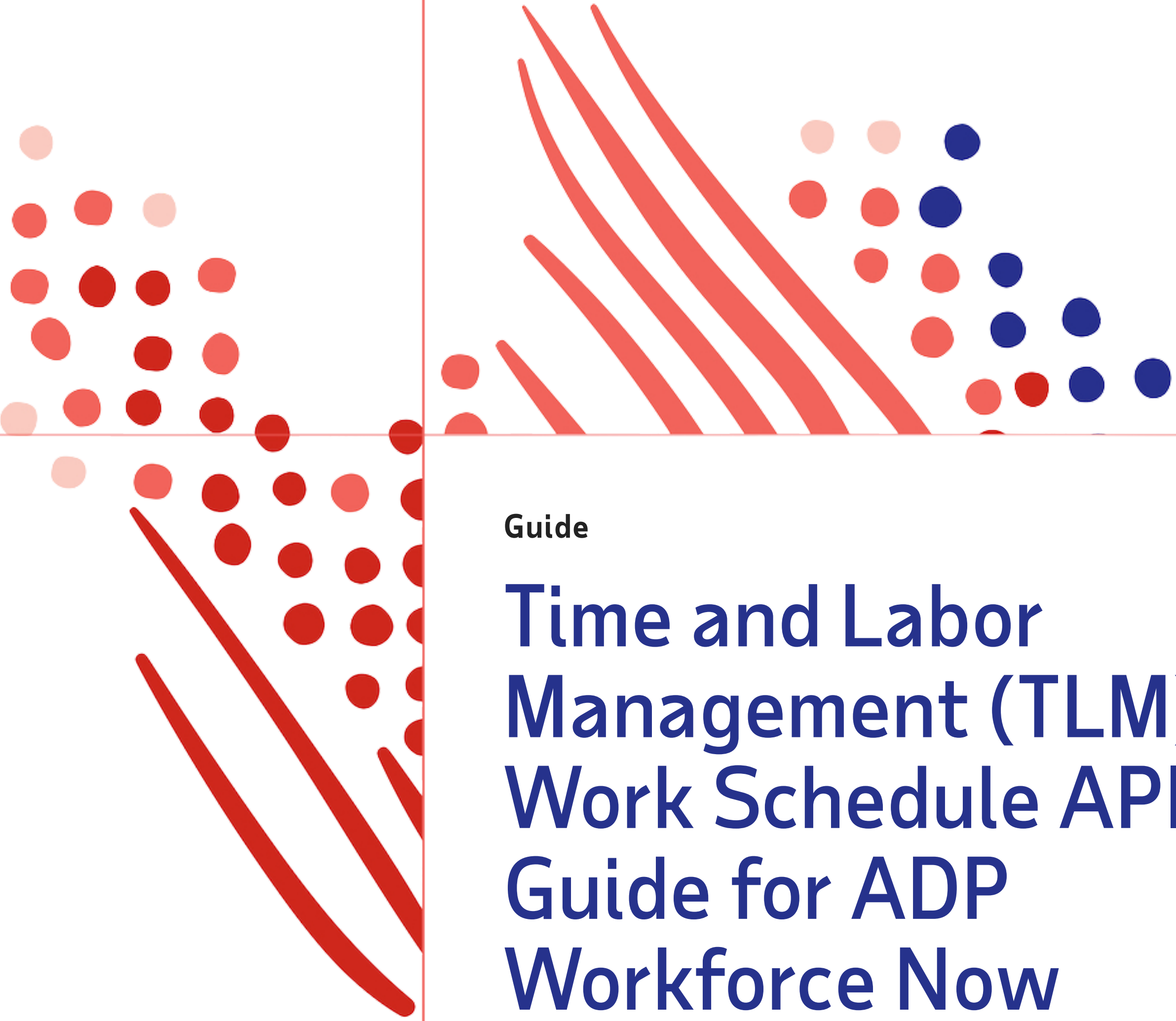


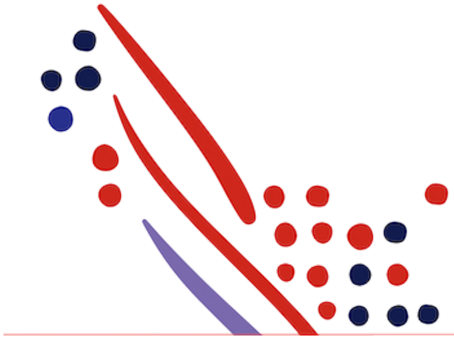
Guide

Time and Labor Management (TLM) Work Schedule API Guide for ADP Workforce Now

Published on
Aug 06, 2020 12:09PM

Last modified
Aug 04, 2022 6:13AM

Always Designing
for People™



ADP Copyright Information

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
Aug 06, 2020 12:09PM

Last modified
Aug 04, 2022 6:13AM

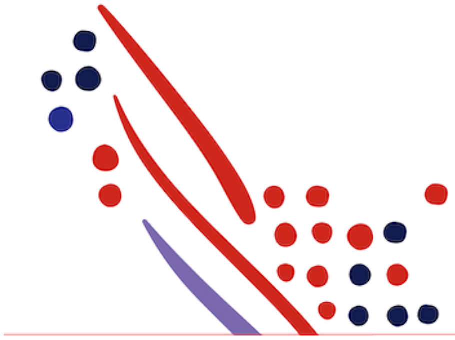


Table of Contents

Description

Suggested Workaround

Chapter 1

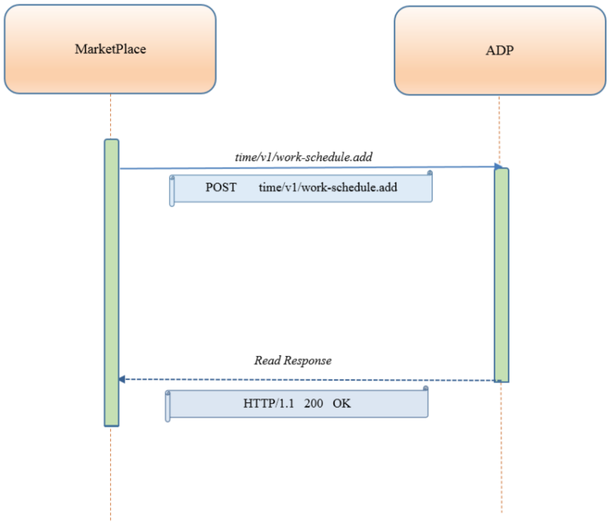
About this API

Summary

The TLM Work Schedule Application Programming Interface (API) is used to add a work schedule for an employee. This data includes an Employee Positions schedule.

Most commonly, a work schedule will be for one associate and for multiple days. Each day can have multiple shifts.
The TLM Work Schedule API will accept and try to process all shifts. If any of the shifts fail, only the specific shift fails, the other shifts will be processed. The details are provided in the response on why the shift add failed. In response, the shifts processed successfully will have a generated **scheduleEntryId** assigned.

The following is an illustration of how the TLM Work Schedule API functions:



What's New in this Guide?

Updated API endpoints and samples to link to API explorer.

Supported Product Version and Customer Base

The TLM Work Schedule API is supported by the latest version of ADP Workforce Now with the essential Time module.

Required Setup Steps

A set of Employee Positions configured to use Time must be available before the TLM Work Schedule API can be used.

Chapter 2

Use Case: Adding a Work Schedule

Use Case Description

This use case adds an employee work schedule (shifts) for a specific date range. For example, employee shifts for one or two weeks.

guide link : Work Schedule API Guide for ADP Workforce Now

API Usage

Method	Uniform Resource Indicator (URI)	Description
POST	/events/time/v1/ work-schedule.add/	Specifies the API used to add a work schedule
