|  |  |
| --- | --- |
|  |  |
|  | LATBC  Eduardo Rodriguez Arreola |

|  |
| --- |
| **[Manual técnico]** |
| Documento que contiene los componentes que se utilizaron para la creación de los servicios |

# 

Contenido

[Control del Documento 4](#_Toc378355555)

[Control de Cambios 4](#_Toc378355556)

[Revisores 4](#_Toc378355557)

[Aprobadores 4](#_Toc378355558)

[Introducción 5](#_Toc378355559)

[Objetivo 5](#_Toc378355560)

[Ambiente 5](#_Toc378355561)

[Generalidades 5](#_Toc378355562)

[Servicio GetLoyaltyInfo 7](#_Toc378355563)

[Diagrama de servicio 7](#_Toc378355564)

[Descripción de Etapas Operación GetCardStatus 8](#_Toc378355565)

[Descripción de Etapas Operación GetBenefits 12](#_Toc378355566)

[Descripción de Etapas Operación GetCashbackInfo 17](#_Toc378355567)

[Descripción de Etapas Operación GetCashbackDetails 21](#_Toc378355568)

[Servicio GetLoyaltyInfoLocal 26](#_Toc378355569)

[Diagrama de servicio 26](#_Toc378355570)

[Descripción de Etapas Operación GetCardStatus 27](#_Toc378355571)

[Descripción de Etapas Operación GetBenefits 31](#_Toc378355572)

[Descripción de Etapas Operación GetCashbackInfo 36](#_Toc378355573)

[Descripción de Etapas Operación GetCashbackDetails 40](#_Toc378355574)

[Recursos del Servicio 46](#_Toc378355575)

[Esquemas XSD 46](#_Toc378355576)

[WSDL 46](#_Toc378355577)

[XQUERY 47](#_Toc378355578)

[Business Service 49](#_Toc378355579)

[Service Account 49](#_Toc378355580)

[AD 49](#_Toc378355581)

[Proxy Service 49](#_Toc378355582)

[Contrato de Servicio 50](#_Toc378355583)

# Control del Documento

## Control de Cambios

37

| Fecha | Autor | Versión | Descripción del Cambio |
| --- | --- | --- | --- |
| 24/01/2014 | Eduardo Rodriguez | 1.0 | Creación del Documento |
|  |  |  |  |
|  |  |  |  |

## Revisores

| Nombre | Rol | Área | Firma |
| --- | --- | --- | --- |
|  |  |  |  |
| Nora Gabriela Arroyo Carranza | Manager | LATBC |  |
|  |  |  |  |
|  |  |  |  |

## Aprobadores

| Nombre | Rol | Área | Firma |
| --- | --- | --- | --- |
|  |  |  |  |
| Nora Gabriela Arroyo Carranza | Manager | LATBC |  |
|  |  |  |  |

# Introducción

Como parte del proceso Tigo App, se desarrollo el servicio GetLoyaltyInfo en dos versiones, una para consumo interno y otra para consumo externo.

|  |  |
| --- | --- |
| Servicio | Detalle |
| GetLoyaltyInfo: | Obtiene información del abonado en el programa de lealtad, este servicio será consumido por sistemas externos |
| GetLoyaltyInfoLocal: | Obtiene información del abonado en el programa de lealtad, este servicio será consumido por sistemas internos |

