle**
Condition	**message txt**	**GitHub SampleRequest Payload**	**GitHub Sample Response Payload**	**Tips to Handle**
Replace the pay data batch with a new batch.		Replace a new batch Request	Replace a new batch Response	
Replace a pay data batch with the **Regular Earning Code**, **passinghours**, and **amount**.		Regular Earning Code Request	Regular Earning Code Response	

Replace a pay data batch with **OverTime Earning Code**, **passinghours**, and **amount**.		OverTime Earning Code Request	OverTime Earning Code Response	
Replace a pay data batch with a different check number, with a different earning code for an associate.		Different PayNumber Request	Different PayNumber Response	
Replace a pay data batch when using **Department**.		Department Request	Department Response	
Replace a pay data batch when using a **Custom Rate**.		Custom rate Request	Custom Rate Response	
Replace a pay data batch for different **PayRates**.		Pay Rates Request	Pay Rates Response	
Replace a pay data batch with negative rate value for an earning amount		ReplacePayDataInput_Neg_Rate_value_REQ.json	Replace_Negative_Rate_value_Res.json	
Replace the pay data batch when an invalid item ID is passed.	"userMessage": { "messageTxt": "No such replaceable pay data found."}	Invalid ItemID Request	Invalid ItemID Response	User to check if a valid item ID in the existing pay data batch is passed in the request.
Replace a pay data batch when the check number is not within 0 through 9.	"userMessage": {"messageTxt": "Events[0].Data.Transform.Payload.PayeePayInputs[0].PayNumber:The field PayNumber must match the regular expression '^[0-9]$'."}	pay-data-input.replace-Paynumber-invalid-http-400-request.json	pay-data-input.replace-Paynumber-invalid-http-400-response.json	User to check if PayNumber should match within the range of 0 through 9.

Chapter 5

Use Case: Retrieving Code Lists

Use Case Description

Codelists make up the selection lists for many of the fields in RUN Powered by ADP. The Codelist API allows your application to retrieve a list of codes for a specific field, so it can be used for insert and update operations. For more information, see

CodeList API Guide For RUN Powered by ADP

Chapter 6

Frequently Asked Questions

Question 1: Do the ADP Partners have to send the rates if the Earning Code is Regular?

List the Earning Codes for which the rate should or should not be sent.

Answer: Yes, for every earning code, one for **rateCode** or **rateValue** should be sent. In the RUN Powered by ADP UI, go to **Company > Earnings Deductions > Set Up Earnings** to know to which earning codes we need to pass rates.

Question 2: What will happen if the ADP Partner sends different rates than what is configured in RUN Powered by ADP?

Answer: The **rateValue** passed in the API is considered the salary amount for an employee and **numberOfHours** is considered as hours for an employee for a pay period. If the rate is to be modified, refer to the Question 3.

Question 3: If the ADP Partners must send custom rates, what should they do?

There were some special occasions where the employees will be paid at a different rate than the regular rate.

Answer: Using the custom rate, partners can send a custom rate. Include the following .json in the payload to use the custom rate:

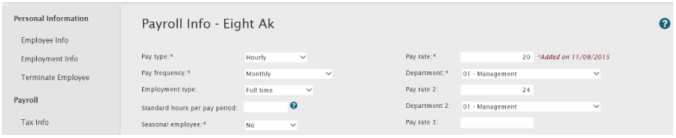
```
"rate": {
"rateValue": "1.00",
"baseUnitCode": {
"codeValue": "hour"
}
}
```

Question 4: Is the Support Shift Differential Pay using the Payroll Data Input API?

Answer: This can be done by setting up Custom Pay Rates on RUN Powered by ADP for the employee. Here is a scenario:

If employees work in non-preferred shifts like 10pm-7am, they will be paid 20% more salary than the normal rate, which is paid during the regular time of 8am-5pm. To achieve this, go to the RUN Powered by ADP UI and select **Dashboard > Employee Directory**. Then, select the desired employee and **Payroll / Payroll Info**.

Assuming the Employees Base Pay is 20 USD per hour, for the non-preferred shift, set the Pay Rate 2 to as 24 US dollars (USD), which is 20% more than the **Base Pay Rate**.



Assuming the employee worked 15 days at 9 hours per day on a normal shift (8am - 5pm) and 15 days at 9 hours per day on an irregular shift (10pm - 7am) for Regular work,
the Partner time keeping application, when sending the paydata input for the Client and the employee in question, should report the following time with the custom rate.