GET	/events/time/v1/work-schedule.add/meta	Returns an event metadata for your application to build the payload.

Application Scope

The canonical uniform resource identifier (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:
/time/timeLaborManagement/scheduleManagement/workScheduleManagement/workSchedule.add

Supported Actors

Request Parameter roleCode Value	Usage
Practitioner	Add a schedules for an employee.

Request Header Parameters

Parameter Name	Required (Y/N)	Usage	Value	Sample
Content-Type	Y		application/json	**Content-Type: application/json**

Other Supported Parameters

There are no other supported parameters.

Sequence of Interactions

The sequence of interactions is as follows:

1. Your consumer application makes a request to the ADP API endpoint for **work-schedule.add**.
2. The ADP endpoint responds to your consumer application with a response.

Data Dictionary

The resources listed can be accessed from the schema locations shown on the following table.

Schema Location	Field Name	Is Required (Y/N)	Note
/events/data/transform/workSchedule/schedulePeriod/ startDate		Y	The start date on the work schedule.
/events/data/transform/workSchedule/schedulePeriod/ endDate		Y	The end date on the work schedule.
/events/data/transform/workSchedule/scheduleDays/scheduleEntries/dateTimePeriod/startDateTime	**INTIME**	Y	The string representation of the start date-time value and follows the ISO-8601:2000 format. For example: **2019-04-16T08:00:00.000-04:00**
/events/data/transform/workSchedule/scheduleDays/scheduleEntries/dateTimePeriod/endDateTime	**OUTTIME**	Y	The string representation of the start date-time value and follows the ISO-8601:2000 format. For example: **2019-04-16T16:00:00.000-04:00**
/events/data/transform/workSchedule/scheduleDays/scheduleDayDate		Y	The day on which the operation is performed.
/events/data/transform/workSchedule/scheduleDays/scheduleEntries/categoryTypeCode/codeValue		N	The allowed values for shift type are **SHIFT** and **PAYCODE**.

