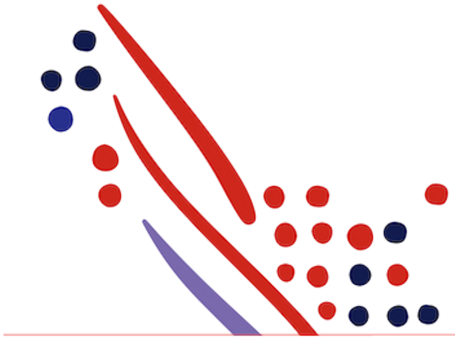


Chapter 2

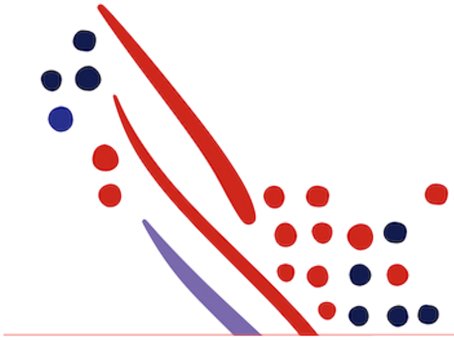
Use Case: Creating and Updating a Benefit Spending Account Plan

from Benefit Spending Account Plan API Guide (HROpen) for ADP Workforce Now Guide

Published on
May 05, 2020 3:30PM

Last modified
Jun 08, 2023 2:30PM

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
May 05, 2020 3:30PM

Last modified
Jun 08, 2023 2:30PM

Chapter Contents

Chapter 2

Use Case: Creating and Updating a Benefit Spending Account Plan

Use Case Description

- The Benefit Account Plan API allows the external benefit provider to create and update an in-progress spending account plan in ADP Workforce Now.
- After the API is successful, the in-progress plan can be viewed and accessed by Practitioners in ADP Workforce Now by selecting **Setup > Benefits > Plan Setup**.

The Benefit Account Plan API allows the external benefit provider to update the plan even after reviewed and approved by the ADP Workforce Now Practitioner.

Important:

While updating an already existing plan make sure that you don't change the following details of the payload. They should match with the details which are provided while creating the Benefit Spending Account Plan.

- **id/value**
- **providerId/value**
- **productCode**

In case any one of these is changed when compared to the one's that are used while creating the plan, a new plan gets created in ADP Workforce Now.

API Editor guide link : Spending Accounts Plan For ADP WFN

API Usage

Method	Uniform Resource Indicator (URI)	Description
POST	/benefits/v1/spending-account-plans	Creates an in-progress spending account plan in ADP Workforce Now.

Application Scope

The canonical URI corresponding to the Benefit Spending Account Plan 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:

/benefits/benefitsManagement/spendingAccountManagement/organizationAccountPlanManagement/accountPlan.add

Supported Actors

Request Parameter roleCode Value	Usage
Practitioner	System user that creates a Benefit Spending Account Plan.

Sequence of Interactions

1. Your consumer application makes a **/benefits/v1/spending-account-plans** request to the ADP API endpoint.
2. The ADP API endpoint responds to the consumer application with the details of **/benefits/v1/spending-account-plans**.

Data Dictionary

Plans created using the Benefit Spending Account Plan API will be available in the ADP Workforce Now user interface (UI) by selecting **Setup > Benefits > Plan Setup.** Resources listed in the following table can be accessed by selecting **Setup > Benefits > Plan Setup.**

When selecting the created and incomplete plan the fields in the following table are shown under the **Plan settings**, **Eligibility Group Details**, and **Define contributions** sections.

Important: All the fields mentioned in the following table cannot be edited by the ADP Workforce Now Client Practitioner. So, make sure you send all the required details in the payload to complete the plan setup.

Schema Location	Field Name in ADP Workforce Now UI	Is Required (Y/N)	Section in ADP Workforce Now UI	Note
id/value				
alternateId/value	**Policy Number**	Y	**Plan settings**	
legalPlanName	**Plan Name**	Y		
providerId		Y		Predefined ID provided by the external provider. **Important:** The external provider name is shown under the **Provider** field in the ADP Workforce Now UI.
providerId/value				
productCode	**Plan Type**	Y		Supported plan types are: • MedicalFSA • Limited Purpose FSA • Dependent Care FSA • Parking • Transit

				The following are the **productCode** types: • "Parking" • "Transit" • "MedicalFSA" • "LimitedPurposeFSA" • "DependentCareFSA"
effectivePeriod				This is the Eligibility Group effective date.
effectivePeriod/validFrom	**Start Date**	N	**Plan settings**	
originalEffectiveDate	Not displayed			
planAnniversary	**Renewal Date [MM/DD]**	Y	Plan settings	This is the plan anniversary day and month.
WaitingRuleType			**Eligibility Group Details**	
WaitingRuleType/rollForward Indicator	**First of month...**			An indicator must determine whether the 'First of month' waiting rule type can be rolled forward to the first of the following month when the date falls on the first of the month.
WaitingRuleType/ruleValue				Value indicating the number of months or days for the waiting rule type code.
WaitingRuleType/ waitingRuleEventCode		Y		The waiting rule can be one of the following: • **WorkEvent** • **NewHire**
WaitingRuleType/ waitingRuleTypeCode		Y		One of the following values can be used for the **waitingRuleTypeCode**: • **HireDate** • **FirstOfMonthAfterXMonths** • **AfterXMonths** • **AfterXDays** • **FirstOfMonthAfterXDays** • **OtherProbation** • **FirstOfMonthAfterHire** • **FirstPayPeriodAfterXDays** • **NoWaitingRule**