# Objetivo

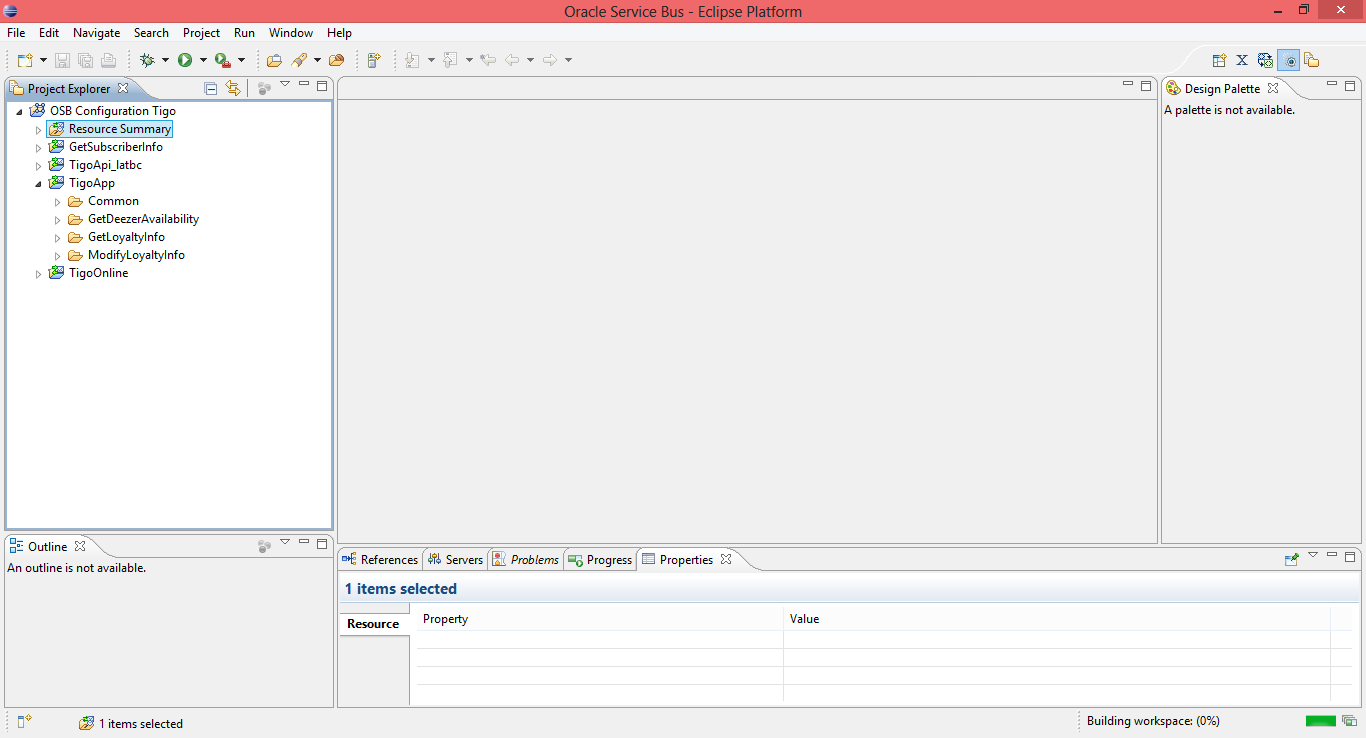
El objetivo de este documento es presentar los componentes que conforman el servicio GetLoyaltyInfo.

# Ambiente

La plataforma de integración en la cual fueron desarrollados los servicios es Oracle Service Bus 11.1.1.5, la cual fue proporcionada por Tigo, así mismo se utilizó como IDE de desarrollo OEPE 11.1.1.5.

