

Price Soft /Price Shoes Web Services Document

Version 4.0

|  |
| --- |
| This document contains confidential and proprietary information of Price Shoes. It is furnished for internal purposes only. Without prior written permission of Price Shoes, this document and information contained herein may not be published, disclosed, or used for any other purpose. |

**Revision History & Approval Details**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Version No** | **Date** | **Brief description of changes** | **Author** | **Reviewed By** | **Approved By** |
| 1.0 | 12-Sep-2017 | Web Service Document | Mukesh | Vikas P |  |
| 2.0 | 15-Sep-2017 | Feedback from IT team | Francisco |  |  |
| 3.0 | 15-Sep-2017 | Explanation from PS team | Mukesh | Vikas P |  |
| 4.0 | 18-Sep-2017 | Added Order & Customer Service | Mukesh | Vikas P |  |

* **Contenido**

Contenido 3

1. Purpose 4

2. Introduction 4

2.1. Scope 4

2.2. Objective and Success Criteria 5

3. Technical Details 5

3.1. Inventory – e-Commerce Inventory 5

3.2. Inventory – Retail/Store Inventory 5

3.3. Product Price 6

3.4. Product Availability 6

3.5. Member Information by ID 7

3.6. Member Information by ID & DOB 7

3.7. Get All Members with Pagination 8

3.8. Update Member without Image 8

3.9. Update Member with Image 9

3.10. Get Balance 9

3.11. Create Customer/Member 9

3.12. Create Member Address List 9

3.13. Create Order 9

* **Purpose**

The purposes of this document to have the proper understanding of web-service in terms of business logic / requirement. Based on this document all the web-services will be designed and developed. Any change after approval of this document will be treated as change and will be considered in the send phase of the project.

* **Introduction**
* **Scope**

Price Shoes is doing an e-commerce business by selling the fashion products online. Price Shoes hast the online store which is currently running on Magento 1.12.0.2 and all the services which are communicating to ERP system I.e Price Soft are written in REST & SOAP.

Now as Price Shoes is upgrading the Magento 1 to Magento 2, all the SOAP services will be re-designed and developed in REST. Also all the store procedures which are written to pull the data will also be re-constructed and will used in REST services to fetch the same data.

Below are the services which are consider as in-scope:

* Inventory
* e-commerce Inventory
* Retail/Store Inventory
* Product Price
* Product Availability
* Socio / Affiliate
* Member Information by ID
* Member Information by ID & DOB
* Get All Member
* Update Member without Image
* Update Member with Image
* Member Balance
* Create Member with Address
* Order
* **Objective and Success Criteria**

Objective of this document to explain the way to do the web-service development based on the requirement provided by Price Shoes team also the deployment of web-services for production environment.