/events/data/transform/workSchedule/scheduleDays/scheduleEntries/payCode/codeValue	**PAYCODE**	N	This shows if the schedule entry has any **payCode**.
/events/data/eventContext/ associateOID	**ASSOCIATEOID**	Y	This is the Associate Object ID (AOID) to whom the schedule entry belongs.
/events/data/transform/workSchedule/scheduleDays/scheduleEntries/entryComments/ textValue	**NOTE**	N	This is the schedule entry comments/notes if there are any.
/events/data/transform/workSchedule/scheduleDays/scheduleEntries/entryComments/commentTypeCode/codeValue			The reason code for a note.
/events/data/transform/workSchedule/scheduleDays/scheduleEntries/earningAllocations/allocationTypeCode/codeValue		N	Shows if the schedule entry has any earnings allocations assigned. For example: Department, Job, Meal Plan, ShiftRule, FlexitimeRule, and so on. Each shift can only have one Department, Job, Meal Plan, ShiftRule, FlexitimeRule, and so on assigned.
/events/data/transform/workSchedule/scheduleDays/scheduleEntries/earningAllocations/allocationCode/codeValue		N	This is the valid code for respective earning allocation type of Code. **{ "allocationTypeCode": { "code": "Department" }, "allocationCode": { "code": "003000" } }**
/events/data/output/workSchedule/scheduleDays/scheduleEntries/scheduleEntryID	**SCHEDULEOID**	Response	This is the Generated Schedule OID.
confirmMessage/requestStatusCode/codeValue			The **requestStatus** can have following values: • **failed** • **partiallyFailed** • **succeeded**
confirmMessage/resourceMessages/resourceMessageID			This identifies the shift and is formed using **ASSOCIATEOID_DATE_INTIME_ OUTTIME** For example: **G3ANQVTXCE1ZBV65_2019-05-10_09:00:00_12:00:00**
confirmMessage/resourceMessages/processMessages/userMessage/ codeValue			The Error Code. For example, **exp_InvalidRequest**.

confirmMessage/resourceMessages/processMessages/userMessage/ messageTxt			This is the description of the error. For example: **Invalid Request**
confirmMessage/resourceMessages/processMessages/ messageTypeCode/codeValue			This is the status of the schedule entry. For example: **ERROR**

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.

guide link : Work Schedule API Guide for ADP Workforce Now

Response Code	Condition	Tips to Handle
200 OK	When the request is successful.	The **RequestStatus** can have the following values in the response: • **failed** • **partiallyFailed** • **succeeded**
200 OK	When the request is unsuccessful.	
200 OK	When the request is partially **Failed**.	
200 OK	When the **Invalid allocation** is passed in the request.	
200 OK	When the **Invalid payCode** is passed in the request.	
400 Bad Request	When the **Invalid Associate ID** is passed in the request.	
500 Internal Server Error	When the **Invalid startDateTime** is passed in the request.	
500 Internal Server Error	When Invalid **endDateTime** is passed in the request.	

Chapter 3

Use Case: Changing a Work Schedule

Use Case Description

This use case changes an employee work schedule (shifts) for a specific date range. For example, employee shifts for one or two weeks.

guide link : Work Schedule API Guide for ADP Workforce Now

API Usage

Method	URI	Description
POST	/events/time/v1/work-schedule.change/	Specifies the API used to change the work schedule.
GET	/events/time/v1/work-schedule.change/meta	Returns an event metadata for your application to build the POST payload.

Application Scope

The following canonical needs to be added to your application scope to enable this use case: **/time/timeLaborManagement/scheduleManagement/workScheduleManagement/workSchedule.change**

Supported Actors

Request Parameter roleCode Value	Usage
Practitioner	Add a schedule for an employee.

Request Header Parameters

Parameter Name	Required (Y/N)	Usage	Value	Sample
Content-Type	Y		application/json	**Content-Type: application/json**

Other Supported Parameters

There are no other supported parameters.

Sequence of Interactions

The sequence of interactions is as follows:

1. Your consumer application makes a request to the ADP API endpoint for **work-schedule.add**.
2. The ADP endpoint responds to your consumer application with a response.

Data Dictionary

The resources listed can be accessed from the schema locations shown on the following table.

