**Budget API Specification Document**

Version 0.1 draft

September 5, 2019

**Prepared by:**

**Piyush Singla**

Document Change Record

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Author** | **Change Details** |
| 05/09/2019 | 0.1 | Piyush Singla |  |
|  |  |  |  |

Objective

Objective of this Document is to explain the API structure for Budget Manager. This will help to understand what the API is about and how to consume this.

API List

1. Check Budget: - This API ill check the budget for the against the entities and amount which should be passed by consuming application. Details of this API is below: -
   1. [URL:-](file:///C:\Users\arun.manglick\AppData\Local\Microsoft\Windows\INetCache\Content.Outlook\76D5EDMO\-) /api/Consumption/CheckBudget/
   2. Type: - POST
   3. Custom Headers: - UserExecutionContext
   4. Request: -

{

"DocumentType": "string",

“DocumentCode”: long

“ParentDocumentCode”: long

"LineInfos": [{

"LineItemId": long,

"ProcessingDate": datetime

"Quantity": decimal,

"UnitPrice": decimal,

"ItemType": decimal,

"LineTotal": decimal,

“IsCancelled”: Boolean,

“FlippedDocumentCode”: long

"Splits": [{

"SplitAmount": decimal,

"SplitQuantity": decimal,

"Entities": [{

" Name": "string",

"Value": "string"

}],

}]

}]

}

|  |  |
| --- | --- |
| DocumentType: | This will represent which document is calling for budget Check. For SMART the values for this will be “Requisition”, ”Order”, “Invoice”. |
| DocumentCode: | This will represent the id or the document. |
| ParenetDocumentCode: | this will contain the id of parent document from where change document has been created. |
| LineInfos: | Array of all the lines with details. |
| LineItemId : | This will represent the line item id which is common across the documents. For SMART it will be P2PLineItemId. |
| ProcessingDate: | This will represent the date based on which budget needs to be checked. |
| Quantity: | This will represent the quantity of that line. |
| UnitPrice: | This will represent the unit price of that line. |
| ItemType: | This will represent the type of the item. It can two values “Material” and “Service”. |
| LineTotal: | This will represent the total amount that line. |
| Splits: | This will represent all the splits for that line. |
| IsCancelled | This will be true, if the line is cancelled on Ui side. |
| FlippedDocumentCode | This the document id of the document from which that document is flipped. For. E.g For Order, it will be id of requisition |
| SplitAmount: | This will represent the total amount that split. |
| SplitQuantity: | This will represent the quantity of that split. |
| Entities: | This will represent the list of accounting entities present for the split. |

* 1. Response: -

{

"IsSuccess": boolean,

"DocumentBudgetAction": "string",,

"BudgetCheckDetails": [

{

"

"LineItemId": long

"Splits": [

{

"SplitBudgetAction": " string,

“SplitAmount”: decimal,

“SplitQuantity”: decimal,

“BudgetEntityAllocationId”: long

}

]

}

]

}

]

}

|  |  |
| --- | --- |
| IsSuccess: | This will represent whether the check budget is successful or not. |
| DocumentBudgetAction: | This will provide the result of budget check at document level. Values can be “No Match found”, ”No Approval”, ”Simple Approval”, “Complex Approval”, “Over Budget” |
| LineItemId: | The line item id of the line. For Smart it will be P2PLineItemId. |
| Splits: | Entities of the splits which are matched. |
| SplitBudgetAction: | This will provide the result of budget check at split level level. Values can be “No Match found”, ”No Approval”, ”Simple Approval”, “Complex Approval”, “Over Budget” |
| SplitAmount | Amount of the Split |
| SplitQuantity | Quantity of the split |
| BudgetEntityAloocationId | This is the budget entity allocation id in budget. This is for documents to save and fetch the budget details after in document process. |

1. Consume Budget- This API ill consume the budget for the against the entities and amount which should be passed by consuming application. Details of this API is below: -
   1. [URL:-](file:///C:\Users\arun.manglick\AppData\Local\Microsoft\Windows\INetCache\Content.Outlook\76D5EDMO\-) /api/Consumption/ConsumeBudget/
   2. Type: - POST
   3. Custom Headers: - UserExecutionContext
   4. Request: -