* **Technical Details**
* **Inventory – e-Commerce Inventory**
* **ServiceURL**: <http://106.51.0.187:86/rest/inventory/online/pages?page=1&size=10>
* **Type**: GET
* **Inputs**: page,size
* **Outputs**: storeId, skuId, stockLevel, size
* **Handler Method**: public ResponseEntity<?> listInventory(@RequestParam(value="page", defaultValue="1", required=false) int page,@RequestParam(value="size", defaultValue="10", required=false) int size,final UriComponentsBuilder uriBuilder,final HttpServletResponse httpServletResponse)
* **Description:** This service fetch the list of SKUs from **stores 1 and 5** from database and paginates the results based on the inputs parameters given in the request.
* **Magento Implementation:** Magento will execute this service daily once (time can be changed at any given point of time) at early hours i.e. 02:00 AM and fetch the response from the service and will update the Magento product inventory. As this services will return the complete inventory for the store so the batch-processing is introduced to make a request to Price Soft with pagination and limit of 1000 product for every call. This services will update the complete inventory from Price Soft to Magento in 30 min (approx)
* **Inventory – Retail/Store Inventory**
* **Service URL**: <http://106.51.0.187:86/rest/inventory/retail/18544>
* **Type**: GET
* **Inputs**: Id\_Art
* **Outputs**: storeId, skuId, stockLevel, size
* **Handler Method**: public ResponseEntity<?> getOnlineInventoryBySkuId(@PathVariable("id") Long id,@RequestParam(value="size", required=false) Double size)
* **Description:** This service fetch the stocklevel for each SKUs belong to specific **Id\_Art**. Input parameter in this service is only **Id\_Art** and response will be the stock/qty available for retail store.
* **Magento Implementation:** Magento will execute this service on demand only on PDP page. As this services will return the complete inventory for all the associated SKU for the id\_art, for each stores. This is product specific service so, always take only one SKU as input parameter.
* **Product Price**
* **Service URL**: <http://106.51.0.187:86/rest/product/prices>
* **Type**: GET
* **Inputs**: sys date (used in the query)
* **Outputs**: sku, caCveN, coCveStr, idArt, taCveN, preTmkN, precioCliente, precioSocio, precioDistributor, precioCentenario, precioGolden, precioOfertaCliente, precioOfertaSocio, precioOfertaDistributor, precioOfertaCentenario, precioOfertaGolden, precEsOferta
* **Handler Method**: public ResponseEntity<?> getPrice()
* **Description:** This service is used to get the prices of the products based on the system date i.e. server date which the service is deployed. This service will return all the products which has updated price in the back-end system. Prices in this service will be of two type and a flag which will help in identification of prices
* With Offer
* Without Offer

If the response contains the offer\_flag set to 1, it mean in response we will have both offer and without offer price for all type of customer i.e. Cliente, Socio, Golden, Centenario & Distribuidor

* **Magento Implementation:** Magento will execute this service based on the business requirement like once a day, twice a day or 4 times a day. Also this can be executed from Magento admin panel on demand bases. If the business will say more than 4 times that also we can set, but recommended frequency will be once a day.

Please make sure that we are not updating this manually from admin panel during the peak time because after changing the price, system has to re-index to avoid any issue.

* **Product Availability**
* **Service URL**: <http://106.51.0.187:86/rest/product/availability>
* **Type**: GET
* **Inputs**: sys date (used in the query)
* **Outputs**: SKU, Date
* **Handler Method**: public ResponseEntity<?> getAvailability()
* **Description:** This service is used to get the availability dates for the products based on the system date i.e. server date which the service is deployed. This service will return all the products which has updated available date in the back-end system.
* **Magento Implementation:** Magento will execute this service based on the business requirement like once a day, twice a day or 4 times a day etc. Also this can be executed from Magento admin panel on demand bases. If the business will say more than 4 times that also we can set, but recommended frequency will be once a day.

Please make sure that we are not triggering this service manually from admin panel during the peak time because after updating the product attribute, system has to re-index to avoid any issue.