/accountBasedPlanSetupType/employerContributionType	**Employer Contribution**	Y	**Define contributions**	Contribution Type can either be **Flat** or **Percent**.
/accountBasedPlanSetupType/accountBasedCoverageLevels/employerContributionType/contributionAmount				
/accountBasedPlanSetupType/ accountBasedCoverageLevels/employerContributionType/contributionAmount/Currency				Currency code is USD.
/accountBasedPlanSetupType/ accountBasedCoverageLevels/employerContributionType/contributionAmount/value				
/accountBasedPlanSetupType/ accountBasedCoverageLevels/employerContributionType/ contributionType				
/accountBasedPlanSetupType/ accountBasedCoverageLevels/minimumContribution	**Minimum Employee Contribution**	Y	**Define contributions**	
/accountBasedPlanSetupType/accountBasedCoverageLevels/ minimumContribution/currency				Currency code is USD.
/accountBasedPlanSetupType/accountBasedCoverageLevels/ minimumContribution/value				
/accountBasedPlanSetupType/accountBasedCoverageLevels/ maximumContribution	**Minimum Employee Contribution**	Y	**Define contributions**	
/accountBasedPlanSetupType/ accountBasedCoverageLevels/maximumContribution/currency				Currency code is USD.
/accountBasedPlanSetupType/accountBased				

CoverageLevels/ maximumContribution/value				
/accountBasedPlanSetupType/accountBasedCoverageLevels/ coverageLevelType	**Coverage level**	Y	**Define contributions**	Visible when Health Savings Account (HSA) plan is created Supported Coverage level types are **Employee** and **Employee Family**.

The following table consists of the following:

- Combinations that can be used for waiting rule type.
- Corresponding field names are enabled in the ADP Workforce Now based on the combinations passed.

The following resources can be accessed in the ADP Workforce Now UI by selecting **Setup > Benefits > Plan Setup.** When selecting the created and incomplete plan the following fields are shown under **Eligibility Group Details**.

Section in ADP Workforce Now UI	Combinations	Field Name in ADP Workforce Now UI
New Hire Waiting Period	"waitingRuleEventCode": "NewHire","waitingRuleTypeCode": "HireDate" Or "waitingRuleEventCode": "NewHire","waitingRuleTypeCode": "NoWaitingRule"	**Date of hire (no waiting period)**
New Hire Waiting Period	"waitingRuleEventCode": "NewHire","waitingRuleTypeCode": "FirstOfMonthAfterXMonths" Or "waitingRuleEventCode": "NewHire","waitingRuleTypeCode": "FirstOfMonthAfterXDays"	**First of month...**
New Hire Waiting Period	"waitingRuleEventCode": "NewHire","waitingRuleTypeCode": "AfterXMonths" Or "waitingRuleEventCode": "NewHire","waitingRuleTypeCode": "AfterXDays"	**following date of hire**
Work Event Waiting Period	"waitingRuleEventCode": "WorkEvent","waitingRuleTypeCode": "HireDate"OR "waitingRuleEventCode": "WorkEvent","waitingRuleTypeCode": "NoWaitingRule"	**Date of work event (no waiting period)**
Work Event Waiting Period	"waitingRuleEventCode": "WorkEvent","waitingRuleTypeCode": "FirstOfMonthAfterXMonths" OR "waitingRuleEventCode": "WorkEvent", "waitingRuleTypeCode": "FirstOfMonthAfterXDays"	**First of month...**
Work Event Waiting Period	"waitingRuleEventCode": "NewHire","waitingRuleTypeCode": "AfterXMonths" OR "waitingRuleEventCode": "NewHire","waitingRuleTypeCode": "AfterXDays"	**following date of work event**

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.