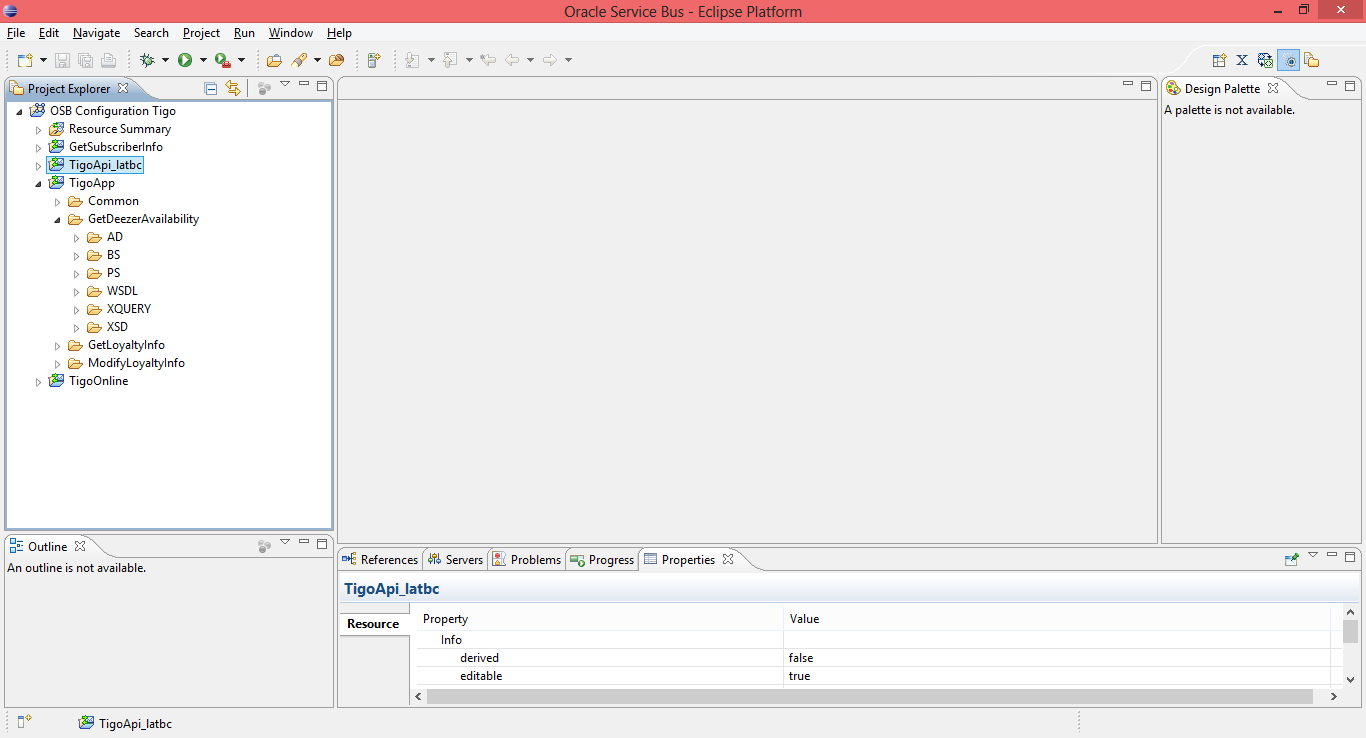
# Generalidades

La estructura que se siguió para el proyecto en el OSB se muestra a continuación



Donde:

* **TigoApp**: es la carpeta agrupadora del proyecto
* **Common**: almacena la estructura de carpetas para los recursos comunes, en este caso almacena los wsdl, xsd, business service, etc correspondientes a los servicios de MapResponse, AuditProducerOSB y ErrorProducerOSB
* **GetLoyaltyInfo**: almacena la estructura de carpetas donde se almacena los wsdl, xsd, business service, etc. para los servicios GetLoyaltyInfo y GetLoyaltyInfoLocal.