- Irregular Shift – Department: Management / Time: 135 hours (15 days * 9 hours per day) / Pay Rate 2
- Regular Shift – Department: Management / Time: 135 hours / Base Pay Rate

```
"payInputs": [{"earningInputs": [{ "earningCode": {"codeValue": "REG"},"rateCode": {"codeValue": "BASE"},
"numberOfHours": "135.00"},  {"earningCode": {"codeValue": "REG" },"rateCode": {"codeValue": "RATE_2"},
"numberOfHours": "135.00"}], "payAllocation": {"allocationTypeCode": {"codeValue": "DEPARTMENT"},
"allocationID": "Management"}}]
```

Similarly, the **Differential PayRate** may be for an **Earning type** other than a **Regular Pay**. Therefore, an appropriate **Earning code** must be sent. For example, **Overtime (OVT)**.

Question 5: What code can my application use to track the earnings for the CARE Act?

Earning Code	Description
FFFMLA	Families First FMLA Expansion
FFPSLEE	Families First Employee Pay
FFPSLFAM	Families First Family Care Pay

The COVID-19 situation is temporary in nature. Therefore, ADP may inactive these earning codes in the future.

Question 6: What is the process to check the data imported from RUN UI ?

Login to RUN application

Navigate to Payroll landing page by clicking the **Payroll** from Left hand navigation menu links

And then Click **Import timesheets** tile on Payroll landing page

 Or

Search with keyword "Time Sheet Import" in Search bar

 It should take to **Review time** flow.

Chapter 7

Known Issues and Limitations

Issue 1: Exception not handled correctly

Description

The Payroll Data Input API currently does not validate data the same way as people using file import.

When invalid pay period start date and end dates are passed in the API request payload, the API responds with 200 OK instead of a 400 Bad Request.

Impacted APIs

Method	URI	roleCode
POST	/events/payroll/v1/pay-data-input.add	Practitioner
GET	/events/payroll/v1/pay-data-input.add/meta	Practitioner
POST	/events/payroll/v1/pay-data-input.replace	Practitioner
GET	/events/payroll/v1/pay-data-input.replace/meta	Practitioner

Suggested Work Around

Build validation logic during the data collection process in your application prior to sending over to ADP for the following conditions.

Code	Message client would see when data is not validated by your application	Data validate
34611	It looks like the pay frequency (%%1) in your pay data import file doesn't match the pay frequency of this payroll. Make sure you selected the correct time sheet file for this pay frequency.	Pay Frequency matches value in RUN Powered by ADP.
34612	The pay period start and end dates (%%1 - %%2) in payroll don't match the pay period start and end dates (%%3 - %%4) in your time and attendance system for this payroll.	Pay periods dates matches value in RUN Powered by ADP.
34613	It looks like your pay data import file contains an employee (%%1) that does not exist in your payroll. Pay data was not imported for this employee.	Employee number matches value in RUN Powered by ADP.
34614	We noticed that the employee's hire date (%%1) is after the period end date (%%2) for the current payroll. Pay data was not imported for this employee. To include this employee's pay data, change the hire date in payroll and reimport the pay data.	Employee's hire date prior to payroll period end date.
34616	There aren't any employees that match the pay frequency (%%1) in this pay data file.	**Pay Frequency matches value in RUN Powered by ADP.**
34617	There aren't any employees that match the pay frequency (%%1) in this pay data file.	Pay Frequency matches value in RUN Powered by ADP.
35102	We couldn't import this employee's time because the employee's status is **Leave of Absence** in payroll.	Employee must be **Active** in RUN Powered by ADP.
35105	It looks like this employee has time included in the pay data file but has a **Terminated** status in payroll. If the employee's termination date is after the start date of the next payroll, then time will be imported for this employee.	Termination date on the **Employment Info** page must be earlier than pay period end date.
34622	It looks like the earnings code %%1 isn't correct and we couldn't import pay data for this earning.	Earning code matches value in RUN Powered by ADP.
34623	It looks like the earnings code %%1 isn't active in the payroll application, and the pay data can't be imported. Can you change %%1 to an active earning code?	Earning code must be active for the company on the **Company Earnings and Deductions** page.