Schema Location	Field Name	Is Required (Y/N)	Note
/events/data/transform/workSchedule/schedulePeriod/startDate		Y	This is the start date on the work schedule to be modified.
/events/data/transform/workSchedule/schedulePeriod/endDate		Y	This is the start date on the work schedule to be modified.
/events/data/transform/workSchedule/scheduleDays/scheduleEntries/ dateTimePeriod/startDateTime	**INTIME**	Y	The string representation of the start date-time value and follows the ISO-8601:2000 format. For example, **2019-04-16T08:00:00.000-04:00**
/events/data/transform/workSchedule/scheduleDays/scheduleEntries/ dateTimePeriod/endDateTime	**OUTTIME**	Y	The string representation of the start date-time value and follows the ISO-8601:2000 format. For example: **2019-04-16T16:00:00.000-04:00**
/events/data/transform/workSchedule/scheduleDays/scheduleDayDate		Y	The day on which the operation is performed.
/events/data/transform/workSchedule/scheduleDays/scheduleEntries/ categoryTypeCode/codeValue		N	The allowed values for shift type are **SHIFT** and **PAYCODE**.
/events/data/transform/workSchedule/scheduleDays/scheduleEntries/ payCode/codeValue	**PAYCODE**	N	Shows if schedule entry has any **payCode**.
/events/data/eventContext/ associateOID	**ASSOCIATEOID**	Y	This is the AOID to whom the schedule entry belongs.
/events/data/transform/workSchedule/scheduleDays/scheduleEntries/ entryComments/textValue	**NOTE**	N	This is the schedule entry comments/notes, if any.
/events/data/transfor			This is the reason code for a note.

m/workSchedule/scheduleDays/scheduleEntries/entryComments/commentTypeCode/codeValue			
/events/data/transform/workSchedule/scheduleDays/scheduleEntries/earningAllocations/allocationTypeCode/codeValue		N	Shows if the schedule entry has any earnings allocations assigned. For example: Department, Job, Meal Plan, ShiftRule, FlexitimeRule, and so on. Each shift can only have one Department, Job, Meal Plan, ShiftRule, FlexitimeRule, and so on assigned.
/events/data/transform/workSchedule/scheduleDays/scheduleEntries/earningAllocations/allocationCode/codeValue		N	This is the valid code for the respective earning allocation type code.
/events/data/transform/workSchedule/scheduleDays/scheduleEntries/ scheduleEntryID	**SCHEDULEOID**	Y	This is the Schedule OID to be modified.
confirmMessage/requestStatusCode/codeValue			The **requestStatus** can have following values • **failed** • **partiallyFailed** • **succeeded**
confirmMessage/resourceMessages/resourceMessageID			This is the **scheduleEntryID** from the request identifying the individual shift.
confirmMessage/resourceMessages/processMessages/userMessage/ codeValue			This is the error code. For example: **exp_InvalidRequest**
confirmMessage/resourceMessages/processMessages/userMessage/ messageTxt			This is the description of error. For example: **Invalid Request**
confirmMessage/resourceMessages/processMessages/messageTypeCode/codeValue			This is the status of the schedule entry. For example: **ERROR**

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.

guide link : Work Schedule API Guide for ADP Workforce Now

Response Code	Condition	Tips to Handle
200 OK	When the request is successful.	More details will be in the body of the Response. The **requestStatus** can have following values: • **failed** • **partiallyFailed** • **succeeded**
200 OK	When **Invalid scheduleEntry ID** is passed and the request is failed.	
200 OK	When the **Invalid payCode** is passed and the request is **partialFailed**.	
200 OK	When the **Invalid department number** is passed, and the request is **partialFailed**.	
400 Bad Request	When **Invalid associate ID** is passed in the request.	
500 Internal Server Error	When **Invalid startDateTime** is passed in the request.	
500 Internal Server Error	When **Invalid endDateTime** is passed in the request.	

Chapter 4

Use Case: Removing a Work Schedule

Use Case Description

This use case removes an employee work schedule (shifts) for a specific date range. For example, employee shifts for one or two weeks.

guide link : Work Schedule API Guide for ADP Workforce Now

API Usage

Method	URI	Description	
POST	/events/time/v1/ work-schedule.remove	Specifies the API used to remove the work schedule.	
GET	/events/time/v1/work-schedule.remove/meta	Returns an event metadata to build the payload for the POST method.	**Important:** Meta API is currently not working.

Application Scope

The following canonical needs to be added to your application scope to enable this use case: **/time/timeLaborManagement/scheduleManagement/workScheduleManagement/ workSchedule.remove**

Supported Actors

Request Parameter roleCode Value	Usage
Practitioner	Adds a schedule for an employee.