* **Member Information by ID**
* **Service URL**: <http://106.51.0.187:86/rest/member/122004041143>
* **Type**: GET
* **Inputs**: memberId
* **Outputs**: member informations
* **Handler Method**: public ResponseEntity<?> getMemberById(@PathVariable("id") final String id,final HttpServletResponse httpServletResponse)
* **Description:** This service is used to get the member information from the price soft database based on the member id
* **Magento Implementation:** Magento will execute this service during registration when a customer already has the affiliate id then based on the ID, Magento will trigger this call and fill all the necessary details in the registration form to make handy for the end user
* **Member Information by ID & DOB**
* **Service URL:** <http://106.51.0.187:86/rest/member/122004041143/19-MAY-1979>
* **Type**: GET
* **Inputs**: memberId & date of birth
* **Outputs**: member informations
* **Handler Method**: public ResponseEntity<?> getMemberByIdAndDOB(@PathVariable("id") final String id, @PathVariable("dob") final Date dob, final HttpServletResponse httpServletResponse)
* **Description:** This service is used to get the member information from the price soft database based on the member id & date of birth
* **Magento Implementation:** Magento will execute this service on My Tool section of Affiliate, only one time and saved the information in the Magento database. Before executing this services system will check whether the information which we have to display in available in Magento or not. If Not, then only this call will be trigger from Magento. Same service will also return the image of the affiliate if available.
* **Get All Members with Pagination**
* **Service URL:** <http://106.51.0.187:86/rest/member/pages>
* **Type**: GET
* **Inputs**: page,size
* **Outputs**: members informations
* **Handler Method**: public ResponseEntity<?> findAll(@RequestParam(value="page", defaultValue="1", required=false) final int page,@RequestParam(value="size", defaultValue="10", required=false) final int size,final UriComponentsBuilder uriBuilder,final HttpServletResponse httpServletResponse)
* **Description:**  This service is used to get all the members available in the database by pagination.
* **Magento Implementation:** N/A
* **Update Member**
* **Service URL:** <http://106.51.0.187:86/rest/member/updateMember>
* **Type**: POST
* **Inputs**: Id (price soft customer id), firstName, lastName, motherName, email, dateOfBirth, image
* **Outputs**: status of member updation
* **Handler Method**: public ResponseEntity<?> updateMember(@RequestParam("id") String id,@RequestParam("firstName") String firstName, @RequestParam("lastName") String lastName, @RequestParam("email") String email,@RequestParam("motherName") String motherName, @RequestParam("dateOfBirth") String dateOfBirth)
* **Description:**  This service is used to update the member information by passing the above details along with image. Only the details which is passed as input parameter will be updated, other details will not be updated. We have created this service because only updating image is taking time and other details are updating in fraction section and we don’t want the customer to wait for few seconds if they are not updating the image.
* **Get Balance**
* **Service URL:** <http://106.51.0.187:86/rest/member/balance/122003180026>
* **Type**: GET
* **Inputs**: Affiliate Id/memberId (price soft customer id)
* **Outputs**: vales, depositos, socios, nc
* **Handler Method**: public ResponseEntity<?> getBalance(@PathVariable("id") String id)
* **Description:**  This service is used to get the balance information about the customer whether the customer is of type Socio or Cliente.
* **Magento Implementation:** This service will be invoked whenever customer is logging in the website to update the balance information which will be used during the payment.
* **Create Customer/Member**
* **Service URL:** <http://106.51.0.187:86/rest/member/createMember>
* **Type**: POST
* **Inputs**: socio:{nombre, paisId, estadoId, apellidoPaterno, apellidoMaterno, medioId, submedioId, tipoSocio, sexo, email, dob, activo, anioAfiliacion, sucursalAfiliacion, tipoSocioCliente, direcciones:{direccion:[{calle, numero, colonia, municipioId, ciudad, codigoPostal, telefono1, paisId, estadoId}]}},psSocio:{soNomStr, soApatStr, soAmatStr, soFnacDt, soFregDt, soCalleStr, soNumStr, soColStr, soNcredN, muCveN, soCdStr , soCpStr, soTel1Str, soEmailStr, soSexoStr, soEstStr, soTipoStr, soCve1Str , soLada1Str, soCve2Str, soLada2Str, usrCvePstr, soVigDt, edCveN, tiCveN, paCveN, tsCveN}}
* **Outputs**: status and id (if member created successfully)
* **Handler Method**: public ResponseEntity<?> createCustomer(@RequestBody MemberBean memberBean)
* **Description:**  This service is used to create the member with address in pricesoft database. Member can be of type Cliente ‘C’ or Socio ‘S’
* **Magento Implementation:** This service will be invoked after customer is placing the order first time on the website and by passing the required request parameter service will create a new customer in Pricesoft, and in response will return the price soft member id and which will be used in creating the order on Pricesoft. Also this id will be stored in Magento for the customer for future orders. If this Id is already available for the customer then there no call to create customer during order in Magento.