34624	We couldn't import hours for this earning. It looks like the %%1 earning isn't valid for this employee, because the payroll application supports an amount, not hours, for this earning.	Earning code matches value in RUN Powered by ADP.
34626	It looks like department %%1 isn't valid and we couldn't import pay data for this department.	Department value matches value in RUN Powered by ADP.
34736	We couldn't import pay data for this earning because the %%1 earning is used for employees and not for contractors.	Employees using earning code defined for Employees in RUN Powered by ADP.
34637	We couldn't import pay data for this earning because the %%1 earning is used for contractors and not for employees.	Contractors using earning code defined for Contractors in RUN Powered by ADP.
34638	It looks like you don't have the correct number of hours for this earning and the hours weren't imported. The %%1 hour earning must be from %%2 to %%3.	Valid number of hours according to range defined in RUN Powered by ADP.
34639	It looks like the %%1 earnings amount isn't correct and it wasn't imported. The %%1 earnings amount must be from %%2 to %%3.	Valid amount for the earning according to range defined in RUN Powered by ADP.
34640	We couldn't import pay data for the separate check %%1.	**Use additional checks instead.** **Note:** **Up to nine additional checks for an employee in the payroll application are supported.**
34656	We couldn't import pay data for the rate code %%1.	Rate code value matches value in RUN Powered by ADP.
34680	We couldn't import pay data for the %%1 rate %%2. To pay the employee, enter the correct rate in the **Payroll Worksheet**.	Valid rate value according to range defined in RUN Powered by ADP.

Issue 2: Wrong error text

Description

Wrong response error text is returned for certain exception conditions.

	Condition	**Expected error message**	**Actual error message**
1	Invalid number of hours.	It looks like you don't have the correct number of hours for this earning and the hours weren't imported. The %%1 hour earning must be from %%2 to %%3.	**numberOfHours:The field numberOfHours must be between 0 and 9999.99**
2	Invalid earning amount.	It looks like the %%1 earnings amount isn't correct and it wasn't imported. The %%1 earnings amount must be from %%2 to %%3.	**RateValue:The field RateValue must be between 0 and 9999999.99**
3	Invalid Pay Number.	Pay data could not be synched for a separate check.	**PayNumber:The field PayNumber must match the regular expression '^[0-9]$**

Impacted APIs

Method	URI	roleCode Value
POST	/events/payroll/v1/pay-data-input.add	Practitioner
GET	/events/payroll/v1/pay-data-input.add/meta	Practitioner
POST	/events/payroll/v1/pay-data-input.replace	Practitioner
GET	/events/payroll/v1/pay-data-input.replace/meta	Practitioner

Suggested Work Around

To avoid this exception, build validation logic during the data collection process in your application prior to sending over to ADP.

Issue 3: 504(Gateway timeout) issue

504 as the API is taking more time to process during the large client base.

Method	URI	roleCode Value
POST	/events/payroll/v1/pay-data-input.add	Practitioner
POST	/events/payroll/v1/pay-data-input.replace	Practitioner
GET	/payroll/v1/pay-data-input	Practitioner

Suggested Work Around

You get a 504 as the API is taking more time to process during the large client base. Though you get a 504 error the data will be reflected in the RUN UI. You can verify whether the data is added or replaced by making the above get call.

There is nothing in implementation on this issue as of now. There are very unique cases where there are more than 100 employees for the clients present under RUN.As of now you implementing the GET Call would be the fastest and best resolution when you get a 504 error for this API .All the Pay data input requests to RUN are successful for the client. The API has a time limit SLA limit of 12 seconds ,but the client is receiving time out response as its exceeding the maximum time limit set (12 seconds)where as the request is still in process at RUN and gets successful. Due to the no of line items to process it was taking more than 12 seconds and is timing out which in turn is causing the 504 error.

The suggestion here is to make a call after 1min to get the successful response for the get call.

Note: There's an existing defect (SBSRUNCORE-28047), that's being created to handle this issue.

Chapter 8

Appendix: Time Sheet Import Errors and To-Do's

We recommend you to make your user aware of the following practice:

After using your integrated application to synch Time Sheet data, a client payroll practitioner should review and resolve errors listed in the RUN Powered by ADP UI under **Home > Time Sheet > Time Sheet Import > Select the Pay Frequency**.

The following chart provides information on how to fix the errors the practitioners might encounter when using your application that leverages the Payroll Data Input APIs. Errors are listed in the order of the error number. Information that changes depending on the message, such as a company code, pay frequency, or employee identifier is represented using the %% symbol. For each error, the data in your application should be fixed and synchronization between your application and RUN Powered by ADP should be reprocessed.