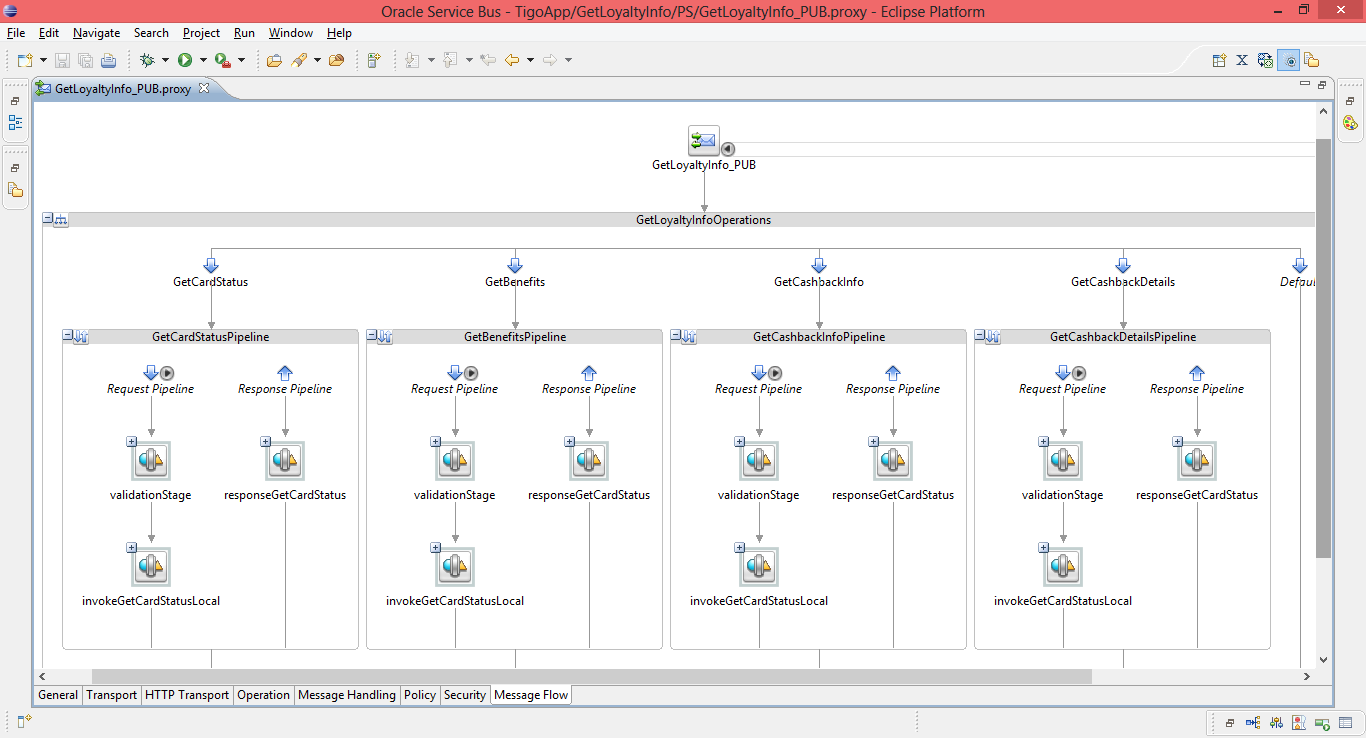
Donde:

* **AD:** almacena los archivos Alert Destination utilizado para la configuración de los SLA’s de los servicios.
* **ADAPTER**: almacena los adaptadores \*
* **BS**: almacena los Business Services generados para el consumo del servicio
* **JAR**: almacena los jars que serán usados en el servicio \*
* **PS**: almacena los proxy services que serán consumidos
* **WSDL**: almacena las wsdl de los servicios invocados por los proxys desarrollados
* **XQUERY**: almacena los archivos xquery utilizados en la integración
* **XSD**: almacena los esquemas que son usados por los servicios.

(\*Capeta con recursos no implementados en el servicio)

# Servicio GetLoyaltyInfo

## Diagrama de servicio



## Descripción de Etapas Operación GetCardStatus

A continuación se describen las etapas que conforman las actividades del flujo GetCardStatusPipeline.

