



TIBCO® Offer and Price Engine

Web Services Guide

Version 6.0.0
March 2023



Contents

Contents	2
About This Product	3
Transport Modes	4
SOAP Over HTTP	4
REST over HTTP/HTTPS	4
Web Services	14
Get Offer Service	14
Validate Offer Service	27
Get Price Service	33
Get Product Information Service	39
Get Price Information Service	44
Search and Browse APIs	47
Search	47
Browse	53
Configuration Service API Samples	80
Actuator APIs	80
Application Metadata APIs	83
Application Properties APIs	84
Configuration File APIs	86
Configuration Notification APIs	87
Shopping Cart APIs	87
Error Codes	91
TIBCO Documentation and Support Services	93
Legal and Third-Party Notices	95

About This Product

TIBCO® Offer and Price Engine is a cloud-native, in-memory omnichannel server of offers and prices for digital service providers. It answers requests from a digital service provider's customer-facing channels for offers and prices, subject to business rules such as customer eligibility and product compatibility.

TIBCO Offer and Price Engine is the next generation of, and partially replaces, TIBCO® Fulfillment Order Management. To better align the direction of TIBCO Fulfillment Order Management with market demand, the product's capabilities have been reorganized into two new products:

- TIBCO® Order Management
- TIBCO® Offer and Price Engine

Customers who are currently on maintenance for TIBCO Fulfillment Order Management are entitled to upgrade to both TIBCO Order Management and TIBCO Offer and Price Engine. TIBCO will continue to support TIBCO Fulfillment Order Management, and there is currently no plan to retire TIBCO Fulfillment Order Management. New capabilities will be developed in TIBCO Order Management and TIBCO Offer and Price Engine.

Transport Modes

The following transport modes are supported in TIBCO Offer and Price Engine:

- [SOAP over HTTP](#)
- [REST over HTTP](#)

SOAP Over HTTP

SOAP over HTTP service calls use the standard synchronous web services message format. The integration pattern is as follows:

SOAP over HTTP Integration Pattern

In the SOAP over HTTP pattern, a service client makes a synchronous service call to the Offer and Price Engine services component. The service calls the appropriate operation and performs the required operation before sending a reply back to the calling application.

The endpoint URL is as follows:

- Offer and Price Engine services: `http://<host>:<port>/opes/api/v1/opeService`

REST over HTTP/HTTPS

REST over HTTP service calls use the standard web services message format. The integration pattern is as follows:

REST over HTTP/HTTPS Integration Pattern

In the REST over HTTP/HTTPS pattern, a service client makes a service call to the Offer and Price Engine services component. The service calls the appropriate operation and performs the required operation before sending a reply back to the calling application.

REST Services

TIBCO Offer and Price Engine consumes online and offline model loading. In the offline model loading, different models (for example: price, product, category, discount, rule, and offer) generated from modeling products, such as TIBCO Product and Service Catalog are loaded through catalog service in pluggable cache and TIBCO Offer and Price Engine reads those models from the pluggable cache object library.

Swagger UI Usage

The Swagger URL for various services are as follows:

Service	Swagger URL
authorization	<i>http://OPES_SERVER_IP:9091/swagger-ui/index.html</i>
configurator	<i>http://OPES_SERVER_IP:9090/swagger-ui/index.html</i>
configurator-ui	<i>http://OPES_SERVER_IP:9104</i>
catalog-service	<i>http://OPES_SERVER_IP:9092/swagger-ui/index.html</i>
opes Endpoints	<i>http://OPES_SERVER_IP:8090/swagger-ui/index.html</i>
shoppingcart	<i>http://OPES_SERVER_IP:8091/swagger-ui/index.html</i>
offersearchindex-service	<i>http://OPES_SERVER_IP:8093/swagger-ui/index.html</i>

Service	Swagger URL
recommendation-engine-api	<i>http://OPES_SERVER_IP:8095/swagger-ui/index.html</i>
recommendation-engine-data-mapper	<i>http://OPES_SERVER_IP:8096/swagger-ui/index.html</i>
recommendation-engine-data-store	<i>http://OPES_SERVER_IP:8094/swagger-ui/index.html</i>
recommendation-engine-core	<i>http://OPES_SERVER_IP:8092</i>

Endpoint details with various operations:

Service	EndPoint	Method	Description
Offer and Price Engine service	<protocol>://<host>:<port>//opes/api/v1/offers/categories	POST	Operation getCategories
	<protocol>://<host>:<port>/opes/api/v1/offers/eligible	POST	Operation getEligibleOffers
	<protocol>://<host>:<port>/opes/api/v1/offers/validate	POST	Operation validateOffers
	<protocol>://<host>:<port>/opes/api/v1/offers/price/info	POST	Operation getPriceInformation
	<protocol>://<host>:<port>/opes/api/v1/offers/prices	POST	Operation getPrices
	<protocol>://<host>:<port>/opes/api/v1/offers/pr	POST	Operation

Service	EndPoint	Method	Description
	products		getProductInformation
Configuration service	<protocol>://<host>:<port>/v1/configuration/{applicationid}	GET	Operation Get Configuration
	<protocol>://<host>:<port>/v1/configuration/{applicationid}	POST	Operation Add Configuration
	<protocol>://<host>:<port>/v1/configuration/{applicationid}	PUT	Operation Update Configuration
	<protocol>://<host>:<port>/v1/configuration/{applicationid}	DELETE	Operation Delete Configuration
	<protocol>://<host>:<port>/v1/configuration/replicateTenantProperties	POST	Operation Replicate Tenant Properties
	<protocol>://<host>:<port>/v1/configuration/uploadConfigFile/{applicationid}	POST	Operation Upload Configuration File
Catalog service	<protocol>://<host>:<port>/v1/productmodel	POST	Operation SubmitProductModel
	<protocol>://<host>:<port>/v1/productmodel/bulk	DELETE	Operation PurgeProductModels
	<protocol>://<host>:<port>/v1/productmodel/all	GET	operation get all productmodels

Service	EndPoint	Method	Description
	<protocol>://<host>:<port>/v1/productModel/bulk	GET	operation get bulk productmodels
	<protocol>://<host>:<port>/v1/productModel/bulk	POST	Operation SubmitProductModels
	<protocol>://<host>:<port>/v1/categoryModel	POST	Operation Submit category Model
	<protocol>://<host>:<port>/v1/categoryModel/bulk	GET	Operation Get Bulk Category Model
	<protocol>://<host>:<port>/v1/categoryModel/bulk	POST	Operation Submit Bulk Category Model
	<protocol>://<host>:<port>/v1/categoryModel/bulk	DELETE	Operation Delete Bulk Category Model
	<protocol>://<host>:<port>/v1/categoryModel/All	GET	Get All Category Model
	<protocol>://<host>:<port>/v1/discountModel	POST	Operation Submit Single Discount Model
	<protocol>://<host>:<port>/v1/discountModel/bulk	GET	Operation Get Bulk Discount Model

Service	EndPoint	Method	Description
	<protocol>://<host>:<port>/v1/discountModel/bulk	POST	Operation Submit Bulk Discount Model
	<protocol>://<host>:<port>/v1/discountModel/all	GET	Operation Get All Discount Model
	<protocol>://<host>:<port>/v1/discountModel/bulk	DELETE	Operation Delete Bulk Discount Model
	<protocol>://<host>:<port>/v1/offerIdMappings	POST	Operation Submit Bulk Offers
	<protocol>://<host>:<port>/v1/offerIdMappings	DELETE	Operation Purge All Offers
	<protocol>://<host>:<port>/v1/offerIdMappings/all	GET	Get All offer Model
	<protocol>://<host>:<port>/v1/priceModel	POST	Operation Submit Single Price Model
	<protocol>://<host>:<port>/v1/priceModel/bulk	POST	Operation Get Bulk Price Model
	<protocol>://<host>:<port>/v1/priceModel/bulk	POST	Operation Submit Bulk Price Models
	<protocol>://<host>:<port>/v1/priceModel/bulk	DELETE	Operation Delete Bulk Price Models

Service	EndPoint	Method	Description
	<protocol>://<host>:<port>/v1/priceModel/all	GET	Operation Get All Price Model
	<protocol>://<host>:<port>/v1/ruleModel	POST	Operation Submit Single Rule Model
	<protocol>://<host>:<port>/v1/ruleModel/bulk	GET	Operation Get Bulk Rule Model
	<protocol>://<host>:<port>/v1/ruleModel/bulk	POST	Operation Submit Bulk Rule Models
	<protocol>://<host>:<port>/v1/ruleModel/bulk	DELETE	Operation Delete Bulk Rule Models
Offer search index service	<protocol>://<host>:<port>/v1/offers/search	POST	Operation searchOffers
Shopping Cart service	<protocol>://<host>:<port>/opes/api/v1/shopping Cart	GET	Operation getShoppingCart
	<protocol>://<host>:<port>/opes/api/v1/shopping Cart	POST	Operation createShoppingCart
	<protocol>://<host>:<port>/opes/api/v1/shopping Cart/{id}	GET	Operation getShoppingCart ById
	prototype://<host>:<port>/opes/api/v1/shopping Cart/{id}	DELETE	Operation deleteShoppingCartById

Service	EndPoint	Method	Description
	<protocol>://<host>:<port>/opes/api/v1/shoppingCart/{id}	PATCH	Operation updateShoppingCart
Authorization service	<protocol>://<host>:<port>/v1/user	GET	Get user
	<protocol>://<host>:<port>/v1/user	POST	Create user
	<protocol>://<host>:<port>/v1/user	POST	Update user
	<protocol>://<host>:<port>/v1/user	DELETE	Delete user
Recommendation Engine Data Mapper	<protocol>://<host>:<port>/v1/jsonDataMapper	POST	Submit data mapper
	<protocol>://<host>:<port>/v1/csvDataMapper		
	<protocol>://<host>:<port>/v1/publishJsonData <protocol>://<host>:<port>/v1/publishCsvData	POST	Publish data mapper
Recommendation Engine Core	<protocol>://<host>:<port>/v1/findAssociationRules/sync	GET	Find association rules
	<protocol>://<host>:<port>/v1/calculateProductsSimilarity/sync	GET	Get similar products
	<protocol>://<host>:<port>/v1/schedule/job/start	POST	Schedule the job start time for each tenant
	<protocol>://<host>:<port>/v1/schedule/job/stop	POST	Schedule the job stop time for each tenant
Recommend	<protocol>://<host>:<port>/v1/pmml/{version}	PUT	Make specific

Service	EndPoint	Method	Description
ation Engine Data Store			version of PMML Model file active
	<protocol>://<host>:<port>/v1/pmml/{version}	DELETE	Delete specific version of PMML Model file
	<protocol>://<host>:<port>/v1/pmml	GET	Download All PMML Model Files for Tenant ID
	<protocol>://<host>:<port>/v1/pmml	POST	Upload PMML file
	<protocol>://<host>:<port>/v1/pmml/metadata	GET	Get metadata of All PMML Model Files for Tenant ID
	<protocol>://<host>:<port>/v1/pmml/active	GET	Download Active PMML Model Files for Tenant ID
Recommendation Engine API	<protocol>://<host>:<port>/v1/queryProductRecommendation	GET	List or find query product recommendation objects
	<protocol>://<host>:<port>/v1/queryProductRecommendation	POST	Create query product recommendation
	<protocol>://<host>:<port>/v1/queryProductRecommendation/{Id}	GET	List or find query product

Service	EndPoint	Method	Description
			recommendation objects

Web Services

TIBCO® Offer and Price Engine provides support for generating and validating the product offering for the customer and provides pricing for the offers. The engine integrates with the product catalog provided by TIBCO® Product and Service Catalog for product definition at design time.