Error Number	Message	What to do next
Error Number	Message	What to do next
34611	It looks like the pay frequency (%%1) in your pay data import file doesn't match the pay frequency of this payroll. Make sure you selected the correct time sheet file for this pay frequency.	If this is the correct file, then check the **Pay Frequency** page under the **Company** tab and, if needed, add the correct pay frequency. If you continue to get this error, contact your ADP Service team.
34612	The pay period start and end dates (%%1 - %%2) in payroll don't match the pay period start and end dates (%%3 - %%4) in your time and attendance system for this payroll.	Make sure you imported the current pay data import file or correct the pay periods in time and attendance. We will use (%%3-%%4) for this payroll.
34613	It looks like your pay data import file contains an employee (%%1) that does not exist in your payroll. Pay data was not imported for this employee.	To pay this employee, verify you have the correct employee number or try hiring the employee in the payroll application and entering the hours on the **Payroll Worksheet**.
34614	We noticed that the employee's hire date (%%1) is after the period end date (%%2) for the current payroll. Pay data was not imported for this employee. To include this employee's pay data, change the hire date in payroll and reimport the pay data.	To pay this employee, verify you have the correct employee number or try hiring the employee in the payroll application and entering the hours on the **Payroll Worksheet**.
34616	There aren't any employees that match the pay frequency (%%1) in this pay data file. Make sure you have the correct file or make any needed changes to your employees' pay frequency on the **Payroll Info** page and import the pay data file again.	If you still get this error, contact your ADP Service team.
34617	There aren't any employees that match the pay frequency (%%1) in this pay data file. Make sure you have the correct file or make any needed changes to your employees' pay frequency on the **Payroll Info** page and import the pay data file again.	If you still get this error, contact your ADP Service team.
35102	We couldn't import this employee's time because the employee's status is **Leave of Absence** in payroll.	Try changing the status to **Active** in the payroll application on the **Employment Info** page. Then, enter the employee's hours on the **Payroll Worksheet**.
35105	It looks like this employee has time included in the pay data file but has a **Terminated** status in payroll. If the employee's termination date is after the start date of the next payroll, then time will be imported for this employee.	If pay data wasn't imported and you want to pay the employee, try changing the termination date on the **Employment Info** page so it is during or after the pay period dates, and update the employee's hours on the **Payroll Worksheet**. If you don't want to pay the employee, then delete the pay data on the **Payroll Worksheet**.
34622	It looks like the earnings code %%1 isn't correct and we couldn't import pay data for this earning.	Contact your ADP Service team for assistance with this earning.

34623	It looks like the earnings code %%1 isn't active in the payroll application, and the pay data can't be imported. Can you change %%1 to an active earning code?	Make sure the earning is set up and active for the company on the **Company Earnings and Deductions** page and enter pay data for the employee on the **Payroll Worksheet**.
34624	We couldn't import hours for this earning. It looks like the %%1 earning isn't valid for this employee, because the payroll application supports an amount, not hours, for this earning.	Try updating the earning on the **Company Earnings and Deductions** page in the payroll application or contact your ADP Service team.
34626	It looks like department %%1 isn't valid and we couldn't import pay data for this department.	Try setting up the department in RUN Powered by ADP and assigning it to the employee. Then, manually add the department to your time and attendance system.
34736	We couldn't import pay data for this earning because the %%1 earning is used for employees and not for contractors.	To pay the contractor in the current payroll, enter pay data on the **Payroll Worksheet**. For the next payroll, assign a different earnings code in the time card, or contact your ADP Service team to set up a new earnings code.
34637	We couldn't import pay data for this earning because the %%1 earning is used for contractors and not for employees.	To pay the employee in the current payroll, enter pay data on the **Payroll Worksheet**. For the next payroll, assign a different earnings code in the time card, or contact your ADP Service team to set up a new earnings code.
34638	It looks like you don't have the correct number of hours for this earning and the hours weren't imported. The %%1 hour earning must be from %%2 to %%3.	Try entering the correct number of hours on the **Payroll Worksheet**. You can also correct the earnings hours in your time and attendance system, regenerate the time sheet import file, and import it into the payroll application again.
34639	It looks like the %%1 earnings amount isn't correct and it wasn't imported. The %%1 earnings amount must be from %%2 to %%3.	Try entering the correct amount for the earning on the **Payroll Worksheet**. You can also correct the earnings hours in your time and attendance system, regenerate the time sheet import file, and import it into the payroll application again.
34640	We couldn't import pay data for the separate check %%1.	Try creating an additional check for the employee on the **Payroll Worksheet**. You can enter up to nine additional checks for an employee in the payroll application.
34656	We couldn't import pay data for the rate code %%1.	Contact your ADP Service team for assistance with this error.
34680	We couldn't import pay data for the %%1 rate %%2. To pay the employee, enter the correct rate in the **Payroll Worksheet**.	For the next payroll, enter the rate on the **Payroll Info** page in the payroll application. If you need to correct the rate configuration in your time and attendance system, contact your ADP Service team.