|  |  |  |
| --- | --- | --- |
| **Actividad** | **Descripción** | **Diagrama** |
| validationStage | Se realiza la validación del elemento de entrada header*, estos deben corresponder a la definición en el XSD RequestHeaderV3.xsd*.  Se realiza la validación del elemento de entrada body*, estos deben corresponder a la definición en el XSD GetLoyaltyInfoRequest.xsd*  Se asigna el Request Header del servicio a la variable “requestHeader”. |  |
| invokeGetCardStatusLocal | Se asigna la variable varRequestBody. Esta es tomada del Request del servicio.  Se leen las credenciales ubicadas en el Service Account GetLoyaltyInfoLocal.sa  Se asigna la variable varRequestHeader. Esta es generada con las credenciales previamente obtenidas.  Se invoca el servicio “GetLoyaltyInfoLocal” con la operación “GetCardStatus” |  |
| responseGetCardStatus | Se valida que la respuesta GetLoyaltyInfoResponse del servicio “GetLoyaltyInfoLocal” no sea un elemento header.  En caso de ser un elemento del tipo ResponseHeader se reemplaza el Header con la respuesta del servicio “GetLoyaltyInfoLocal” y se elimina todo el contenido de la variable body.  En caso de no ser un elemento del tipo ResponseHeader se reemplaza el Body con la respuesta del servicio “GetLoyaltyInfoLocal” y se asigna la variable serviceHeaderResponse con el Xquery TransformOKToResponseHeader.xq para posteriormente reemplazarlo en el Header del servicio. |  |
| getError | Se recupera el código de error.  fn:data($fault/ctx:errorCode) |  |
| callMapResponse | Se realiza la asignación del mensaje de Request para el servicio MapResponse.  Se invoca el servicio MapResponse para obtener la descripción del código de error arrojado en RequestPipeLine.  Se realiza una validación para saber si el servicio devolvió la descripción del código de error, en caso contrario se asigna por default el código DES-999 a la variable de respuesta del servicio. |  |
| assignFaultResponse | Se asigna la variable de respuesta faultResponse (del tipo ResponseHeader) usando el xquery TransformMapResponse ubicado en la carpeta Common/XQUERY, pasando como parámetro de entrada la respuesta del servicio MapResponse |  |
| invokeErrorProducer | Se realiza la asignación del mensaje de Request para el servicio ErrorProducerOSB Se utiliza el xquery TransformToErrorRequest ubicado en la carpeta Common/XQUERY.  Se invoca el servicio ErrorProducerOSB para registrar el error ocurrido a nivel de RequestPipeline, haciendo replace del mensaje Request en la variable body. |  |
| responsePipeLine1 | Se realiza un replace con la variable faultResponse como contenido del nodo de respuesta del servicio (variable Header del servicio).  Se elimina el contenido del nodo Body.  Se responde con bandera en *Success*. |  |
| saveHeaderResponse | Se asigna la variable “faultResponse” con xquery TransformFaultToResponseHeader ubicado en la carpeta Common/XQUERY. |  |
| invokeErrorProducer | Se realiza la asignación del mensaje de Request para el servicio ErrorProducerOSB Se utiliza el xquery TransformToErrorRequest ubicado en la carpeta Common/XQUERY.  Se invoca el servicio ErrorProducerOSB para registrar el error ocurrido a nivel de RequestPipeline, haciendo replace del mensaje Request en la variable body. |  |
| responseErrorService | Se realiza un replace con la variable faultResponse como contenido del nodo de respuesta del servicio (variable Header del servicio).  Se elimina el contenido del nodo Body.  Se responde con bandera en *Failure*. |  |

## Descripción de Etapas Operación GetBenefits

A continuación se describen las etapas que conforman las actividades del flujo GetBenefitsPipeline.