Request Header Parameters

Parameter Name	Required (Y/N)	Usage	Value	Sample
Content-Type	Y		application/json	**Content-Type: application/json**

Other Supported Parameters

There are no other supported parameters.

Sequence of Interactions

The sequence of interactions is as follows:

1. Your consumer application makes a request to the ADP API endpoint for **work-schedule.add**.
2. The ADP endpoint responds to your consumer application with a response.

Data Dictionary

The resources listed can be accessed from the schema locations shown on the following table.

Schema Location	Field Name	Is Required (Y/N)	Note
Schema Location	**Field Name**	**Is Required (Y/N)**	**Note**
events/data/eventContext/ associateOID	**ASSOCIATEOID**	Y	This is an AOID to whom the schedule entry belongs.

events/data/eventContext/ scheduleID		Y	This is the start and end date from which schedule is to be deleted. The format is the following: **StartDatezzEndDate** For example: **2019-05-10zz2019-05-14**
confirmMessage/ requestStatusCode/codeValue			The **requestStatus** can have following values: • **failed** • **partiallyFailed** • **succeeded**
confirmMessage/ resourceMessages/ processMessages/userMessage/ codeValue			The Error Code. For example: **exp_InvalidRequest**
confirmMessage/ resourceMessages/ processMessages/userMessage/ messageTxt			This is the description of the error. For example: **Invalid Request**
confirmMessage/ resourceMessages/ processMessages/ messageTypeCode/codeValue			This is the status of schedule entry. For example: **ERROR**

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.

guide link : Work Schedule API Guide for ADP Workforce Now

Response Code	Condition	Tips to Handle
200 OK	When the request is successful.	More details will be in the Response body. The **requestStatus** can have following values: • **failed** • **partiallyFailed** • **succeeded**
200 OK	When **Invalid date** is passed in the request.	
200 OK	When **Invalid associate ID** is passed in the request.	

Chapter 5

Use Case: Copying a Work Schedule

Use Case Description

This use case copies a work schedule from one associate to another. The copy is done from **startDateCopyTo** and consecutive dates.

guide link : Work Schedule API Guide for ADP Workforce Now

API Usage

Method	**URI**	**Description**	
POST	/events/time/v1/work-schedule.copy/	Specifies the API used to remove the work schedule.	
GET	/events/time/v1/work-schedule.copy/meta	Returns an event metadata to build the payload for the POST method.	**Important:** Meta API is currently not working.

Application Scope

The following canonical needs to be added to your application scope to enable this use case: **/time/timeLaborManagement/scheduleManagement/workScheduleManagement/workSchedule.copy**

Supported Actors

Request Parameter roleCode Value	**Usage**
Practitioner	Adds a schedule for an employee.

Request Header Parameters

Parameter Name	**Required (Y/N)**	**Usage**	**Value**	**Sample**
Content-Type	Y		application/json	**Content-Type: application/json**

Other Supported Parameters

There are no other supported parameters.

Sequence of Interactions

The sequence of interactions is as follows:

1. Your consumer application makes a request to the ADP API endpoint for **work-schedule.add**.
2. The ADP endpoint responds to your consumer application with a response.

Data Dictionary

The resources listed can be accessed from the schema locations shown on the following table.

Schema Location	Field Name	Is Required (Y/N)	Note
Schema Location	**Field Name**	**Is Required (Y/N)**	**Note**
/events/data/transform/workSchedule/scheduleDays/scheduleEntries/dateTimePeriod/startDateTime	**INTIME**	Y	The string representation of the start date-time value and follows the ISO-8601:2000 format. For example: **2019-04-16T08:00:00.000-04:00**
/events/data/transform/workSchedule/scheduleDays/scheduleEntries/dateTimePeriod/endDateTime	**OUTTIME**	Y	The string representation of the start date-time value and follows the ISO-8601:2000 format. For example: **2019-04-16T16:00:00.000-04:00**
/events/data/transform/workerCopyTo/associateOID		Y	This is the AOID to which the work schedule should be copied.
/events/data/transform/ startDateCopyTo		Y	This is the start date from which the copy should take place.
/events/data/transform/numberOfRecurrences		N	This is always 1.
/events/data/transform/workSchedule/scheduleDays/scheduleDayDate		Y	The day on which operation is performed.
/events/data/transform/workSchedule/scheduleDays/scheduleEntries/categoryTypeCode/codeValue		N	The allowed shift type values are **SHIFT** and **PAYCODE**.
/events/data/transfor	**PAYCODE**	N	Shows if schedule entry has