{

"DocumentType": "string",

“DocumentCode”: long

“ParentDocumentCode”: long

"LineInfos": [{

"LineItemId": long,

"ProcessingDate": datetime

"Quantity": decimal,

"UnitPrice": decimal,

"ItemType": decimal,

"LineTotal": decimal,

“IsCancelled”: Boolean,

“FlippedDocumentCode”: long

"Splits": [{

"SplitAmount": decimal,

"SplitQuantity": decimal,

"Entities": [{

" Name": "string",

"Value": "string"

}],

}]

}]

}

|  |  |
| --- | --- |
| DocumentType: | This will represent which document is calling for budget Check. For SMART the values for this will be “Requisition”, ”Order”, “Invoice. |
| DocumentCode: | This will represent the id or the document. |
| ParenetDocumentCode: | This will contain the id of parent document from where change document has been created. |
| LineInfos: | Array of all the lines with details. |
| LineItemId : | This will represent the line item id which is common across the documents. For SMART it will be P2PLineItemId. |
| ProcessingDate: | This will represent the date based on which budget needs to be checked. |
| Quantity: | This will represent the quantity of that line. |
| UnitPrice: | This will represent the unit price of that line. |
| ItemType: | This will represent the type of the item. It can two values “Material” and “Service”. |
| LineTotal: | This will represent the total amount that line. |
| Splits: | This will represent all the splits for that line. |
| IsCancelled | This will be true, if the line is cancelled on Ui side. |
| FlippedDocumentCode | This the document id of the document from which that document is flipped. For. E.g For Order, it will be id of requisition |
| SplitAmount: | This will represent the total amount that split. |
| SplitQuantity: | This will represent the quantity of that split. |
| Entities: | This will represent the list of accounting entities present for the split. |

1. Response: -

{

"IsSuccess": boolean,

"DocumentBudgetAction": "string",,

"BudgetCheckDetails": [

{

"

"LineItemId": long

"Splits": [

{

"SplitBudgetAction": " string,

“SplitAmount”: decimal,

“SplitQuantity”: decimal,

“BudgetEntityAllocationId”: long

}

]

}

]

}

]

}

|  |  |
| --- | --- |
| IsSuccess: | This will represent whether the check budget is successful or not. |
| DocumentBudgetAction: | This will provide the result of budget check at document level. Values can be “No Match found”, ”No Approval”, ”Simple Approval”, “Complex Approval”, “Over Budget” |
| LineItemId: | The line item id of the line. For Smart it will be P2PLineItemId. |
| Splits: | Entities of the splits which are matched. |
| SplitBudgetAction: | This will provide the result of budget check at split level level. Values can be “No Match found”, ”No Approval”, ”Simple Approval”, “Complex Approval”, “Over Budget” |
| SplitAmount | Amount of the Split |
| SplitQuantity | Quantity of the split |
| BudgetEntityAloocationId | This is the budget entity allocation id in budget. This is for documents to save and fetch the budget details after in document process. |

1. Release Budget: - This API ill release the budget for the against the entities and amount which should be passed by consuming application. Details of this API is below: -
   1. [URL:-](file:///C:\Users\arun.manglick\AppData\Local\Microsoft\Windows\INetCache\Content.Outlook\76D5EDMO\-) /api/Consumption/ReleaseBudget/
   2. Type: - POST
   3. Custom Headers: - UserExecutionContext
   4. Request: -

{

"DocumentType": "string",

“DocumentCode”: long

}

|  |  |
| --- | --- |
| DocumentType: | This will represent which document is calling for budget Check. For SMART the values for this will be “Requisition”, ”Order”, “Invoice”, “Change Requisition”, “Change Order”. |
| DocumentCode: | This will represent the id or the document. |

* 1. Response: -

{

"IsSuccess": boolean,

"Message": "string",

}

|  |  |
| --- | --- |
| IsSuccess: | This will represent whether the release budget is successful or not. |
| Message | This will provide a meaningful message for release budget failure. |

1. GetMasterEntities: - This API will provide the list of all the standard entities on which budget can be setup on. This API will help the consuming application to know the entities which they have o pass with values for other budget operations.
   1. [URL:-](file:///C:\Users\arun.manglick\AppData\Local\Microsoft\Windows\INetCache\Content.Outlook\76D5EDMO\-) /api/MasterEntity/GetMasterEntities/
   2. Type: - GET
   3. Custom Headers: - UserExecutionContext
   4. Response: -

{

“EntityNames”: []

}

|  |  |
| --- | --- |
| EntityNames | List of Entity names |