|  |  |  |
| --- | --- | --- |
| **Actividad** | **Descripción** | **Diagrama** |
| validationStage | Se realiza la validación del elemento de entrada header*, estos deben corresponder a la definición en el XSD RequestHeaderV3.xsd*.  Se realiza la validación del elemento de entrada body*, estos deben corresponder a la definición en el XSD GetLoyaltyInfoRequest.xsd*  Se asigna el Request Header del servicio a la variable “requestHeader”. |  |
| invokeGetBenefitsLocal | Se asigna la variable varRequestBody. Esta es tomada del Request del servicio.  Se leen las credenciales ubicadas en el Service Account GetLoyaltyInfoLocal.sa  Se asigna la variable varRequestHeader. Esta es generada con las credenciales previamente obtenidas.  Se invoca el servicio “GetLoyaltyInfoLocal” con la operación “GetBenefits” |  |
| responseGetBenefits | Se valida que la respuesta GetLoyaltyInfoResponse del servicio “GetLoyaltyInfoLocal” no sea un elemento header.  En caso de ser un elemento del tipo ResponseHeader se reemplaza el Header con la respuesta del servicio “GetLoyaltyInfoLocal” y se elimina todo el contenido de la variable body.  En caso de no ser un elemento del tipo ResponseHeader se reemplaza el Body con la respuesta del servicio “GetLoyaltyInfoLocal” y se asigna la variable serviceHeaderResponse con el Xquery TransformOKToResponseHeader.xq para posteriormente reemplazarlo en el Header del servicio. |  |
| getError | Se recupera el código de error.  fn:data($fault/ctx:errorCode) |  |
| callMapResponse | Se realiza la asignación del mensaje de Request para el servicio MapResponse.  Se invoca el servicio MapResponse para obtener la descripción del código de error arrojado en RequestPipeLine.  Se realiza una validación para saber si el servicio devolvió la descripción del código de error, en caso contrario se asigna por default el código DES-999 a la variable de respuesta del servicio. |  |
| assignFaultResponse | Se asigna la variable de respuesta faultResponse (del tipo ResponseHeader) usando el xquery TransformMapResponse ubicado en la carpeta Common/XQUERY, pasando como parámetro de entrada la respuesta del servicio MapResponse |  |
| invokeErrorProducer | Se realiza la asignación del mensaje de Request para el servicio ErrorProducerOSB Se utiliza el xquery TransformToErrorRequest ubicado en la carpeta Common/XQUERY.  Se invoca el servicio ErrorProducerOSB para registrar el error ocurrido a nivel de RequestPipeline, haciendo replace del mensaje Request en la variable body. |  |
| responsePipeLine1 | Se realiza un replace con la variable faultResponse como contenido del nodo de respuesta del servicio (variable Header del servicio).  Se elimina el contenido del nodo Body.  Se responde con bandera en *Success*. |  |
| saveHeaderResponse | Se asigna la variable “faultResponse” con xquery TransformFaultToResponseHeader ubicado en la carpeta Common/XQUERY. |  |
| invokeErrorProducer | Se realiza la asignación del mensaje de Request para el servicio ErrorProducerOSB Se utiliza el xquery TransformToErrorRequest ubicado en la carpeta Common/XQUERY.  Se invoca el servicio ErrorProducerOSB para registrar el error ocurrido a nivel de RequestPipeline, haciendo replace del mensaje Request en la variable body. |  |
| responseErrorService | Se realiza un replace con la variable faultResponse como contenido del nodo de respuesta del servicio (variable Header del servicio).  Se elimina el contenido del nodo Body.  Se responde con bandera en *Failure*. |  |

## Descripción de Etapas Operación GetCashbackInfo

A continuación se describen las etapas que conforman las actividades del flujo GetCashbackInfoPipeline.