m/ workSchedule/scheduleDays/ scheduleEntries/payCode/codeValue			any **payCode**.
/events/data/eventContext/ associateOID	**ASSOCIATEOID**	Y	This is the AOID to whom schedule entry belongs.
/events/data/transform/ workSchedule/scheduleDays/ scheduleEntries/entryComments/ textValue	**NOTE**	N	This is the schedule entry comments/notes, if any.
/events/data/transform/ workSchedule/scheduleDays/ scheduleEntries/entryComments/ commentTypeCode/codeValue			This is the reason code for a note.
/events/data/transform/ workSchedule/scheduleDays/ scheduleEntries/earningAllocations/ allocationTypeCode/codeValue		N	Shows if the schedule entry has any earnings allocations assigned. For example: Department, Job, Meal Plan, ShiftRule, FlexitimeRule, and so on. Each shift can only have one Department, Job, Meal Plan, ShiftRule, FlexitimeRule, and so on assigned.
/events/data/transform/ workSchedule/scheduleDays/ scheduleEntries/earningAllocations/ allocationCode/codeValue		N	This is the valid code for the respective earning allocation type code.
/events/data/transform/ workSchedule/scheduleDays/ scheduleEntries/scheduleEntryID		Y	This is the source Work Schedule Entry OID.
/events/data/output/ workSchedule/ scheduleDays/scheduleEntries/ scheduleEntryID	**SCHEDULEOID**	Response	This is the generated Schedule OID.
confirmMessage/requestStatusCode/codeValue			The **requestStatus** can have following values: • **failed** • **partiallyFailed** • **succeeded**
confirmMessage/resourceMessages/resourceMessageID			Identifies the shift formed using **ASSOCIATEOID_DATE_INTIME_OUTTIME**.

			For example: **G3ANQVTXCE1ZBV65_2019-05-10_09:00:00_12:00:00**
confirmMessage/resourceMessages/processMessages/userMessage/ codeValue			This is the error code. For example: **exp_InvalidRequest**
confirmMessage/resourceMessages/processMessages/userMessage/ messageTxt			This is the description of the error. For example: **Invalid Request**
confirmMessage/resourceMessages/processMessages/ messageTypeCode/codeValue			This is the status of schedule entry. For example: **ERROR**

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.

guide link : Work Schedule API Guide for ADP Workforce Now

Response Code	Condition	Tips to Handle
200 OK	When the request is to copy the day shift to another day for the same employee.	More details will be in the Response body. The **requestStatus** can have following values: - **failed** - **partiallyFailed** - **succeeded**
200 OK	When the request is to copy the day shift to another day for a different employee.	
200 OK	When the request is to copy the day shift, but the shift overlaps.	
200 OK	When the **Invalid allocation** is passed in the request.	
200 OK	When the **Invalid payCode** is passed in the request.	
400 Bad Request	When the **Invalid associate ID** is passed in the request.	

Chapter 6

Use Case: Reading a Work Schedule

Use Case Description

The **Work-schedules HTTP GET** method is used to retrieve all employee schedules reporting to a supervisor who requested the TLM Work Schedule API.

guide link : Work Schedule API Guide for ADP Workforce Now

API Usage

Method	URI	Description
GET	/time/v1/work-schedules	Retrieves all schedules for employees reporting to the Practitioner.
GET	/time/v1/work-schedules?$filter=schedulePeriod/ startDate ge 'YYYY-MM-DD' and schedulePeriod/endDate le 'YYYY-MM-DD'	Retrieves all schedules for employees, with date range, reporting to the Practitioner.
GET	/time/v1/workers/{associateoid}/work-schedules	Retrieves requested employee work schedules.
GET	/time/v1/workers/G332EFV3Q6J0048K/ work-schedules?$filter=schedulePeriod/startDate ge 'YYYY-MM-DD' and schedulePeriod/endDate le 'YYYY-MM-DD'	Retrieves requested employee work schedules with date range.

Application Scope

The following canonical needs to be added to your application scope to enable this use case: **/time/timeLaborManagement/scheduleManagement/workScheduleManagement/workSchedule.read**

Request Header Parameters

Header Name	Sample Value	Comments
Accept	application/json	Accept: application/json

Data Dictionary

The resources listed can be accessed from the response locations shown on the following table.