To generate and validate the offer, get the price for the generated offer, and get the product and price information, TIBCO Offer and Price Engine exposes the following services:

- [Get Offer Service](#)
- [Validate Offer Service](#)
- [Get Price Service](#)
- [Get Product Information Service](#)
- [Get Price Information Service](#)

The following samples have been added for the TIBCO® Offer and Price Engine web services:

- Offer and price engine services model samples located in `$OPE_HOME/samples/models`
- Web service requests and responses, including the Offer ID functionality, located in `$OPE_HOME/samples/request_response`.

Get Offer Service

This service validates the incoming offer with the existing basket of the customer.

The validation considers product integrity, product compatibility with the products in the offer, segment compatibility with basket items, and group requirements. In case of ceased products, it checks if the item is present in TIBCO Product and Service Inventory in a blocked status. Depending upon the migrations created in the product model, the offer includes the products to be migrated for otherwise incompatible products. Eligibility is evaluated for all products including the auto-provisioned products. In case any auto-

provisioned child is invalid, the product is ineligible.

The services utilize the following relationships:

- CompatibleSegment
- IncompatibleSegment
- Category
- CompatibleProduct
- IncompatibleProduct
- Migration including the existing ProductComposedOf and ProductRequiredFor

The offer is generated on a particular set of inputs and returns the list of products that can be offered to the subscriber. The inputs in the request can include any of the following filters:

- Segments
- Categories
- Promotions
- Record Types and Record Subtypes
- Focus (focuses on products in the subscriber's basket)
- Characteristics
- Product Status

In case these filters are present, the products satisfying only these criteria are considered for the offer. In case no filters are present, all product offerings are considered for generating offers, and the validation rules are applied.

You can utilize the following request flags for this web service:

Get Offer Request Flags

Request Flag Name	Default
ReturnIneligibleProducts	True

Request Flag Name	Default
ReturnBundleOfferings	False
ReturnCharacteristicsDetails	True
ReturnPrices	False
ReturnProductInformation	True
validateData	False
validateProductRequiredForGroups	True
validateProductComprisedOfGroups	True
validateProductCompatibility	True
validateSegmentCompatibility	True
decomposeProducts	True
ValidateProdDate	False
enforceCompatibleSegment	False
skipChildFilterValidation	False

Get Offer Filters

OPE uses the following filters to get the initial list of eligible products, which are then validated through the normal process of eligible products evaluation:

Filter	Description
Focus	The focus filter considers the products that are present in the basket, order, or an existing inventory to get the initial list of eligible products. If the focus filter is present, the rest of the filters are ignored. Because the

Filter	Description
	product in focus is already present with the customer, the optional products for the selected focus are used as the initial list of eligible products.
Promotions	The products in promotions filters are directly used to get the initial list of eligible products. If promotions are present, all other filters except focus are ignored to get the initial list of eligible products.
Segment	The segment filter provides a way to get a list of products belonging to the same segment. The products are related to the segment name and segment types. The initial list of eligible products contains a union of all the products, which includes the same and different segment types unless they have an incompatible relationship defined. Any products that do not have any compatible or incompatible relationships defined are considered to be compatible with that segment type.
Record Type	This filter provides a way to get the initial list of all the products belonging to the record type specified.
Record Sub Type	This filter provides a way to get the initial list of products belonging to the subrecord types specified.
Category	This filter provides a way to get the initial list of products belonging to the specified category.
Input Field	This field sets linking relevant fields and returns eligible products using those fields. This means that it checks for the link definitions from the relationships, such as ProductComprisedOf, ProductRequiredFor, and compatibleProducts. If those link definitions are present in the input filter list, the product is considered eligible.
Order Date Filter	This filter defines the order date for the request. If this is not set, the engine assumes the current date as the order date.
Return Characteristics Details	This filter provides the option to only return the inputted characteristics details, such as InstanceMax, InstanceMin, and Itemspec.

Filter	Description
Return Characteristics Name	This filter provides the option to only return the inputted characteristic name.
Return Characteristics Description	This filter provides the option to only return the inputted characteristic description.
Return Characteristics Value Type	This filter provides the option to only return the inputted characteristic value type.

Get Offer Request

Get Offer Request Body Details

Element Name	Element Type	Description
BusinessTransactionID	String (Optional)	Used for the upstream web service to correlate a response with the appropriate request.
ReturnIneligibleProducts	Boolean (Optional)	Toggles whether products, which are found to be ineligible as a result of a specific reason, are returned in the response. The products that are present in the Eligible Products Holder and are removed as being incompatible with a product, or child of a product, in the order, are identified as ineligible products once detected. The default is value true.
ReturnBundleOfferings	Boolean	Bundle support, which is for

Element Name	Element Type	Description
	(Optional)	getting eligible bundles for the current basket. Only those products which have optional children with <code>Autoprovision=false</code> in the order request are eligible. The default value is <code>false</code> .
ReturnPrices	Boolean (Optional)	If set, prices for the generated offer are returned. The default value is <code>false</code> .
ReturnCharacteristicsDetails	Boolean (Optional)	If set to <code>false</code> , it stops sending details such as "InstanceMax", "InstanceMin", and "Itemspec" details for the <code>ProductComprisedOf</code> characteristic and skips the ineligible optional products in the response altogether. For <code>ReturnCharacteristicsDetails</code> , the default value is <code>true</code> .
ReturnProductInformation	Boolean (Optional)	If set to <code>true</code> , it returns the product information for the eligible product. The default value is <code>true</code> .
RequestControl/validateData	Boolean (Optional)	Not used in this service.
RequestControl/ validateProductRequiredForGroups	Boolean (Optional)	Toggles if <code>ProductRequiredFor</code> groups are evaluated in eligibility. The default value is <code>true</code> .

Element Name	Element Type	Description
RequestControl/ validateProductComprisedOfGroups	Boolean (Optional)	Toggles if ProductComprisedOf groups are evaluated in eligibility. The default value is true.
RequestControl/validateProductCompatibility	Boolean (Optional)	Toggles if ProductCompatibility is evaluated in eligibility. The default value is true.
RequestControl/validateSegmentCompatibility	Boolean (Optional)	Toggles if SegmentCompatibility is evaluated in eligibility. The default value is true.
RequestControl/validateProdDate	Boolean (Optional)	Toggles if the engine checks that the products are valid based on the configured start dates and end dates. The default value is false for validateProdDate.
RequestControl/decomposeProducts	Boolean (Optional)	Toggles if products are decomposed into auto-provisioned children or requiredForchildren. The default value is true.
RequestControl/ReturnProductInformation Data	String (Optional)	With this element, you can configure characteristic filters for returning product information. There is an "or" relationship between characteristic name (ReturnCharacteristicsName), characteristic description (ReturnCharacteristicsDescriptio

Element Name	Element Type	Description
		n), characteristic details (ReturnCharacteristicsDetails), and characteristic value type (ReturnCharacteristicsValueType).
SubscriberID	String (Optional)	Eligible products are presented for the subscriber specified in these fields. This information is used in the context of the order and inventory to perform compatibility and integrity checks.
CustomerID	String (Optional)	Eligible products are presented for the subscriber specified in these fields. This information is used in the context of the order and inventory to perform compatibility and integrity checks.
Segment	Boolean (Optional)	<p>Information used to derive the initial list of eligible products. Per segment type, a segment element and all related segments must be mapped as names.</p> <p>If multiple segments are provided for a SegmentType, the union of all associated products is returned. Multiple SegmentTypes return an intersection of all products compatible with all SegmentTypes.</p>

Element Name	Element Type	Description
OrderDate	dateTime (Optional)	If OrderDate is entered, products in the eligible product list are filtered based on configured start and end dates of each product for that date, otherwise, the system time is taken.
RecordTypes	String (Optional)	<p>If one or more RecordTypes values are included in the request, products in the eligible product list are filtered after the expansion to include AutoProvision and Non Mandatory children. This means that all the products in the eligible product list that meet the filtering criteria are returned. RecordType is compared to the configured class of the product.</p> <p>Note: Multiple RecordTypes results in a union of all products having one of the provided RecordTypes.</p>
Promotions	String (Optional)	<p>Promotion products is a reduced list of products that is evaluated by this request. Promotion products are filtered according to the other criteria specified in the request and tested for compatibility with products in the order.</p>

Element Name	Element Type	Description
RecordSubType	String (Optional)	<p>If one or more RecordSubTypes are included in the request, products in the eligible product list are filtered while the expansion includes AutoProvision and Non Mandatory children. This means that all of the products in the eligible product list that meet the filtering criteria, plus any children or children of children, are returned. RecordSubType is compared to the configured Subclass of the Product.</p> <p>Note: Multiple RecordSubTypes result in a union of all products having one of the provided RecordSubTypes.</p>
Categories	String (Optional)	<p>If one or more categories are included in the request, products in the eligible product list are filtered for these categories. If you provide multiple categories, an intersection of the results for each category is presented.</p> <p>If one auto-provisioned child is not compatible with a category, the parent is not either.</p>
Focus	Focus (Optional)	With this field, you can set a basket into focus. The focused item's child products are

Element Name	Element Type	Description
		evaluated and attached as eligible products.
InputField	Field (Optional)	With this field, you can set linking to relevant fields and returns to evaluate eligible products using those fields.
orderRequest	orderRequest (Optional)	Products in the basket can be passed into the eligibility engine through the order to maintain compatibility between products that are already ordered.
Status	String	This element is not being used.

Get Offer Response

Get Offer Response Details

Element Name	Element Type	Description
BusinessTransactionID	String (Optional)	If provided, it is passed from the request and can be used to correlate service calls across disparate services to track actual business process flows in the system.

Get Offer Result Status Element Details

Element Name	Element Type	Description
operation	String	The name of the operation is "GetOffer"

Element Name	Element Type	Description
	(Optional)	
component	String (Optional)	The name of the service is "GetOffer"
severity	String (Optional)	The possible values are as follows: S = for success ERROR = for error
code	String (Optional)	The error code is a string that identifies the error and the TIBCO Fulfillment Orchestration Suite component that produced the error. An error code of 0 indicates a successful invocation of the service.
message	String (Optional)	A descriptive text string associated with code.

Eligible or Ineligible Product Element Details

Element Name	Element Type	Description
ProductID	String (Mandatory)	The ID of the eligible product.
InternalID	String (Optional)	The internal ID of the product instance that is used for correlation with the ParentInternalID of a child product.
ParentInternalID	String (Optional)	The internal ID of the product's parent instance that is used for correlation with the InternalID of a parent product.
SubscriberID	String (Optional)	The subscriber ID for which the eligibility request was made.

Element Name	Element Type	Description
CustomerID	String (Optional)	The customer ID for which the eligibility request was made.
UDFs	UDFType (Optional)	Each UDF contains a name field and a value field. Eligibility tries to enrich products with UDFs as much as possible by mapping UDFs from related (compatible, parent, child) products. If linking relevant UDFs could not be mapped from inventory or basket items, they are created with a default value <code>#{new}</code> .
Messages	String (Optional)	Attached ValidationMessage elements which display validation results.
Price	ChildPriceBaseType (Optional)	If set from request, it returns the price for this product.

Validation Message Details

Element Name	Element Type	Description
Code	String (Mandatory)	The code of the validation message.
Type	String (Optional)	The type of validation message. It is mostly an error message because validation messages are only created in case of errors. The only exception is in the case of a migration because the possibility of migration is indicated using a message of type info.
Details	String (Optional)	The corresponding textual message, configurable in the GVs located in: AFF/OfferConfigurationValidation/

Element Name	Element Type	Description
Constants/Errors/OfferValidation		
CausingItem	String (Optional)	The CausingItem field indicates which type of item caused the validation error or message, such as TrolleyItem or UDF.
CausingItemID	String (Optional)	The CausingItemID field displays the ID of the item causing the validation error or message, such as the ProductID or the UDF Name.
CausingItemPosition	String (Optional)	The CausingItemPosition indicates the position type of the item causing the validation error or message, such as "LineNumber" for TrolleyItem
CausingItemPositionID	String (Optional)	The CausingItemPositionID indicates the position of the item causing the validation error or message, such as a LineNumber for TrolleyItem
RuleViolation	String (Optional)	The RuleViolation displays the concrete violation of the particular rule which created this message, for example, violation of a maximum constraint of the ProductComprised group would be "Group.Max: 2.0 / ActualCount: 3.0".