|  |  |  |
| --- | --- | --- |
| **Actividad** | **Descripción** | **Diagrama** |
| validationStage | Se realiza la validación del elemento de entrada header*, estos deben corresponder a la definición en el XSD RequestHeaderV3.xsd*.  Se realiza la validación del elemento de entrada body*, estos deben corresponder a la definición en el XSD GetLoyaltyInfoRequest.xsd*  Se asigna el Request Header del servicio a la variable “requestHeader”. |  |
| invokeGetCashbackInfoLocal | Se asigna la variable varRequestBody. Esta es tomada del Request del servicio.  Se leen las credenciales ubicadas en el Service Account GetLoyaltyInfoLocal.sa  Se asigna la variable varRequestHeader. Esta es generada con las credenciales previamente obtenidas.  Se invoca el servicio “GetLoyaltyInfoLocal” |  |
| responseGetCashbackInfo | Se valida que la respuesta GetLoyaltyInfoResponse del servicio “GetLoyaltyInfoLocal” no sea un elemento header.  En caso de ser un elemento del tipo ResponseHeader se reemplaza el Header con la respuesta del servicio “GetLoyaltyInfoLocal” y se elimina todo el contenido de la variable body.  En caso de no ser un elemento del tipo ResponseHeader se reemplaza el Body con la respuesta del servicio “GetLoyaltyInfoLocal” y se asigna la variable serviceHeaderResponse con el Xquery TransformOKToResponseHeader.xq para posteriormente reemplazarlo en el Header del servicio. |  |
| getError | Se recupera el código de error.  fn:data($fault/ctx:errorCode) |  |
| callMapResponse | Se realiza la asignación del mensaje de Request para el servicio MapResponse.  Se invoca el servicio MapResponse para obtener la descripción del código de error arrojado en RequestPipeLine.  Se realiza una validación para saber si el servicio devolvió la descripción del código de error, en caso contrario se asigna por default el código DES-999 a la variable de respuesta del servicio. |  |
| assignFaultResponse | Se asigna la variable de respuesta faultResponse (del tipo ResponseHeader) usando el xquery TransformMapResponse ubicado en la carpeta Common/XQUERY, pasando como parámetro de entrada la respuesta del servicio MapResponse |  |
| invokeErrorProducer | Se realiza la asignación del mensaje de Request para el servicio ErrorProducerOSB Se utiliza el xquery TransformToErrorRequest ubicado en la carpeta Common/XQUERY.  Se invoca el servicio ErrorProducerOSB para registrar el error ocurrido a nivel de RequestPipeline, haciendo replace del mensaje Request en la variable body. |  |
| responsePipeLine1 | Se realiza un replace con la variable faultResponse como contenido del nodo de respuesta del servicio (variable Header del servicio).  Se elimina el contenido del nodo Body.  Se responde con bandera en *Success*. |  |
| saveHeaderResponse | Se asigna la variable “faultResponse” con xquery TransformFaultToResponseHeader ubicado en la carpeta Common/XQUERY. |  |
| invokeErrorProducer | Se realiza la asignación del mensaje de Request para el servicio ErrorProducerOSB Se utiliza el xquery TransformToErrorRequest ubicado en la carpeta Common/XQUERY.  Se invoca el servicio ErrorProducerOSB para registrar el error ocurrido a nivel de RequestPipeline, haciendo replace del mensaje Request en la variable body. |  |
| responseErrorService | Se realiza un replace con la variable faultResponse como contenido del nodo de respuesta del servicio (variable Header del servicio).  Se elimina el contenido del nodo Body.  Se responde con bandera en *Failure*. |  |

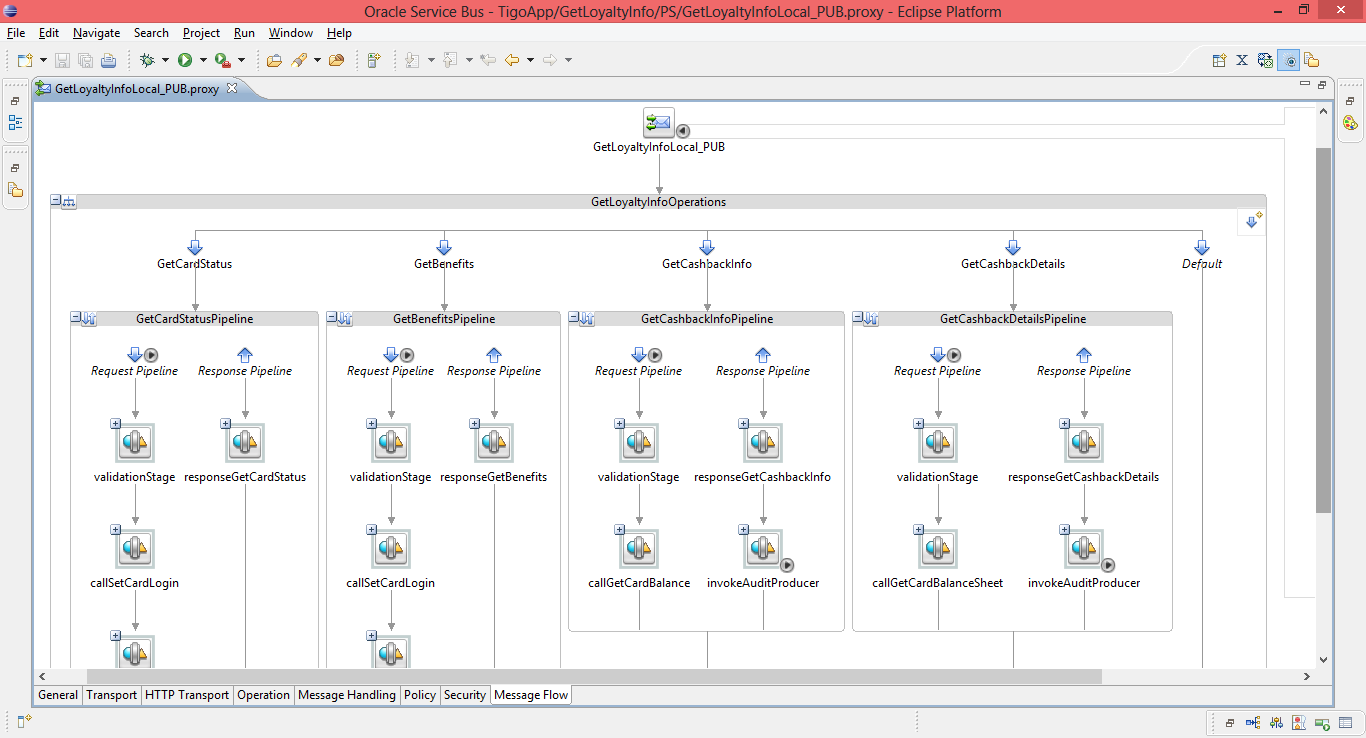
## Descripción de Etapas Operación GetCashbackDetails