Response Location	ADP Workforce Now Field Name	Note
Response Location	**ADP Workforce Now Field Name**	**Note**

/workSchedules/scheduleID		This is the date range for which schedules are returned in format **startdatezzenddate**. For example: **2019-08-02zz2019-08-08**
/workSchedules/associateOID		This is the AOID for whom schedules are returned.
/workSchedules/workerName/	**Supervisor/employee name details**	This is the AOID name.
/workSchedules/workerName/givenName	**employee/firstname**	This is the employee's first name.
/workSchedules/workerName/ familyName1	**employee/lastname**	This is the employee's last name.
/workSchedules/workerName/ formattedName	**employee/name**	This is the employee's last name, first name.
/workSchedules/schedulePeriod/		This is the date range for which schedules are returned and provides the requested start and end date of the schedule period.
/workSchedules/schedulePeriod/startDate		This field provides the requested start of the schedule period.
/workSchedules/schedulePeriod/endDate		This field provides the requested end date of the schedule period.
/workSchedules/scheduleDays		This field provides a list of scheduled days and each day shift details.
/workSchedules/scheduleDays/ scheduleDayDate		This is the date of the scheduled day.
/workSchedules/scheduleDays/ daySequenceNumber		This is the sequence of days within a week starting Sunday, which is **0** and Monday, which is **1**, and so on.
/workSchedules/scheduleDays/ scheduleEntries		These are the schedules for a day.
/workSchedules/scheduleDays/ scheduleEntries/actions/operationID		The **OperationID** indicates the schedule type. The following are the different **OperationIDs** allowed: • Deviation – Shift is the deviation to recurring or time off. It is also called the daily shift. • Recurring – Recurring shift. • Time off – Schedule type is time off.
/workSchedules/scheduleDays/ scheduleEntries/scheduleEntryID		This is the **Uniqueid** used to identify a shift.
/workSchedules/scheduleDays/scheduleEntries/dateTimePeriod/startDateTime		This field provides the details on **Schedule start date** and **intime**.

/workSchedules/scheduleDays/scheduleEntries/dateTimePeriod/endDateTime		This field provides the details on **Schedule end date** and **outtime**.
/workSchedules/scheduleDays/scheduleEntries/totalTime/timeValue	**Schedule total time value**	**Note:** For schedules with **lunchplan**, total time represents the total time after deducting lunch time from total scheduled hours.
/workSchedules/scheduleDays/scheduleEntries/totalTime/nameCode	**Schedule total time code name**	**Note:** The default and supported name code is 'hour'.
/workSchedules/scheduleDays/scheduleEntries/totalTime/nameCode/shortName	**Schedule total time short name**	This is always **Hours**. **Note:** Default and supported name code is 'hours'.
/workSchedules/scheduleDays/scheduleEntries/totalTime/nameCode/codeValue		This is always **hour**.
/workSchedules/scheduleDays/scheduledHours/hoursQuantity		This is the total amount of hours scheduled in that day. In a day if any employee has multiple shifts then **hoursQuantity** provides the total work hours, which combines all schedules.
/workSchedules/scheduleDays/appliedTemplateID		This field provides the scheduled template name. This field is populated only when shift is **Template shift**, which is also called a recurring shift.
/workSchedules/scheduleDays/scheduleEntries/earningAllocations/allocationTypeCode/codeValue	**Job** **Department** **ShiftRule** **LunchPlan** **FlexitimeRule**	This is the **allocationTypeCode** that represents the meta data for each of the allocation types. The fields listed are allowed allocation types for a schedule.
/workSchedules/scheduleDays/scheduleEntries/earningAllocations/allocationTypeCode/shortName		This is the description details for the **allocationTypeCode**.
/workSchedules/scheduleDays/scheduleEntries/earningAllocations/allocationCode/codeValue		This is the value for the **allocationTypeCode**. For example: The **alloctionTypeCode** of **LunchPlan** value is **30ACTL**.
/workSchedules/scheduleDays/scheduleEntries/earningAllocations/allocationCode/shortName		This is the description details for the **allocationCode**. For example: The **allocationCode** of **30ACTL** is **30-Min Always Punch – Actual**.
/workSchedules/scheduleDays/scheduleEntries/categoryTypeCode		This indicates the shift type category. The following are allowed category types: • **SHIFT** – Shift without paycode. • **PAYCODE** – Shift with paycode.

/workSchedules/scheduleDays/scheduleEntries/payCode/codeValue		This is the **payCode** associated to the schedule
/workSchedules/scheduleDays/scheduleEntries/payCode/shortName		This is the **payCode** description.

Responses

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

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

Response Code	Condition	GitHub Sample Request Payload
200 OK	Successful request call for multiple employees.	NA
200 OK	Successful request call for multiple employees with date range parameter in the end point.	NA
200 OK	Successful request call for single employee.	NA
200 OK	Successful request call for single employee with date range parameter in the end point.	NA

Chapter 7

Known Issues and Limitations

Issue 1: Meta API Not Working

Impacted APIs

Method	URI	roleCode Value
POST	/events/time/v1/work-schedule.add/	Practitioner
GET	/events/time/v1/work-schedule.add/meta	Practitioner
POST	/events/time/v1/work-schedule.change/	Practitioner
GET	/events/time/v1/work-schedule.change/meta	Practitioner