API Editor guide link : Spending Accounts Plan For ADP WFN

Response Code	Condition	messageText	Tips to Handle
Response Code	Condition	messageText	Tips to Handle

201 Created	Adding or creating a Benefit Plan Flexible Spending Account (FSA) in ADP Workforce Now.	{
		"type":"SUCCESS", "code": "CREATED", "description": "SUCCESS"}	

201 Created	Adding or creating a Healthcare Spending Account (HSA) Plan in ADP Workforce Now.	{
		"type":"SUCCESS", "code": "CREATED", "description": "SUCCESS"}	

201 Created	Adding or creating a Benefit Plan without passing **waitingRuleType**.	{
		"type":"SUCCESS", "code": "CREATED", "description": "SUCCESS"}	

201 Created	Updating a Practitioner approved benefit plan.	{
		"type":"SUCCESS", "code": "CREATED", "description": "SUCCESS"}
			To update a benefit plan, make sure that you provide the following details same as what you provided while adding a plan
			• id/value
			• providerId/value
			• productCode

201 Created	Updating a created plan, which is in a pending state (not reviewed and approved by the ADP Workforce Now Practitioner).	{
		"type": "SUCCESS",
		"code": "OK",
		"description": "SUCCESS"
		}	

400 Bad Request	Adding or updating a Benefit plan with a maximum contribution more than the plan limits.	{
		"type": "ERROR",
		"code": "BAD_REQUEST",
		"description": " Maximum contribution exceeds plan limits. "
		}
			Make sure the maximum contribution is within the IRS plan limits of the respective benefit spending account plans.

400 Bad Request	Adding or updating a Benefit spending account plan with an invalid coverage level type.	"userMessage": { "codeValue": "JSON Error ", "messageTxt": "JSON parse error : Cannot deserialize value of type `com.adp.wfn.benefits.externalfsa.dto. AccountBasedCoverageLevel$CoverageLevelType` from String \"Emplo\": not one of the values accepted for Enum class: [EmployeeDependent, EmployeeFamily, EmployeeSpouse, ChildOnly, EmployeeChildren, Employee, SpouseOnly];	For now, supported **coverageLevel Types** are the following: • Employee for FSA plans, transit, parking • Employee and EmployeeFamily for HSA plans.
400 Bad Request	Adding a plan with an invalid **productCode**.	userMessage": { "codeValue": "JSON Error ", "messageTxt": "JSON parse error: Cannot deserialize value of type `com.adp.wfn. benefits.plansetup. pojo.ProductCode List` from String \"Hsp\": value not one of declared Enum instance names: [LimitedPurposeFSA, MedicalFSA, DependentCareFSA, HSA, Parking, Transit];}	Supported plan types are: • MedicalFSA, • Limited Purpose FSA • Dependent Care FSA • Parking, Transit The **productCode** types are: • "Parking"

			• "Transit" • "MedicalFSA" • "LimitedPurposeFSA" • "DependentCareFSA"
400 Bad Request	Adding a plan with invalid **ProviderID**.	{ "code": "BAD_REQUEST", "description": "Invalid PCProvider Id."}	Give a pre-determined **providerID** provided by the external benefit provider.
400 Bad Request	Adding a plan with blank **ProviderID** value.	{ "code": "BAD_REQUEST", "description": "Invalid PCProvider Id."}	
400 Bad Request	Adding a plan with a blank **productCode** value.	"userMessage": { "codeValue": "JSON Error ", "messageTxt": "JSON parse error: Cannot deserialize value of type` com.adp.wfn. benefits.plansetup. pojo.ProductCode List` from String \"Hsp\": value not one of declared Enum instance names: [LimitedPurposeFSA, MedicalFSA, DependentCareFSA, HSA, Parking, Transit];}	
400 Bad Request	Adding a plan with blank **alternateID** value.	{ "code": "BAD_REQUEST", "description": "Alternate Id is not present. "}	
400 Bad Request	Adding a plan with a blank or Invalid **waitingRule EventCode** value.	"userMessage": { "codeValue": "JSON Error ", "messageTxt": "JSON parse error: Cannot deserialize value of type` com.adp.wfn. benefits.plansetup. pojo.WaitingRule Type$WaitingRuleEventCode` from String \"\": value not one of declared Enum instance names: [WorkEvent, NewHire, LifeEvent]}	The waiting rule event can be one of the following: • **WorkEvent** • **NewHire**
400 Bad Request	Adding a plan with blank or Invalid **waitingRule TypeCode** value.	"userMessage": { "codeValue": "JSON Error ", "messageTxt": "JSON parse error: Cannot deserialize value of type ` com.adp.wfn. benefits.plansetup. pojo.WaitingRule Type$WaitingRule TypeCode . from String \"\": value not one of declared Enum instance names: [OtherProbation, HireDate, AfterXDays, FirstOfMonthAfter Hire, FirstOfMonthAfterXMonths, FirstPayPeriodAfterXDays, AfterXMonths, NoWaitingRule, FirstOfMonthAfterXDays]}	One of the following values can be used for the **waitingRuleTypeCode**: • "HireDate" • "FirstOfMonth AfterXMonths" • "AfterXMonths" • "AfterXDays" • "FirstOfMonthAfterXDays" • "Other Probation" • "FirstOfMonth AfterHire" • "NoWaiting Rule"
500 Internal Server Error	Adding a plan with blank plan **Anniversary**.	{ "type": "ERROR", "code": "INTERNAL_SERVER_ERROR"}	
500 Internal Server Error	When the request payload is incorrect.	{ "type": "ERROR", "code": "INTERNAL_SERVER_ERROR"}	