Validate Offer Service

This service validates the request based on the number of inputs including segments, order, and offer. When you submit an order request, that order is submitted to TIBCO Offer and Price Engine. Based on the product model and group constraints, it returns a message stating if the products are valid or not.

The validation considers product integrity, product compatibility with the products in offer, segment compatibility, category compatibility, and group requirements of basket items. In the case an inventory is supplied, the service validates inventory with the items in the basket and checks for single-use, product compatibility with basket items, and group requirements. In the case of cease products, the service checks if the item is present in inventory in a blocked status. In case any of these checks fail, this service invalidates the

offer and returns an appropriate error response.

You can apply IncompatibleSegment to a LinkID for the ValidateOffer web service. To implement this, add a UDF with the name SegmentLinkID under the seg:Segment element and under the orderline UDF where you want to map the LinkID under the validateOffer request.

Validate Offer Request Flags

Request Flag Name	Default
validateData	False
validateProductRequiredForGroups	True
validateProductComprisedOfGroups	True
validateProductCompatibility	True
validateSegmentCompatibility	True
decomposeProducts	True
validateProdDate	False
enforceCompatibleSegment	False
skipChildFilterValidation	False

Validate Offer Request

Validate Offer Request Details

Element Name	Element Type	Description
BusinessTransactionID	String (Optional)	Used for the upstream web services to correlate a response with the appropriate

Element Name	Element Type	Description
		request.
CalculateConsequences	Boolean (Mandatory)	Used to enforce migration for basket items. If it is set to false, those items are invalid. If it is set to true, those items become valid, and the corresponding migration actions are done during the request.
RequestControl/validateData	RequestControl (Optional)	With this element, you can turn off the validation of UDFs, such as type, mandatory, and compatibility, to support a staged order configuration in the sales process. If this is not set, it defaults to false.
RequestControl/validateProdDate	Boolean (Optional)	Toggles if the engine checks that the products are valid based on the configured start dates and end dates. The default value for validateProdDate is false.
RequestControl/validateProductRequiredForGrou	Boolean	Toggles if

Element Name	Element Type	Description
ps	(Optional)	ProductRequiredFor groups are evaluated in eligibility. The default value is true.
RequestControl/validateProductComprisedOfGroups	Boolean (Optional)	Toggles if ProductComprisedOf groups are evaluated in eligibility. The default value is true.
RequestControl/validateProductCompatibility	Boolean (Optional)	Toggles if ProductCompatibility is evaluated in eligibility. The default value is true.
validateSegmentCompatibility	Boolean (Optional)	Toggles if SegmentCompatibility is evaluated in eligibility. The default value is true.
RequestControl/decomposeProducts	Boolean (Optional)	Toggles if products are decomposed into auto-provisioned children or requiredForchildren. The default value is true.
Segments	Boolean (Optional)	Segment information is used to validate the compatibility of order items against the provided segments.

Element Name	Element Type	Description
orderRequest	orderRequest (Optional)	Products in the basket can be passed into the validation engine through the order.

Validate Offer Response

Validate Offer Result Status Element Details

Element Name	Element Type	Description
Component	String (Optional)	The name of the service is ValIdateOffer
Operation	String (Optional)	The name of the operation is ValIdateOffer
Severity	String (Optional)	The possible values are as follows: S = for success ERROR = for error
Code	String (Optional)	The error code is the string that identifies the error and the TIBCO Fulfillment Orchestration Suite component that produced the error. An error code of 0 (zero) indicates a successful invocation of the service.
Message	String (Optional)	A descriptive text string associated with the code

Validation Pass Details

Element Name	Element Type	Description
ValidationPass	Boolean (Mandatory)	True if the request is validated and no messages of type 'ERROR' are thrown

Validate Offer Result Message Details

Element Name	Element Type	Description
LineNumber	String (Mandatory)	Indicates on which item the messages were created.
ProductID	String (Mandatory)	The product ID for which the product the message was created.
Code	String (Mandatory)	The code of the validation message.
Type	String (Optional)	The type of validation message. It is mostly an error message because validation messages are only created in case of errors. The only exception is in the case of a migration.
Details	String (Optional)	The corresponding textual message is configurable in the GVs located in AFFOfferConfigurationValidation/Constants/Errors/OfferValidation
CausingItem	String (Optional)	The CausingItem field indicates which type of item caused the validation error or message, for example, TrolleyItem or UDF.
CausingItemID	String (Optional)	The CausingItemID field displays the ID of the item causing the validation error or message, for example, the Product ID or the UDF name.
CausingItemPosition	String (Optional)	The CausingItemPosition indicates the position type of the item causing the validation error or message, for example, LineNumber for Orderline.
CausingPositionID	String (Optional)	The CausingItemPositionID indicates the position of the item causing the validation error or message, for example, LineNumber for Orderline.

Element Name	Element Type	Description
RuleViolation	String (Optional)	The RuleViolation displays the concrete violation of the particular rule that created this message, for example, violation of a maximum constraint of Product comprised group is Group.Max: 2.0 / Actual count: 3.0.
RelatedSegment	String (Optional)	<p>Segments for subscribers are mapped into these elements. They are taken into account for the PriceRequiresSegment and DiscountRequiresSegment mechanisms.</p> <p>Segments must be mapped in the Syntax as follows:</p> <ol style="list-style-type: none"> 1. If SegmentName and SegmentType does not contain dash(-), the syntax must be SegmentName-SegmentType 2. If SegmentName and SegmentType contain a dash(-) like Segment-Name, syntax must be Segment-Name SegmentType <pre><Segment> <Name>Segment-Name</Name> <Type>SegmentType</Type> </Segment></pre>

Validation Pass Details

Element Name	Element Type	Description
Order	Order (Mandatory)	The complete offer is returned, regardless of whether order lines are added or not.

Get Price Service

The getPrices request accepts multiple combinations of products in the request and generates multiple prices in the single response of these combinations.

The pricing service enables the pricing of a particular offer or order based on the following:

- Configuration of product and prices based on the offer catalog
- Configuration of the given set of products in conjunction with segments
- Characteristics pass as part of the getPrices request

The pricing service supports multiple pricing requests in a single service request. This service also calculates the prices in combination with optional products. The pricing service can accept multiple combinations to generate requests. These are mapped to multiple request elements in the GetPrices requestBody elements. The response for requestBody is contained in the ResponseBody. Pricing is calculated per RequestBody and not across them.

Get Price Request

Request Header Details

Element Name	Element Type
BusinessTransactionID	String (Optional)
ReturnTotals	Boolean (Optional)

Request Body Details

Element Name	Element Type	Description
ProductID	Product (Mandatory)	These are the products considered for price calculation. They could be the products in the basket or the final offer. Any entities from other request bodies within the getPrices request are treated separately.
InternalID	String (Optional)	The internal ID of the product instance that is used for correlation with ParentInternalID of a child

Element Name	Element Type	Description
		product.
ParentInternalID	String (Optional)	The internal ID of the product's parent instance is used for correlation with the InternalID of a parent product.
SubscriberID	String (Optional)	The subscriber ID for which the request was made.
CustomerID	String (Optional)	The customer ID for which the request was made.
ProductName	String (Optional)	The product's name that is configured in the catalog.
ProductType	String (Optional)	The class or type of the product that is configured in the catalog.
ProductSubType	String (Optional)	The subclass or subtype of product that is configured in the catalog.
calculationproduct/linenumber	String (Mandatory)	The line number of the order line.
Product/Action	String (Mandatory)	Action for the product to be calculated. Prices are only calculated for PROVIDE Calculation products.
Product/ProductID	String (Mandatory)	Product ID of the product.

Element Name	Element Type	Description
Quantity	Quantity (Mandatory)	The quantity is specified by the quantity and the amount together with the unit of measurement of the quantity. There is no validation on the quantity's unit of measurement. They are not evaluated nor are quantities recalculated for different units of measurement, such as milliliter to liter.
UDF	UDF Type (Optional)	Each UDF contains a name field and a value field specified at the product level (Product and OptionalProduct).
OptionalProduct	OptionalProduct (Optional)	These products are considered optional for the calculation. A price is calculated for each combination of the CalculationProducts and each optional product. For example, for a request with Product A and Product B as CalculatedProducts, and Product C, Product D, and Product E as optional products, three results are available that contain a combination of products A, B, and C, products A, B, and D, and products A, B, and E.
RelatedSegment	String (Optional)	Segments for subscribers are mapped into these elements. They are then taken into account for the PriceRequiresSegment and DiscountRequiresSegment mechanism. Segments must be

Element Name	Element Type	Description
		<p>mapped in the Syntax: SegmentName-SegmentType.</p> <p>Alternatively, segments can be mapped using RelatedSegment/Segment/Name and RelatedSegment/Segment/Type.</p> <p>For example,</p> <pre><ns6:RelatedSegment> <Segment> <Name>Web Self_ Service</Name> <Type>GROUP_ CHANNEL</Type> </Segment> </ns6:RelatedSegment></pre>

Get Price Response

The response mapping rule maps the calculated prices and their attributes into UDF elements and creates corresponding modifier elements for existing discounts. Specified UDFs on the product level (Product and OptionalProduct) are passed through for easier identification and processing of products further on. Price characteristics and Discounts characteristics are mapped additionally to the predefined UDFs.

Get Price Response Body Details

Element	Description
CorrelationID	CorrelationID of the request body
Result	The result element with all result details
Result/ReturnCode	The return code for the pricing request

Element	Description
CalculationResult	The pricing response body corresponding to the combination of calculated and optional products
CalculationResult/CalculationResultLine	The calculated product in the combination
CalculationResultLine /lineNumber	The line number of calculated and optional products
CalculationResultLine/Action	The action of the calculated product and optional product
CalculationResultLine/ProductInformation	ProductInformation of the calculated product
Quantity	The quantity of the calculation of the product and optional product
PriceItem	The price applied for the calculated product. This includes the prices of all the child products from this line as well
Priceitem/UDFs	The pricing UDFs
PriceItem/Modifier items	The discounts and child discounts applied to the price
ModifierItems/UDFs	The discount UDFs
CalculationResult/Totals	The total for the offer
Totals/TotalPrice	The total per identical price type
Totalprice/Type	The price type
Totalprice /Item	The price item details
Item/ItemID	The comma-separated price IDs

Element	Description
Item/ItemSubTotal	The price for the item before discount
Item/ ItemDiscount	The total Discounts applied to these items
Item/FinalPrice	The final price after discounts
Currency	The currency of the price item

The ResultType contains a ReturnCode that reflects an overall result of the particular price request, valid values are 'SUCCESS', 'TECHNICAL-ERROR' and 'BUSINESS-ERROR'.

Get Price Response Return Messages

Return Message	Reason
SUCCESS	No errors
TECHNICAL- ERROR	The unexpected behavior due to bugs or catalog configuration errors
BUSINESS-ERROR	A product has no prices attached or an invalid product is passed in the request.

Each response element can have multiple results based on the number of OptionalProduct provided in the request. Each calculation result contains lines with each line having a product associated with 0 to n prices. The particular price characteristics, including discount characteristics, are stored in UDFs.

Get Product Information Service

This service gives the product ID, relationships, attributes, and so on when requesting product information. It also gets the information present in the product model for the product.