{"socio":{"soNomStr":"BETARIZ","soApatStr":"MARTINEZ","soAmatStr":"VILLAGRAN","soFnacDt":"1964-03-23","soFregDt":"2004-12-20","soCalleStr":"EMILIANO ZAPATA","soNumStr":"10","soColStr":"SAN LORENZO","soNcredN":"5","muCveN":"16","soCdStr":"NEXICOZ","soCpStr":"16018","soTel1Str":"6418990","soEmailStr":"BETARIZ@hotmail.com","soSexoStr":"F","soEstStr":"A","soTipoStr":"R","soCve1Str":"01","soLada1Str":"555","soCve2Str":"01","soLada2Str":"555","usrCvePstr":"JONATHANAP","soVigDt":"2015-03-16","edCveN":"9","tiCveN":"5","paCveN":"1","tsCveN":"19"},"addresses":[{"edCveN":"15","muCveN":"33","paCveN":"1","sodCalleStr":"RIO HONDA","sodCdStr":"ECATEPEC DE MORELOS","sodColStr":"LA PRESA","sodCpStr":"55400","sodEstStr":"A","sodNumStr":"SN","sodObsStr":"","sodTel1Str":"51262106","sodTel2Str":""}],"image":""}

* **Create Order**
* **Service URL:** <http://106.51.0.187:86/rest/order/saveOrder>
* **Type**: POST
* **Inputs**: pedidoLite:{pedidoIdPs, guia, estatus, referencia}, agregarGuia, actualizarEstatus, pedido:{socioIdPs, direccionEnvio:{idDireccionPs, calle, numero, colonia, municipioId, ciudad, codigoPostal, telefono1, telefono2, paisId, estadoId, observaciones}, direccionPago:{ idDireccionPs, calle, numero, colonia, municipioId, ciudad , codigoPostal, telefono1, telefono2, paisId , estadoId, observaciones}, tipoPedido, importeTotal, gastosEnvio, seguroMensajeria, descuentoPedido, items:{item:[{articulo:{idArticuloPs, talla, precio, tallaEQ, apartado}, cantidad, descuento}]}, formasPagos:{item:[{formaPago, cantidad, referencia}]}, formaPedido, pedidoMagento }, formaPagoPedido:{formaPago, cantidad, referencia}
* **Outputs**: psPedtmkl, formaPagoPedido: {formaPago, cantidad, referencia}, value, bandera, banderaSaldo
* **Handler Method**: public ResponseEntity<PedidoRespuesta> savePedido(@RequestBody PedidoConsulta consulta)
* **Description:**  This service is used to create the new order in Pricesoft database.
* **Magento Implementation:** This service will be invoked immediately after placing the order in Magento. Request will be created by considering the Pricesoft parameter like Country Id, State Id, Muncipality Id etc with Item Information, Billing & Shipping address and the payment information. Successful creation of order in Magento will return ps\_order\_id which will be stored in Magento for future reference.

{"pedidoLite":{"pedidoIdPs":"","guia":"","estatus":"","referencia":""},"agregarGuia":"true", "actualizarEstatus":"true","pedido":{"socioIdPs":"**050500053946**", "direccionEnvio":{"idDireccionPs":"336","calle":"Av Los insurgentes 116", "numero":"116", "colonia":"insurgentes", "municipioId":"123", "ciudad":"Mexico City", "codigoPostal":"06600", "telefono1":"5540363806", "telefono2":"","paisId":"1","estadoId":"09","observaciones":""}, "direccionPago":{"idDireccionPs":"337", "calle":"Av los insurgentes 116 ","numero":"116", "colonia":"Insurgentes", "municipioId":"1234", "ciudad":"Mexico City","codigoPostal":"06600", "telefono1":"5540363806", "telefono2":"", "paisId":"1","estadoId":"09","observaciones":""},"tipoPedido":"M","importeTotal":"227.00", "gastosEnvio":"5.5","seguroMensajeria":"2.5", "descuentoPedido":"0","items":{"item":[ {"articulo":{"idArticuloPs":"109707","talla":"22.5","precio":"219.00","tallaEQ":"22.5","apartado":"a"}, "cantidad":"1","descuento":"0"}]},"formasPagos":{"item":[{"formaPago":"paypal", "cantidad":"1", "referencia":""}]},"formaPedido":"M","pedidoMagento":"**1000836905**"}, "formaPagoPedido":{"formaPago":"paypal", "cantidad":"1","referencia":""}}