POST	/events/time/v1/work-schedule.remove	Practitioner
GET	/events/time/v1/work-schedule.remove/meta	Practitioner
POST	/events/time/v1/work-schedule.copy	Practitioner
GET	/events/time/v1/work-schedule.copy/meta	Practitioner

Description

The meta is returning the following error responses for work schedule day:

- ADD APIs
- Change APIs
- Remove APIs
- Copy APIs

```
{
"confirmMessage": {
"createDateTime": "2019-07-26T06:16:47-04:00",
"requestStatusCode": {
"codeValue": "failed"
},
"requestMethodCode": {
"codeValue": "GET"
},
"processMessages": [
{
"messageTypeCode": {
"codeValue": "ERROR"
},
"userMessage": {
"codeValue": "err_NoEmployeePositionFound",
"messageTxt": "err_NoEmployeePositionFound"
}
}
]
}
}
```

Suggested Workaround

There are no workarounds available.

Issue 2: Work Schedule Read API for a Single Employee Ignoring an Invalid Associate in the End Point and Returning Employees' Information

Impacted APIs

Method	URI	roleCode Value
GET	/time/v1/workers/{aoid}/work-schedules	Practitioner

Description

For the Work Schedule, Read API (**/time/v1/workers/{aoid}/work-schedules**), ignoring the invalid associate ID in the end point and returning multiple employees' information in the response.

For example:
End Point: /time/v1/workers/Invalid/work-schedules

Suggested Workaround

There are no workarounds available.

Issue 3: Currently, Work Schedule API doesn't Support Multiple Employee Positions

Impacted APIs

Method	URI	roleCode Value
POST	/events/time/v1/work-schedule.add/	Practitioner
GET	/events/time/v1/work-schedule.add/meta	Practitioner
POST	/events/time/v1/work-schedule.change/	Practitioner
GET	/events/time/v1/work-schedule.change/meta	Practitioner
POST	/events/time/v1/work-schedule.remove	Practitioner
GET	/events/time/v1/work-schedule.remove/meta	Practitioner

POST	/events/time/v1/work-schedule.copy	Practitioner
GET	/events/time/v1/work-schedule.copy/meta	Practitioner

Description

Multiple position employee is returning the following error responses for the Work Schedule Day:

- ADD APIs
- Change APIs
- Remove APIs
- Copy APIs

```
{
"confirmMessage": {
"createDateTime": "2019-09-05T02:13:28-04:00",
"protocolStatusCode": {
"codeValue": "400"
},
"requestStatusCode": {
"codeValue": "failed"
},
"requestMethodCode": {
"codeValue": "POST"
},
"processMessages": [
{
"messageTypeCode": {
"codeValue": "ERROR"
},
"userMessage": {
"codeValue": "exp_MultiplePositionsFound",
"messageTxt": "You have more than one position configured for mobile access which is not supported at
this time. Please contact your company administrator to correct this setting."
}
}
]
}
}
```

Suggested Workaround

There are no workarounds available.

Issue 4: State of Record (SOR) Returns 200 OK as a Status Code When the Request is Partially or Completely Failed

Impacted APIs

Method	URI	roleCode Value
POST	/events/time/v1/work-schedule.add/	Practitioner
GET	/events/time/v1/work-schedule.add/meta	Practitioner
POST	/events/time/v1/work-schedule.change/	Practitioner
GET	/events/time/v1/work-schedule.change/meta	Practitioner
POST	/events/time/v1/work-schedule.copy	Practitioner
GET	/events/time/v1/work-schedule.copy/meta	Practitioner

Description

The TLM Work Schedule API returns a 200 OK status code when there are partial and complete failures in the response.
For example:

- When the work schedule overlaps partially or completely, the response describes the failure, but the status code remains 200 OK.
- When the **payCode** number and allocation number are **Invalid**, the response describes the failure, but the status code remains 200 OK.

Suggested Workaround

There are no workarounds available.

Issue 5: Event Notifications Not Supported for Work Schedule Day APIs

Impacted APIs

Method	URI	roleCode Value
POST	/events/time/v1/work-schedule.add/	Practitioner
GET	/events/time/v1/work-schedule.add/meta	Practitioner
POST	/events/time/v1/work-schedule.change/	Practitioner
GET	/events/time/v1/work-schedule.change/meta	Practitioner
POST	/events/time/v1/work-schedule.remove	Practitioner
GET	/events/time/v1/work-schedule.remove/meta	Practitioner
POST	/events/time/v1/work-schedule.copy	Practitioner
GET	/events/time/v1/work-schedule.copy/meta	Practitioner

Description

Event Notifications are not supported.

Suggested Workaround

There are no workarounds available.