You can search for a product by the following information:

- Product ID
- Record Type
- Record Subtype
- Segment
- Category

The engine then provides all the products that fit the search criteria.

Get Product Information Request

Get Product Information Request Details

Element Name	Element Type	Description
BusinessTransactionID	String (Optional)	Used for the upstream web services to correlate a response with the appropriate request.
Query	String (Optional)	You have a choice: products can either be retrieved by specifying a query or by explicitly the product IDs.
IDList	IDList (Optional)	You have a choice: Products can either be retrieved by specifying a query or by explicitly the product IDs.

Get Product Information Request Query Details

Element Name	Element Type	Description
Class	String (Optional)	If one or more class values are included in the request, products in the eligible product list are filtered after the expansion to include AutoProvision and Non Mandatory children. This means that all of the products in the eligible product list that meets the filtering criteria are returned.

Element Name	Element Type	Description
		<p>Class is compared to the configured record type of the product.</p> <p>Note: Multiple subclasses result in a union of all products having one of the provided classes.</p>
SubClass	String (Optional)	<p>If one or more subclasses are included in the request, the products in the eligible product list are filtered while expansion includes AutoProvision and Non Mandatory children. This means that all the products in the eligible product list that meet the filtering criteria, plus any children or children of children are returned. The subclass is compared to the configured RecordSubType of the Product.</p> <p>Note: Multiple subclasses results in a union of all products having one of the provided subclasses.</p>
Category	String (Optional)	<p>If one or more categories are included in the request, the products in the eligible product list are filtered for these categories. If you provide multiple categories, an intersection of the results for each category is presented.</p> <p>If one auto-provisioned child is not compatible with a category, the parent is not either.</p>
Segment	Boolean (Optional)	<p>Information is used to derive the initial list of eligible products. Per segment type, a segment element and all related segments must be mapped as names.</p> <p>If multiple segments are provided for a SegmentType, the union of all associated products is returned. Multiple SegmentTypes returns an intersection of all products compatible with all SegmentTypes.</p>

Get Product Information Response

Get Product Information Response Details

Element Name	Element Type	Description
BusinessTransactionID	String (Optional)	If provided, it is passed through from the request and can be used to correlate service calls across disparate services to track actual business process flows in the system.

Get Product Information Result Status Details

This class is a result structure common for all operations. For any operation, the reply schema contains this mandatory object. It specifies if the service execution succeeded or failed. Any client of an operation must first check the value of the code field to identify if the execution was successful or not.

Element Name	Element Type	Description
Component	String (Optional)	The name of the service is Product Information.
Operation	String (Optional)	The name of the operation is Product Information.
Severity	String (Optional)	The possible values are as follows: S = for success ERROR = for error
Code	String (Optional)	The error code is the string that identifies the error and the TIBCO Fulfillment Orchestration Suite

Element Name	Element Type	Description
		component that produced the error. An error code of 0 (zero) indicates a successful invocation of the service.
Message	String (Optional)	A descriptive text string associated with the code.

Get Product Information Element Details

Element Name	Description
Product/ProductID	The ID of the product that is configured in the catalog.
Product/ ProductType	The class or type of product that is configured in the catalog.
Product / ProductName	The product's name that is configured in the catalog.
Product / Description	The description of the product that is configured in the catalog. Note: The long description is not returned by this service.
Product/Characteristic	The characteristics of the product.

Characteristic Element Details

Element Name	Description
Name	The ProductID of the characteristic.
Values/Value/Value	The value of the characteristic. The optional DefaultValue is configured to show the DefaultValue of the Characteristic. If the RelationshipValue is configured in the Characteristic relationship, this value shows, and Value and the Characteristic's DefaultValue

Element Name	Description
	shows as DefaultValue. If no relationship value is configured, Value and DefaultValue are the same.
Values/Value/ValueType	The product's name that is configured in the catalog.
Product / Description	The valueType of the Characteristic.

Get Price Information Service

This service fetches the pricing for all products, including individual products and the total of all products within the offer.

The Get Price service uses the Product Priced By relationship. It accepts multiple price calculation requests in a single service call.

Each request body contains the following inputs:

- **Calculation Product**
Used for the multiple products for which the price needs to be calculated
- **Optional Product**
Used for each product considered in combination with the calculation of products
- **Segment**
Used for the prices belonging to a particular segment filter are calculated for the offer
- **Quantity**
Used for quantity for each product

Price is not displayed for child products separately. Prices are aggregated towards parents. The price contains child prices identified by the PriceComprisedOf relationship. If the price contains discounts, it is identified by the price altered by discount. If a discount contains child discounts, it is identified by the DiscountComprisedOf relationship. If a parent's price contains a discount, that discount is applied to the child as well.

Get Price Information Request

Get Price Information Request Details

Element Name	Element Type	Description
BusinessTransactionID	String (Optional)	Used to allow the upstream web services to correlate a response with the appropriate request
ProductIDs	String (Mandatory)	The list of product IDs, for which the price information is required

Get Price Information Response

Get Price Information Response Details

Element Name	Element Type	Description
Return Code	Result	Returns a code for the operation status. The valid values are Success, BUSINESS ERROR, and TECHNICAL ERROR.

Element Name	Element Type	Description
Product ID	Price Product	The product ID of the product passed in the request
PriceInformationType	Price Product	The price of the product

Element Name	Element Type	Description
Class [Attribute]	PriceInformationType	The price type of the price
Price/ID	PriceInformationType	The price record ID of the price

Element Name	Element Type	Description
Price/Info/Class	PriceInformationType	The price type of the price
Price/Info/Priority	PriceInformationType	Price Priority in the product
Price/Info/isActive	PriceInformationType	Record_status of the price record
Price/Info/StartDate	PriceInformationType	Start date as defined in price model
Price/Info/EndDate	PriceInformationType	End date as defined in the price model
Price/Info/ModelCharacteristics	PriceInformationType	The characteristics of the price model
Price/Info/Restrictions	PriceInformationType	PriceRequiresCharac, PriceRequiresSegment, and PriceRequiresProductGroup relationship
Price/Info/Amount/Value	PriceInformationType	The charge value of the price
Price/Info/Amount/UOM	PriceInformationType	The charge unit of measurement of the price
Price/Info/MinAmount/Value	PriceInformationType	The minimum amount of the price
Price/Info/isOverride	PriceInformationType	The subclass of the price
Price/ChildPrices/Price	PriceInformationType	The child price of the parent price
Price/ChildPrices/Price[Lvl]	PriceInformationType	The level of child price in hierarchy

Element Name	Element Type	Description
Price/ChildPrices/Price[parentID]	PriceInformationType	The price ID of the parentID
Price/ChildPrices/Price[ID]	PriceInformationType	The price ID of the child price
Price/ChildPrices/Info	PriceInformationType	The details of the price ID of the child price

Search and Browse APIs

These Search and Browse APIs are supported by offer search index service with the following swagger UI URL:

`<protocol>://<host>:<port>/v1/offers/search`

Search

- Search Fields

The following fields of the Product data model are always considered for text searches.

- productID

```
<ns0:productId>PRODUCT_ID_1<ns0:productId>
```

- Product_Description

```
<ns0:description>PRODUCT_DESCRIPTION</ns0:description>
```

- Category

```
<ns1:characteristics>
    <ns1:name>CATEGORY</ns1:name>
```

- CompatibleSegment

The name and type of the segment.

```
<ns1:characteristics>
  <ns1:name>AS</ns1:name>
  <ns1:description>CompatibleSegment</ns1:description>
```

- Record_Type and Record_Sub_Type

The name of the record type and the subtype.

```
<ns1:characteristics>
  <ns1:name>RECORD_TYPE</ns1:name>
  <ns1:value>
    <ns1:type>Feature</ns1:type>
    <ns1:discreteValue>P0</ns1:discreteValue>
  </ns1:value>
</ns1:characteristics>
<ns1:characteristics>
  <ns1:name>RECORD_SUB_TYPE</ns1:name>
  <ns1:value>
    <ns1:type>Feature</ns1:type>
  </ns1:value>

  <ns1:discreteValue>ProductOffering</ns1:discreteValue>
</ns1:value>
```

- Characteristics

By default, characteristics are not searched so they have to be explicitly enabled for text searches.

```
<ns0:characteristics>
  <ns0:name>OPE_Char_A</ns0:name>
  <ns0:description>Characteristic</ns0:description>
  <ns0:simpleRule>
    <ns0:name>ISFULLSEARCHABLE</ns0:name>
    <ns0:ruleSetOutcome>true</ns0:ruleSetOutcome>
  </ns0:simpleRule>
```

- Keyword or phrases

You can search all products indexed into the Offer Search Engine for specific keywords or phrases.

When searching, a corresponding MatchQuery or MatchQueryPhrase is executed to look for occurrences of the specified text or phrase across all products and a score is

attached to each. The response includes the products that contain the specified text or phrase that has the highest score. The score is based on the term frequency-inverse document frequency (TFIDF) which is a numerical statistic that reflects how relevant the text or keyword is to search. A higher score implies a higher relevancy and a closer match to the specified text or keyword.

```
Request Payload
{
    "keyword": "OPE_Prod_A",
    //if it's a phrase
    "isPhrase": "true"
}
```

- Category

You can search for a specific category or multiple categories across all products indexed into the Offer Search Engine.

When searching for a specific category, the products that match the category are returned.

When searching for multiple categories, the union of matching products across all categories are returned.

```
Request Payload
{
    "searchOfferQuery" : {
        "aggregate" : "false",
        "globalAggregate" : "false",
        "categories" : [
            {
                "id": "test1",
                "name": "WEB"
            }
        ]
    }
}
```

You can search for categories along with keywords or phrases. In such a search you would first find products based on a category (or categories) and then narrow the results using keywords or phrases.

The query in the following example searches for products that have 2GB and that fall in the WEB category:

Request Payload

```
{
  "keyword": "2GB"
  "searchOfferQuery" : {
    "aggregate" : "false",
    "globalAggregate" : "false",
    "categories" : [
      {
        "id": "test1",
        "name": "WEB"
      }
    ]
  }
}
```

The category search can be used with attributes to filter out only those categories that have the particular attribute set. For example, the following request payload filters products based on keyword and category attributes across specific categories:

```
{
  "keyword": "PROD_A1",
  "aggAllCategories": "false",
  "aggAllSegments": "false",
  "searchOfferQuery" : {
    "aggregate" : "false",
    "globalAggregate" : "false",
    "attributes": "false",
    "categories" : [
      {
        "id": "category_1",
        "name": "category_1"
      }
    ]
  },
  "attributeFilters": {
    "attributes": {
      "IMP_DATE": [
        "Samsung"
      ]
    }
  }
}
```

```

Response Payload
{
  "totalHits": 1,
  "offers": {
    "offer": [
      {
        "offerId": "PROD_A1",
        "offerDescription": ""
      }
    ]
  },
  "suggestions": {
    "suggestion": null
  }
}

```

- Segment

You can search for a specific segment name or type across all products indexed into the Offer Search Engine.

When searching for a specific segment name or type, the products that match the criteria are returned.

When searching for multiple segment names or types, the union of products that match the criteria are returned.

The query in the following request searches for products that match CHANNEL, WEB, and NOWEB.

```

Request Payload
{
  "searchOfferQuery" : {
    "aggregate" : "false",
    "globalAggregate" : "false",
    "segments" : [
      {
        "id": "test1",
        "type": "GROUP_CHANNEL",
        "name": "External Party"
      },
      {
        "id": "test2",
        "type": "SUB_BRAND",
        "name": "LOOP"
      }
    ]
  }
}

```

```

    }
  ]
}

```

You can search for segments along with keywords or phrases. In such a search you would first find products based on a segment and then narrow the results using keywords or phrases.

The query in the following example searches for products that have 2GB in the model that also have the CHANNEL, WEB, and NOWEB classifications.

```

{
  "keyword": "2GB"
  "searchOfferQuery" : {
    "aggregate" : "false",
    "globalAggregate" : "false",
    "classes" : [
      {
        "id": "test1",
        "type": "CHANNEL",
        "names": ["WEB", "NOWEB"]
      }
    ]
  }
}

```

Search Flags

Flag	Description
isfullsearchable	Makes a particular characteristic searchable for full-text search.
com.tibco.af. ope.flags.suggest=true	An application configuration that enables spelling suggestions to be included in response.
aggallcategories	A request flag that returns aggregation of all categories linked to the product.
aggallsegments	A request flag that returns aggregation of all segments linked to the product.

aggofferclasses

A request flag that returns aggregation of all record types/subtypes linked to the product.

See sample search requests and responses at <OPE_HOME>/samples/request_response/search.

Browse

The Search API presents filters and attributes that enables you to browse using category, segment, and record types and subtypes as specified in the Product data model.

This functionality enables you to browse for products using the following navigation paths:

- **Category**

You can browse for products using the category (or categories) modeled in the Product data model. This feature is implemented by using the category name against the indexed product models in the Offer Search Engine. For example, using the category-based filters you are presented with aggregate values such as the total number of offers for a specific or all categories, the offers for a particular category or categories, or offers filtered based on attributes of a category.

Categories are loaded into the Offer Search Engine using the Model Loading Engine. The Category data model contains a root category along with the subcategories and the associated characteristics.

The Product data model is related to the Category data model by the relationship defined in the Product data model. When a product with the category relationship is indexed in the Offer Search Engine, guided browsing is enabled using the implemented aggregation API calls.

The Offer Search Engine uses the Category model to maintain a list of the categories present in the OP Core engine. Using these models, you can query the categories and subcategories to reach the leaf categories for a specific root category.

Use cases for browse functionality:

- **Browse all categories with count** - Returns all the categories present in the product models along with the count. The following flag is used to aggregate all categories: **"aggAllCategories":"true"**

```
{
    "aggAllCategories":"true"
}
Response payload
```

- **Browse specific categories with count** - The search engine exposes Get Categories REST api which returns the root category along with sub categories. The user can drill down to the leaf category and then use the Search API to get the total number of products available for a leaf category with optionally returning those products. This is enabled by turning the request attribute "aggregate=true" as in the following request sample:

```
{
    "aggAllCategories":"false",
    "searchOfferQuery" : {
        "aggregate" : "true",
        "globalAggregate" : "false",
        "categories" : [
            {
                "id": "test1",
                "name": "WEB"
            }
        ]
    }
}
```

- **Browse specific category along with other categories** - Supports browsing a specific category and returns offers and their count. Additionally it also returns all other categories present in all product models. This is useful particularly in use cases where there is a need to filter by particular category and also maintain the global context of all categories (modeled in product model) and their counts:

```
{
    "aggAllCategories":"false",
    "searchOfferQuery" : {
        "aggregate" : "false",
        "globalAggregate" : "true",
```

```

        "categories" : [
        {
            "id": "test1",
            "name": "WEB"
        }
        ]
    }
}

```

- **Category attribute-based browse and filters** - This functionality supports browsing and search of products based on different category attributes. Category attributes are modeled as characteristics in Category model with characteristic name as the attribute of that category. The corresponding characteristic is modeled in the product model linked to that category with the value of that attribute. The engine this way determines the characteristics belonging to the category being browsed or searched and returns the offers and their count per filter. To enable just browsing the characteristics for a particular category, the following request payload is used:

```

{
    "searchOfferQuery" : {
        "aggregate" : "true", // to get count of all products
        "globalAggregate" : "false",
        "attributes": "true",
        "categories" : [
            {
                "id": "test1",
                "name": "category_2"
            }
        ]
    }
}

```

Response Payload

```

{
    "totalHits": 1,
    "offers": {
        "offer": [
            {
                "offerId": "PROD_A1",

```

```

        "offerDescription": ""
    }
]
}
"categoryAggregations": {
    "currentTotal": 1,
    "otherTotal": 0,
    "categories": [
        {
            "name": "test1",
            "currentTotal": 1,
            "otherTotal": 0,
            "id": null
        }
    ]
},
"filterAggregations": {
    "total": 2,
    "filterAggregations": [
        {
            "currentTotal": 1,
            "otherTotal": 0,
            "id": "IMP_DATE",
            "childFilterAggregations": [
                {
                    "currentTotal": 1,
                    "otherTotal": 0,
                    "id": "Samsung",
                    "childFilterAggregations": []
                }
            ]
        }
    ]
},
"global": {
    "segmentAggregations": null,
    "categoryAggregations": null,
    "recordClassAggregations": null
}
}

```

- **Category attribute-based range filter**

This filter supports product browsing and search based on different category attributes and range filters. Category attributes are modeled as characteristic names

in the Category model. The corresponding characteristic is modeled in the product model linked to the category and takes the value of the category attribute. Modeling enables the search engine to determine the characteristics of the browsed or searched category. The search engine returns the results and their count, according to the range filter criteria provided in the query.

Using the filter's search criteria, `minRange` and `maxRange` (integer or double data type), you can search for characteristics that include the `externalRuleset` attribute where the value of the `RANGE_VALUE` parameter is `true`, and where the `discreteValue` attribute has a data type of `integer` or `double`. If the characteristic contains a `discreteValue` attribute with a `char` or `string` data type, the system logs a parsed error and the characteristic is not processed.

Note: Elasticsearch range aggregation excludes the "from value (`minRange`)" and includes the "to value (`maxRange`)" for each range. However, if the range limit (lower bound or upper bound) in the range filter is `x.5`, the search engine uses a rounded range limit to round off the value to `x.6`.

For example, when aggregating cellphones from a 5 to a 5.5 screen size, the search engine changes the upper bound limit (5.5) to 5.6, applying the elasticsearch range aggregation from a 5 to a 5.6 screen size.

Use the following payload to enable browsing for characteristics with a range filter:

```
{
  "searchOfferQuery":{
    "aggregate":"true",    // to get count of all products
    "globalAggregate":"true",
    "attributes":"true",
    "categories":[
      {
        "id":"CellPhones",
        "name":"CellPhones"
      }
    ]
  },
  "attributeFilters":{
    "ranges":[{"
      "attributeName":"ScreenSize",
      "minRange":5,
      "maxRange":8
```

```

    }
  }
}

```

Response payload:

```

{
  "totalHits": 9,
  "offers": {
    "offer": [
      {
        "offerId": "iPhone11-Black",
        "offerDescription": "iPhone11 Black from Apple"
      },
      {
        "offerId": "s10-red",
        "offerDescription": "Samsung Galaxy s10 Red"
      },
      {
        "offerId": "iPhone11-Black-Max",
        "offerDescription": "iPhone11 Black Max from Apple"
      },
      {
        "offerId": "vivo",
        "offerDescription": "vivo"
      },
      {
        "offerId": "redmi",
        "offerDescription": "redmi"
      },
      {
        "offerId": "s10-silver",
        "offerDescription": "Samsung Galaxy s10 Silver"
      },
      {
        "offerId": "s10-black",
        "offerDescription": "Samsung Galaxy s10 Red"
      },
      {
        "offerId": "iPhone11-gold",
        "offerDescription": "iPhone11 Gold from Apple"
      }
    ]
  }
}

```

```

        "offerId": "oppo",
        "offerDescription": "oppo"
    }
]
},
"segmentAggregations": {
    "currentTotal": 0,
    "otherTotal": 0,
    "types": []
},
"categoryAggregations": {
    "currentTotal": 9,
    "otherTotal": 0,
    "categories": [
        {
            "name": "CellPhones",
            "currentTotal": 9,
            "otherTotal": 0,
            "id": null
        }
    ]
},
"offerClasses": {
    "currentTotal": 0,
    "otherTotal": 0,
    "id": null,
    "types": []
},
"filterAggregations": {
    "total": 36,
    "filterAggregations": [
        {
            "currentTotal": 9,
            "otherTotal": 0,
            "id": "Color",
            "childFilterAggregations": [
                {
                    "currentTotal": 5,
                    "otherTotal": 0,
                    "id": "Gold",
                    "childFilterAggregations": [],
                    "childRangeFilterAggregations": []
                },
                {

```

```

        "currentTotal": 3,
        "otherTotal": 0,
        "id": "Black",
        "childFilterAggregations": [],
        "childRangeFilterAggregations": []
    },
    {
        "currentTotal": 1,
        "otherTotal": 0,
        "id": "Red",
        "childFilterAggregations": [],
        "childRangeFilterAggregations": []
    }
],
"childRangeFilterAggregations": []
},
{
    "currentTotal": 9,
    "otherTotal": 0,
    "id": "Brands",
    "childFilterAggregations": [
        {
            "currentTotal": 3,
            "otherTotal": 0,
            "id": "Apple",
            "childFilterAggregations": [],
            "childRangeFilterAggregations": []
        },
        {
            "currentTotal": 3,
            "otherTotal": 0,
            "id": "Samsung",
            "childFilterAggregations": [],
            "childRangeFilterAggregations": []
        },
        {
            "currentTotal": 1,
            "otherTotal": 0,
            "id": "oppo",
            "childFilterAggregations": [],
            "childRangeFilterAggregations": []
        },
        {
            "currentTotal": 1,

```

```

        "otherTotal": 0,
        "id": "redmi",
        "childFilterAggregations": [],
        "childRangeFilterAggregations": []
    },
    {
        "currentTotal": 1,
        "otherTotal": 0,
        "id": "vivo",
        "childFilterAggregations": [],
        "childRangeFilterAggregations": []
    }
],
"childRangeFilterAggregations": []
},
{
    "currentTotal": 9,
    "otherTotal": 0,
    "id": "MemoryStorage",
    "childFilterAggregations": [
        {
            "currentTotal": 8,
            "otherTotal": 0,
            "id": "64G",
            "childFilterAggregations": [],
            "childRangeFilterAggregations": []
        },
        {
            "currentTotal": 1,
            "otherTotal": 0,
            "id": "128G",
            "childFilterAggregations": [],
            "childRangeFilterAggregations": []
        }
    ],
    "childRangeFilterAggregations": []
},
{
    "currentTotal": 9,
    "otherTotal": 0,
    "id": "ScreenSize",
    "childFilterAggregations": [],
    "childRangeFilterAggregations": [
        {

```

```

        "currentTotal": 9,
        "otherTotal": 0,
        "id": "ScreenSize",
        "from": 5,
        "to": 8,
        "childFilterAggregations": []
    }
}
],
},
"global": {
    "segmentAggregations": {
        "currentTotal": 0,
        "otherTotal": 0,
        "types": []
    },
    "categoryAggregations": {
        "currentTotal": 19,
        "otherTotal": 0,
        "categories": [
            {
                "name": "CellPhones",
                "currentTotal": 9,
                "otherTotal": 0,
                "id": null
            },
            {
                "name": "FamilyPlan",
                "currentTotal": 3,
                "otherTotal": 0,
                "id": null
            },
            {
                "name": "Unlimited",
                "currentTotal": 3,
                "otherTotal": 0,
                "id": null
            },
            {
                "name": "Tablets",
                "currentTotal": 2,
                "otherTotal": 0,
                "id": null
            }
        ]
    }
}

```

```

    },
    {
      "name": "Watches",
      "currentTotal": 2,
      "otherTotal": 0,
      "id": null
    }
  ]
},
"recordClassAggregations": {
  "currentTotal": 0,
  "otherTotal": 0,
  "id": null,
  "types": []
}
}
}

```

- **Record or Subtype**

You can browse for products using the record type modeled in the Product data model. This feature is implemented by using the record name or type against the indexed products in the Offer Search Engine. For example, using the record type-based filters you are presented with aggregate values such as the total number of offers for a specific or all segments or the offers for a particular segment. All filters might optionally contain both the record name and type.

The product is related to the record type and the subtype by the characteristics in the Product data model. When a product with the record and subtype relationship is indexed in the Offer Search Engine, guided browsing is enabled using the implemented aggregation API calls.

Using these models, you can browse all record types and subtypes. Additionally, you can use the total number of products that meet the specified criteria. For example:

```

Request
{
  "aggOfferClasses" : "true",
  "searchOfferQuery" : {
    "aggregate" : "true",
    "globalAggregate" : "true",
    "classes" : [

```

```

        {
            "id": "test1",
            "type": "BPO",
            "names": ["INTERNET","NOWEB"]
        }
    ]
}

response
{
    "totalHits": 1,
    "offers": {
        "offer": [
            {
                "offerId": "Prod108_TFOM-5410",
                "offerDescription": "Prod108_TFOM-5410"
            }
        ]
    },
    "segmentAggregations": {
        "currentTotal": 0,
        "otherTotal": 0,
        "types": []
    },
    "categoryAggregations": {
        "currentTotal": 0,
        "otherTotal": 0,
        "categories": []
    },
    "offerClasses": {
        "currentTotal": 1,
        "otherTotal": 0,
        "id": null,
        "types": [
            {
                "currentTotal": 1,
                "otherTotal": 0,
                "id": null,
                "type": "BPO",
                "subTypes": [
                    {
                        "currentTotal": 1,
                        "otherTotal": 0,

```

```

        "id": null,
        "name": "Internet"
    }
    ]
},
{
    "currentTotal": 1,
    "otherTotal": 0,
    "id": null,
    "type": "BPO",
    "subTypes": []
}
],
},
"filterAggregations": {
    "total": 0,
    "filterAggregations": []
},
},
"global": {
    "segmentAggregations": {
        "currentTotal": 0,
        "otherTotal": 0,
        "types": []
    },
    "categoryAggregations": {
        "currentTotal": 0,
        "otherTotal": 0,
        "categories": []
    },
    "recordClassAggregations": {
        "currentTotal": 35,
        "otherTotal": 0,
        "id": null,
        "types": [
            {
                "currentTotal": 16,
                "otherTotal": 0,
                "id": null,
                "type": "ELEMENTO_ABILITATO",
                "subTypes": []
            },
            {
                "currentTotal": 8,
                "otherTotal": 0,

```

```

        "id": null,
        "type": "OFFERTA",
        "subTypes": []
    },
    {
        "currentTotal": 4,
        "otherTotal": 0,
        "id": null,
        "type": "PO",
        "subTypes": []
    },
    {
        "currentTotal": 2,
        "otherTotal": 0,
        "id": null,
        "type": "BUNDLE",
        "subTypes": []
    },
    {
        "currentTotal": 2,
        "otherTotal": 0,
        "id": null,
        "type": "SERVIZIO_ABILITANTE",
        "subTypes": []
    },
    {
        "currentTotal": 2,
        "otherTotal": 0,
        "id": null,
        "type": "UNCLASSIFIED",
        "subTypes": []
    },
    {
        "currentTotal": 1,
        "otherTotal": 0,
        "id": null,
        "type": "BPO",
        "subTypes": [
            {
                "currentTotal": 1,
                "otherTotal": 0,
                "id": null,
                "name": "Internet"
            }
        ]
    }

```

```

    }
  }
}
]
]

```

Set the `aggOfferClasses` flag to `true` to aggregate all types and subtypes. For example:

```

{
  "aggOfferClasses": "true"
}

Response
{
  "totalHits": 37,
  "offers": {
    "offer": [
      {
        "offerId": "CS1_TFOM-5164",
        "offerDescription": "CS1_TFOM-5164"
      },
      {
        "offerId": "TESTIMPORT_PRODUCT5_TFOM-5280",
        "offerDescription": "TESTIMPORT_PRODUCT5_TFOM-5280"
      },
      {
        "offerId": "TESTIMPORT_PRODUCT10_TFOM-5280",
        "offerDescription": "TESTIMPORT_PRODUCT10_TFOM-5280"
      },
      {
        "offerId": "OB275-TIM_ELITE_TFOM-5280",
        "offerDescription": "OB275-TIM_ELITE_TFOM-5280"
      },
      {
        "offerId": "ODR62-TIM_YOUNG_MUSIC_FULL_TFOM-5280",
        "offerDescription": "ODR62-TIM_YOUNG_MUSIC_FULL_TFOM-5280"
      },
      {
        "offerId": "SIM_CARD_TFOM-5280",
        "offerDescription": "SIM_CARD_TFOM-5280"
      }
    ]
  }
}

```

```

        },
        {
            "offerId": "UC1_P2_TFOM-5280",
            "offerDescription": "P2_TFOM-5280"
        },
        {
            "offerId": "769214_ITIM_SAMSUNG_GALAXY_S6_32GB_
BLACK_TFOM-5280",
            "offerDescription": "769214_ITIM_SAMSUNG_GALAXY_S6_
32GB_BLACK_TFOM-5280"
        },
        {
            "offerId": "TESTIMPORT_PRODUCT2_TFOM-5280",
            "offerDescription": "TESTIMPORT_PRODUCT2, TEST
COMMA2_TFOM-5280"
        },
        {
            "offerId": "UC1_SA_TFOM-5280",
            "offerDescription": "ABILITANTE_TFOM-5280"
        }
    ]
},
"segmentAggregations": {
    "currentTotal": 0,
    "otherTotal": 0,
    "types": []
},
"categoryAggregations": {
    "currentTotal": 0,
    "otherTotal": 0,
    "categories": []
},
"offerClasses": {
    "currentTotal": 35,
    "otherTotal": 0,
    "id": null,
    "types": [
        {
            "currentTotal": 16,
            "otherTotal": 0,
            "id": null,
            "type": "ELEMENTO_ABILITATO",
            "subTypes": []
        },
    ],

```

```

{
  "currentTotal": 8,
  "otherTotal": 0,
  "id": null,
  "type": "OFFERTA",
  "subTypes": []
},
{
  "currentTotal": 4,
  "otherTotal": 0,
  "id": null,
  "type": "PO",
  "subTypes": []
},
{
  "currentTotal": 2,
  "otherTotal": 0,
  "id": null,
  "type": "BUNDLE",
  "subTypes": []
},
{
  "currentTotal": 2,
  "otherTotal": 0,
  "id": null,
  "type": "SERVIZIO_ABILITANTE",
  "subTypes": []
},
{
  "currentTotal": 2,
  "otherTotal": 0,
  "id": null,
  "type": "UNCLASSIFIED",
  "subTypes": []
},
{
  "currentTotal": 1,
  "otherTotal": 0,
  "id": null,
  "type": "BPO",
  "subTypes": [
    {
      "currentTotal": 1,
      "otherTotal": 0,

```

```

        "id": null,
        "name": "Internet"
      }
    ]
  },
  "filterAggregations": {
    "total": 0,
    "filterAggregations": []
  },
  "global": {
    "segmentAggregations": null,
    "categoryAggregations": null,
    "recordClassAggregations": null
  }
}

```

- **Segment**

You can browse for products using the segment modeled in the Product data model. This feature is implemented by using the segment name or type against the indexed products in the Offer Search Engine. For example, using the segment type or name-based filters you are presented with aggregate values such as the total number of offers for a specific or all segments or the offers for a particular segment. All filters might optionally contain both the segment name and type.

The product is related to the Segment type and the subtype by the `compatiblesegment` relationship rule in the Product data model. When a product with the segment relationship is indexed in the Offer Search Engine, guided browsing is enabled using the implemented aggregation API calls.

Using these models you can browse all segment types and you can use the total number of products that meet the specified criteria.

Set the `aggAllSegments` flag to `true` to aggregate all segment names and types. For example:

```

{
  "aggAllSegments": "true"
}
Response
{
  "totalHits": 37,

```

```

    "offers": {
      "offer": [
        {
          "offerId": "CS1_TFOM-5164",
          "offerDescription": "CS1_TFOM-5164"
        },
        {
          "offerId": "TESTIMPORT_PRODUCT5_TFOM-5280",
          "offerDescription": "TESTIMPORT_PRODUCT5_TFOM-5280"
        },
        {
          "offerId": "TESTIMPORT_PRODUCT10_TFOM-5280",
          "offerDescription": "TESTIMPORT_PRODUCT10_TFOM-5280"
        },
        {
          "offerId": "OB275-TIM_ELITE_TFOM-5280",
          "offerDescription": "OB275-TIM_ELITE_TFOM-5280"
        },
        {
          "offerId": "ODR62-TIM_YOUNG_MUSIC_FULL_TFOM-5280",
          "offerDescription": "ODR62-TIM_YOUNG_MUSIC_FULL_TFOM-5280"
        },
        {
          "offerId": "SIM_CARD_TFOM-5280",
          "offerDescription": "SIM_CARD_TFOM-5280"
        },
        {
          "offerId": "UC1_P2_TFOM-5280",
          "offerDescription": "P2_TFOM-5280"
        },
        {
          "offerId": "769214_ITIM_SAMSUNG_GALAXY_S6_32GB_BLACK_TFOM-5280",
          "offerDescription": "769214_ITIM_SAMSUNG_GALAXY_S6_32GB_BLACK_TFOM-5280"
        },
        {
          "offerId": "TESTIMPORT_PRODUCT2_TFOM-5280",
          "offerDescription": "TESTIMPORT_PRODUCT2, TEST COMMA2_TFOM-5280"
        },
        {

```

```

        "offerId": "UC1_SA_TFOM-5280",
        "offerDescription": "ABILITANTE_TFOM-5280"
    },
    {
        "offerId": "TIM_YOUNG_TFOM-5280",
        "offerDescription": "TIM_YOUNG_TFOM-5280"
    },
    {
        "offerId": "UC1_G2_TFOM-5280",
        "offerDescription": "Modified description"
    },
    {
        "offerId": "TIM_NEXT_TFOM-5280",
        "offerDescription": "TIM_NEXT_TFOM-5280"
    },
    {
        "offerId": "Prod132_TFOM-5425",
        "offerDescription": "Prod132_TFOM-5425"
    },
    {
        "offerId": "Prod133_TFOM-5425",
        "offerDescription": "Prod133_TFOM-5425"
    },
    {},
    {
        "offerId": "CS2_TFOM-5164",
        "offerDescription": "CS2_TFOM-5164"
    },
    {
        "offerId": "OFFERTA_MOBILE_BASE_TFOM-5280",
        "offerDescription": "OFFERTA_MOBILE_BASE_TFOM-5280"
    },
    {
        "offerId": "TESTIMPORT_PRODUCT7_TFOM-5280",
        "offerDescription": "TESTIMPORT_PRODUCT7_TFOM-5280"
    },
    {
        "offerId": "RT577-TIM_NEXT_E_2GB_LTE_A_209E_(RESTITUZIONE_A_12M)_TFOM-5280",
        "offerDescription": "RT577-TIM_NEXT_E_2GB_LTE_A_209E_(RESTITUZIONE_A_12M)_TFOM-5280"
    },
    {
        "offerId": "TESTIMPORT_PRODUCT8_TFOM-5280",

```

```

        "offerDescription": "TESTIMPORT_PRODUCT8_TFOM-5280"
    },
    {
        "offerId": "Prod131_TFOM-5425",
        "offerDescription": "Prod131_TFOM-5425"
    },
    {
        "offerId": "Prod108_TFOM-5410",
        "offerDescription": "Prod108_TFOM-5410"
    },
    {
        "offerId": "OPE_Prod_A",
        "offerDescription": "OPE_Prod_A"
    },
    {
        "offerId": "PROD_A1",
        "offerDescription": ""
    },
    {
        "offerId": "Prod21_TFOM-5375",
        "offerDescription": "Prod21_TFOM-5375"
    },
    {},
    {
        "offerId": "OFFER_MOBILE_BASE_TFOM-5280",
        "offerDescription": "OFFER_MOBILE_BASE_TFOM-5280"
    },
    {
        "offerId": "SMARTPHONE_TFOM-5280",
        "offerDescription": "SMARTPHONE_TFOM-5280"
    },
    {
        "offerId": "TESTIMPORT_PRODUCT6_TFOM-5280",
        "offerDescription": "TESTIMPORT_PRODUCT6_TFOM-5280"
    },
    {
        "offerId": "TESTIMPORT_PRODUCT1_TFOM-5280",
        "offerDescription": "TESTIMPORT_PRODUCT1, TEST
COMMA_TFOM-5280"
    },
    {
        "offerId": "SERVIZIO_MOBILE_PP_TFOM-5280",
        "offerDescription": "SERVIZIO_MOBILE_PP_TFOM-5280"
    },
    },

```

```

        {
            "offerId": "UC4_G1_TFOM-5280",
            "offerDescription": "UC4_G1_TFOM-5280"
        },
        {
            "offerId": "UC1_P1_TFOM-5280",
            "offerDescription": "P1_TFOM-5280"
        },
        {
            "offerId": "TESTIMPORT_PRODUCT3_TFOM-5280",
            "offerDescription": "TESTIMPORT_PRODUCT3_TFOM-5280"
        },
        {
            "offerId": "TESTIMPORT_PRODUCT4_TFOM-5280",
            "offerDescription": "TESTIMPORT_PRODUCT4_TFOM-5280"
        },
        {
            "offerId": "UC1_G1_TFOM-5280",
            "offerDescription": "G1_TFOM-5280"
        }
    ]
},
"segmentAggregations": {
    "currentTotal": 3,
    "otherTotal": 0,
    "types": [
        {
            "type": "SUB_BRAND",
            "currentTotal": 3,
            "otherTotal": 0,
            "names": [
                {
                    "currentTotal": 2,
                    "otherTotal": 0,
                    "name": "SP"
                },
                {
                    "currentTotal": 1,
                    "otherTotal": 0,
                    "name": "LOOP"
                }
            ]
        }
    ]
}
]

```

```

    },
    "categoryAggregations": {
        "currentTotal": 0,
        "otherTotal": 0,
        "categories": []
    },
    "offerClasses": {
        "currentTotal": 0,
        "otherTotal": 0,
        "id": null,
        "types": []
    },
    "filterAggregations": {
        "total": 0,
        "filterAggregations": []
    },
    "global": {
        "segmentAggregations": null,
        "categoryAggregations": null,
        "recordClassAggregations": null
    }
}

```

To browse specific segments with count - The search API gets the total number of products available for a segment name and type with optionally returning those products. This use case supports only single segment name/type search element to see the count. In case multiple segment names/types are present in the request, the count is shown for each segment name and type and the search result yields a union of all segments. This is enabled by turning the request attribute "aggregate=true" as in the following request sample:

```

{
    "aggAllCategories": "false",
    "aggAllSegments": "false",
    "searchOfferQuery": {
        "aggregate": "true",
        "globalAggregate": "false",
        "segments": [
            {
                "id": "test2",
                "type": "SUB_BRAND",
                "names": ["LOOP", "SP"]
            }
        ]
    }
}

```

```

    ]
  }
}

Response
{
  "totalHits": 2,
  "offers": {
    "offer": [
      {
        "offerId": "UC1_G2_TFOM-5280",
        "offerDescription": "Modified description"
      },
      {
        "offerId": "UC1_G1_TFOM-5280",
        "offerDescription": "G1_TFOM-5280"
      }
    ]
  },
  "segmentAggregations": {
    "currentTotal": 3,
    "otherTotal": 0,
    "types": [
      {
        "type": "SUB_BRAND",
        "currentTotal": 3,
        "otherTotal": 0,
        "names": [
          {
            "currentTotal": 1,
            "otherTotal": 0,
            "name": "LOOP"
          },
          {
            "currentTotal": 2,
            "otherTotal": 0,
            "name": "SP"
          }
        ]
      }
    ]
  }
},

```

```

    "categoryAggregations": {
      "currentTotal": 0,
      "otherTotal": 0,
      "categories": []
    },
    "offerClasses": {
      "currentTotal": 0,
      "otherTotal": 0,
      "id": null,
      "types": []
    },
    "filterAggregations": {
      "total": 0,
      "filterAggregations": []
    },
    "global": {
      "segmentAggregations": null,
      "categoryAggregations": null,
      "recordClassAggregations": null
    }
  }
}

```

Note: Attribute is not part of the segment and is not used with them.

You can browse specific segments along with other segments, and return offers and their count. Additionally it also returns all other segments present in all product models. This is useful particularly in use cases where there is a need to filter by particular segment and also maintain the global context of all segments(modeled in product model) and their counts:

```

Request
{
  "aggAllCategories":"false",
  "aggAllSegments":"false",
  "searchOfferQuery" : {
    "aggregate" : "true",
    "globalAggregate" : "true",
    "segments" : [
      {
        "id": "test2",
        "type": "SUB_BRAND",
        "names": ["LOOP","SP"]
      }
    ]
  }
}

```

```

    }
  ]
}

Response
{
  "totalHits": 2,
  "offers": {
    "offer": [
      {
        "offerId": "UC1_G2_TFOM-5280",
        "offerDescription": "Modified description"
      },
      {
        "offerId": "UC1_G1_TFOM-5280",
        "offerDescription": "G1_TFOM-5280"
      }
    ]
  },
  "segmentAggregations": {
    "currentTotal": 3,
    "otherTotal": 0,
    "types": [
      {
        "type": "SUB_BRAND",
        "currentTotal": 3,
        "otherTotal": 0,
        "names": [
          {
            "currentTotal": 1,
            "otherTotal": 0,
            "name": "LOOP"
          },
          {
            "currentTotal": 2,
            "otherTotal": 0,
            "name": "SP"
          }
        ]
      }
    ]
  }
},

```

```

    "categoryAggregations": {
      "currentTotal": 0,
      "otherTotal": 0,
      "categories": []
    },
    "offerClasses": {
      "currentTotal": 0,
      "otherTotal": 0,
      "id": null,
      "types": []
    },
    "filterAggregations": {
      "total": 0,
      "filterAggregations": []
    },
    "global": {
      "segmentAggregations": {
        "currentTotal": 3,
        "otherTotal": 0,
        "types": [
          {
            "type": "SUB_BRAND",
            "currentTotal": 3,
            "otherTotal": 0,
            "names": [
              {
                "currentTotal": 2,
                "otherTotal": 0,
                "name": "SP"
              },
              {
                "currentTotal": 1,
                "otherTotal": 0,
                "name": "LOOP"
              }
            ]
          }
        ]
      }
    },
    "categoryAggregations": {
      "currentTotal": 0,
      "otherTotal": 0,
      "categories": []
    },
  },

```

```
        "recordClassAggregations": {  
            "currentTotal": 0,  
            "otherTotal": 0,  
            "id": null,  
            "types": []  
        }  
    }  
}
```

Configuration Service API Samples

- [Actuator APIs](#)
- [Application Metadata APIs](#)
- [Application Properties APIs](#)
- [Configuration Files APIs](#)
- [Configuration Notification APIs](#)

Actuator APIs

- [Actuator web endpoint 'refresh'](#)
- [Get 'loggers-name'](#)
- [Post 'loggers-name'](#)
- [Actuator root web endpoint](#)
- [Actuator web endpoint 'loggers'](#)
- [Actuator web endpoint 'health'](#)
- [Actuator web endpoint 'health-path'](#)

i Note: Health and Info APIs are not secured so that they can interact with other services easily.

Actuator web endpoint 'refresh'

Method: HTTP POST

Endpoint: http://<host_address>:<port_address>/management/refresh

```
curl -X 'POST' \
  'http://localhost:9090/management/refresh' \
  -H 'accept: */*' \
  -d ''
```

Get loggers name

Method: HTTP GET

Endpoint: http://<host_address>:<port_address>/management/loggers/{name}

Parameters: name (mandatory)

You can find the possible name parameters in \$OPE_HOME/roles/<service-name>/standalone/config/logback_<service-name>.xml file.

Sample:

```
curl -X 'GET' \
  'http://localhost:9090/management/loggers/com.tibco.fom.bootstrap' \
  -H 'accept: */*'
```

Post loggers name

Get Configuration

Method: HTTP POST

Endpoint: http://<host_address>:<port_address>/management/loggers/{name}

Parameters: name (mandatory)

You can find the possible name parameters in \$OPE_HOME/roles/<service-name>/standalone/config/logback_<service-name>.xml file.

Sample:

```
curl -X 'POST' \
  'http://localhost:9090/management/loggers/com.tibco.fom.bootstrap' \
  -H 'accept: */*' \
  -H 'Content-Type: application/json' \
  -d '{
    "configuredLevel": "INFO, WARNING, ERROR"
  }
```

Actuator root web endpoint

Method: HTTP GET

Endpoint: http://<host_address>:<port_address>/management

```
curl -X 'GET' \
  'http://localhost:9090/management' \
  -H 'accept: */*'
```

Actuator web endpoint 'loggers'

Method: HTTP GET

Endpoint: http://<host_address>:<port_address>/management/loggers

Sample:

```
curl -X 'GET' \
  'http://localhost:9090/management/loggers' \
  -H 'accept: */*'
```

Actuator web endpoint 'health'

Method: HTTP GET

Endpoint: http://<host_address>:<port_address>/management/health

```
curl -X 'GET' \
  'http://localhost:9090/management/health' \
  -H 'accept: */*'
```

Actuator web endpoint 'health-path'

Method: HTTP GET

Endpoint: `http://<host_address>:<port_address>/management/health/{name}`

Parameters: name path parameter can have one of the following values:
readiness or liveness

Example:

```
curl -X 'GET' \
  'http://localhost:9090/management/health/readiness' \
  -H 'accept: */*'
```

Application Metadata APIs

Get supported files metadata

This API returns metadata information such as the list of supported files (both application properties files and configuration files), application Id, and application description for all applications supported by the Configurator.

Method: HTTP GET

Endpoint: `http://<host_address>:<port_address>/v1/configuration/configFilesMetadata`

Save supported files metadata

Method: HTTP POST

Endpoint: `http://<host_address>:<port_address>/v1/configuration/configFilesMetadata`

Example:

```
[
  {
    "applicationId": "string",
    "applicationDescription": "string",
    "appProperties": "string",
    "configurationFiles": [
      "string"
    ]
  }
]
```

Delete supported files metadata

Method: HTTP DELETE

Endpoint: `http://<host_address>:<port_address>/v1/configuration/configFilesMetadata/{applicationId}`

Parameters: applicationid (Select value from the drop-down menu)

Application Properties APIs

Get Application Properties by Application ID

Method: HTTP GET

Endpoint: `http://<host_address>:<port_address>/v1/configuration/{applicationId}`

Parameters: applicationid and applicationId (Select these mandatory values from the drop-down menu)

Update Application Properties for Application ID

Method: HTTP PUT

Endpoint: `http://<host_address>:<port_address>/v1/configuration/{applicationId}`

Parameters: applicationid and applicationId (mandatory values)

Example:

```
[
  {
    "propName": "string",
    "propValue": "string",
    "valueType": "Number, String, Boolean, Password",
    "allowedValues": [
      "string"
    ]
  }
]
```

Save Application Properties for Application ID

Method: HTTP POST

Endpoint: `http://<host_address>:<port_address>/v1/configuration/{applicationId}`

Parameters: applicationId and applicationId (mandatory values)

Example:

```
{
  "appDescription": "string",
  "categories": [
    {
      "categoryName": "string",
      "properties": [
        {
          "propName": "string",
          "propDescription": "string",
          "propValue": "string",
          "allowedValues": [
            "string"
          ],
          "valueType": "Number",
          "isTenantProperty": true
        }
      ]
    }
  ]
}
```

Delete Application Properties for Application ID

Method: HTTP DELETE

Endpoint: `http://<host_address>:<port_address>/v1/configuration/{applicationId}`

Parameters: applicationId and applicationId (mandatory values)

Example of deleteRequest:

```
[  
  "string"  
]
```

Replicate Tenant Properties

Method: HTTP POST

Endpoint: `http://<host_address>:<port_address>/v1/configuration/replicateTenantProperties/{sourceTenantId}`

Parameters: sourceTenantId and tenantID (mandatory values)

Get list of applications available with Configurator

Method: HTTP GET

Endpoint: `http://<host_address>:<port_address>/v1/configuration/availableApplications`

Configuration File APIs

Download Configuration File for Application ID

Method: HTTP GET

Endpoint: `http://<host_address>:<port_address>/v1/configuration/configFile/{applicationid}`

Parameters: applicationid (Select value from the drop-down menu)

Upload Configuration File for Application ID

Method: HTTP POST

Endpoint: `http://<host_address>:<port_address>/v1/configuration/configFile/{applicationid}`

Parameters: applicationid (Select value from the drop-down menu)

Request body: On the request body, browse and choose the required file to upload.

Delete Configuration Files for Application ID

Method: HTTP DELETE

Endpoint: `http://<host_address>:<port_address>/v1/configuration/{applicationid}`

Parameters: applicationid (Select value from the drop-down menu)

Example of deleteRequest:

```
[  
  "string"  
]
```

Configuration Notification APIs

GET Configurator notification API

Method: HTTP GET

Endpoint: `http://<host_address>:<port_address>/v1/configurator/events`

Shopping Cart APIs

The following APIs are used for various shopping cart functions.

- [Get Shopping Cart](#)
- [Create Shopping Cart](#)

- [Get Shopping Cart By Id](#)
- [Delete Shopping Cart By Id](#)
- [Update Shopping Cart](#)
- [Get Shopping Cart Update Event](#)
- [Get Shopping Cart Delete Event](#)
- [Get Shopping Cart Create Event](#)



Note: See the shopping cart samples from the \$OPE_HOME/samples/request_response/ShoppingCartRequest.json file.

Get Shopping Cart

Method: HTTP GET

Endpoint: *http://<host_address>:<port_address>/v1/shoppingCart*

Create Shopping Cart

Method: HTTP POST

Endpoint: *http://<host_address>:<port_address>/v1/shoppingCart*

Get Shopping Cart By Id

Method: HTTP GET

Endpoint: *http://<host_address>:<port_address>/v1/shoppingCart/{id}*

Parameters: `fields` and `Id` (mandatory)

Delete Shopping Cart By Id

Method: HTTP DELETE

Endpoint: *http://<host_address>:<port_address>/v1/shoppingCart/{id}*

Parameters: Id (mandatory)

Update Shopping Cart

Method: HTTP PATCH

Endpoint: *http://<host_address>:<port_address>/v1/shoppingCart/{id}*

Get Shopping Cart Update Event

Method: HTTP GET

Endpoint: *http://<host_address>:<port_address>/v1/shoppingCart/updateevent*

Get Shopping Cart Delete Event

Method: HTTP GET

Endpoint: *http://<host_address>:<port_address>/v1/shoppingCart/deleteevent*

Get Shopping Cart Create Event

Method: HTTP GET

Endpoint: *http://<host_address>:<port_address>/v1/shoppingCart/createevent*



Note:

The Shopping Cart Update Event, Delete Event, and Create Event APIs are Streaming APIs and are not yet supported on OpenAPI Specification 3.0 version. You can perform these API functions on a web browser.

To get notifications about any of the events (update, delete, or create), you must subscribe to the event.

Error Codes

The following error codes are currently implemented:

Code	Message
TIBCO-AFF-OPE-100000	An unexpected error occurred. Contact system Administrator for more details
TIBCO-AFF-OPE-100001	Product model [{0}] is not active or is not valid for the required date
TIBCO-AFF-OPE-100002	Product model not found for promotion [{0}]
TIBCO-AFF-OPE-100003	Inventory ID not supplied for action(mode) [{0}] and order line [{1}]
TIBCO-AFF-OPE-100004	Action Mode is not specified for action [{0}] for product [{1}] in orderline [{2}]
TIBCO-AFF-OPE-100005	Items in focus [{0}] could not be resolved for line/inventory id [{1}]
TIBCO-AFF-OPE-100006	Incompatible udf was found
TIBCO-AFF-OPE-100007	Mandatory UDF is missing
TIBCO-AFF-OPE-100008	UDF length exceeded
TIBCO-AFF-OPE-100009	UDF datatype does not conform to the configured regex
TIBCO-AFF-OPE-100010	UDF range conform to the configured range
TIBCO-AFF-OPE-100011	UDF range conform to the configured regex

Code	Message
TIBCO-AFF-OPE-100012	UDF record type value does not conform to the configured value in catalog
TIBCO-AFF-OPE-100013	Incompatible products found in offer
TIBCO-AFF-OPE-100014	Single Use Incompatible products found in offer
TIBCO-AFF-OPE-100015	Segment Incompatible products found in offer
TIBCO-AFF-OPE-100016	Group constraint violation. Count of products in group not between maximum and minimum.
TIBCO-AFF-OPE-100017	Group constraint violation. Count of products in group is greater than maximum.
TIBCO-AFF-OPE-100018	Group constraint violation. Count of products in group is lesser than minimum.
TIBCO-AFF-OPE-100019	Record constraint violation. Count of products lesser than minimum.
TIBCO-AFF-OPE-100020	Record constraint violation. Count of products greater than maximum.
TIBCO-AFF-OPE-100021	Record constraint violation. Count of products is not between maximum and minimum.
TIBCO-AFF-OPE-100022	Mandatory Link UDF missing
TIBCO-AFF-CART-100003	Shopping Cart for this cart id is not present
TIBCO-AFF-CART-100004	Cannot delete shopping cart 404 NOT_FOUND

TIBCO Documentation and Support Services

For information about this product, you can read the documentation, contact TIBCO Support, and join TIBCO Community.

How to Access TIBCO Documentation

Documentation for TIBCO products is available on the [TIBCO Product Documentation](#) website, mainly in HTML and PDF formats.

The [TIBCO Product Documentation](#) website is updated frequently and is more current than any other documentation included with the product.

Product-Specific Documentation

The following documentation for this product is available on the [TIBCO® Offer and Price Engine](#) documentation page:

- *TIBCO® Offer and Price Engine Release Notes*
- *TIBCO® Offer and Price Engine Installation and Configuration Guide*
- *TIBCO® Offer and Price Engine Concepts Guide*
- *TIBCO® Offer and Price Engine User Guide*
- *TIBCO® Offer and Price Engine Web services Guide*
- *TIBCO® Offer and Price Engine Security Guidelines*

How to Contact TIBCO Support

Get an overview of [TIBCO Support](#). You can contact TIBCO Support in the following ways:

- For accessing the Support Knowledge Base and getting personalized content about products you are interested in, visit the [TIBCO Support](#) website.
- For creating a Support case, you must have a valid maintenance or support contract with TIBCO. You also need a user name and password to log in to [TIBCO Support](#) website. If you do not have a user name, you can request one by clicking **Register** on

the website.

How to Join TIBCO Community

TIBCO Community is the official channel for TIBCO customers, partners, and employee subject matter experts to share and access their collective experience. TIBCO Community offers access to Q&A forums, product wikis, and best practices. It also offers access to extensions, adapters, solution accelerators, and tools that extend and enable customers to gain full value from TIBCO products. In addition, users can submit and vote on feature requests from within the [TIBCO Ideas Portal](#). For a free registration, go to [TIBCO Community](#).

Legal and Third-Party Notices

SOME TIBCO SOFTWARE EMBEDS OR BUNDLES OTHER TIBCO SOFTWARE. USE OF SUCH EMBEDDED OR BUNDLED TIBCO SOFTWARE IS SOLELY TO ENABLE THE FUNCTIONALITY (OR PROVIDE LIMITED ADD-ON FUNCTIONALITY) OF THE LICENSED TIBCO SOFTWARE. THE EMBEDDED OR BUNDLED SOFTWARE IS NOT LICENSED TO BE USED OR ACCESSED BY ANY OTHER TIBCO SOFTWARE OR FOR ANY OTHER PURPOSE.

USE OF TIBCO SOFTWARE AND THIS DOCUMENT IS SUBJECT TO THE TERMS AND CONDITIONS OF A LICENSE AGREEMENT FOUND IN EITHER A SEPARATELY EXECUTED SOFTWARE LICENSE AGREEMENT, OR, IF THERE IS NO SUCH SEPARATE AGREEMENT, THE CLICKWRAP END USER LICENSE AGREEMENT WHICH IS DISPLAYED DURING DOWNLOAD OR INSTALLATION OF THE SOFTWARE (AND WHICH IS DUPLICATED IN THE LICENSE FILE) OR IF THERE IS NO SUCH SOFTWARE LICENSE AGREEMENT OR CLICKWRAP END USER LICENSE AGREEMENT, THE LICENSE(S) LOCATED IN THE “LICENSE” FILE(S) OF THE SOFTWARE. USE OF THIS DOCUMENT IS SUBJECT TO THOSE TERMS AND CONDITIONS, AND YOUR USE HEREOF SHALL CONSTITUTE ACCEPTANCE OF AND AN AGREEMENT TO BE BOUND BY THE SAME.

This document is subject to U.S. and international copyright laws and treaties. No part of this document may be reproduced in any form without the written authorization of Cloud Software Group, Inc.

TIBCO, the TIBCO logo, and the TIBCO O logo are either registered trademarks or trademarks of Cloud Software Group, Inc. in the United States and/or other countries.

Java and all Java based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

This document includes fonts that are licensed under the SIL Open Font License, Version 1.1, which is available at: <https://scripts.sil.org/OFL>

Copyright (c) Paul D. Hunt, with Reserved Font Name Source Sans Pro and Source Code Pro.

All other product and company names and marks mentioned in this document are the property of their respective owners and are mentioned for identification purposes only.

This software may be available on multiple operating systems. However, not all operating system platforms for a specific software version are released at the same time. See the readme file for the availability of this software version on a specific operating system platform.

THIS DOCUMENT IS PROVIDED “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT.

THIS DOCUMENT COULD INCLUDE TECHNICAL INACCURACIES OR TYPOGRAPHICAL ERRORS. CHANGES ARE PERIODICALLY ADDED TO THE INFORMATION HEREIN; THESE CHANGES WILL BE INCORPORATED IN NEW EDITIONS OF THIS DOCUMENT. CLOUD SOFTWARE GROUP, INC. MAY MAKE IMPROVEMENTS AND/OR CHANGES IN THE PRODUCT(S) AND/OR THE PROGRAM(S) DESCRIBED IN THIS DOCUMENT AT ANY TIME.

THE CONTENTS OF THIS DOCUMENT MAY BE MODIFIED AND/OR QUALIFIED, DIRECTLY OR INDIRECTLY, BY OTHER DOCUMENTATION WHICH ACCOMPANIES THIS SOFTWARE, INCLUDING BUT NOT LIMITED TO ANY RELEASE NOTES AND "READ ME" FILES.

This and other products of Cloud Software Group, Inc. may be covered by registered patents. Please refer to TIBCO's Virtual Patent Marking document (<https://www.tibco.com/patents>) for details.

Copyright © 2010-2023. Cloud Software Group, Inc. All Rights Reserved.