A continuación se describen las etapas que conforman las actividades del flujo GetCashbackDetailsPipeline.

|  |  |  |
| --- | --- | --- |
| **Actividad** | **Descripción** | **Diagrama** |
| validationStage | Se realiza la validación del elemento de entrada header*, estos deben corresponder a la definición en el XSD RequestHeaderV3.xsd*.  Se realiza la validación del elemento de entrada body*, estos deben corresponder a la definición en el XSD GetLoyaltyInfoRequest.xsd*  Se asigna el Request Header del servicio a la variable “requestHeader”. |  |
| invokeGetCashbackDetailsLocal | Se asigna la variable varRequestBody. Esta es tomada del Request del servicio.  Se leen las credenciales ubicadas en el Service Account GetLoyaltyInfoLocal.sa  Se asigna la variable varRequestHeader. Esta es generada con las credenciales previamente obtenidas.  Se invoca el servicio “GetLoyaltyInfoLocal” |  |
| responseGetCashbackDetails | Se valida que la respuesta GetLoyaltyInfoResponse del servicio “GetLoyaltyInfoLocal” no sea un elemento header.  En caso de ser un elemento del tipo ResponseHeader se reemplaza el Header con la respuesta del servicio “GetLoyaltyInfoLocal” y se elimina todo el contenido de la variable body.  En caso de no ser un elemento del tipo ResponseHeader se reemplaza el Body con la respuesta del servicio “GetLoyaltyInfoLocal” y se asigna la variable serviceHeaderResponse con el Xquery TransformOKToResponseHeader.xq para posteriormente reemplazarlo en el Header del servicio. |  |
| getError | Se recupera el código de error.  fn:data($fault/ctx:errorCode) |  |
| callMapResponse | Se realiza la asignación del mensaje de Request para el servicio MapResponse.  Se invoca el servicio MapResponse para obtener la descripción del código de error arrojado en RequestPipeLine.  Se realiza una validación para saber si el servicio devolvió la descripción del código de error, en caso contrario se asigna por default el código DES-999 a la variable de respuesta del servicio. |  |
| assignFaultResponse | Se asigna la variable de respuesta faultResponse (del tipo ResponseHeader) usando el xquery TransformMapResponse ubicado en la carpeta Common/XQUERY, pasando como parámetro de entrada la respuesta del servicio MapResponse |  |
| invokeErrorProducer | Se realiza la asignación del mensaje de Request para el servicio ErrorProducerOSB Se utiliza el xquery TransformToErrorRequest ubicado en la carpeta Common/XQUERY.  Se invoca el servicio ErrorProducerOSB para registrar el error ocurrido a nivel de RequestPipeline, haciendo replace del mensaje Request en la variable body. |  |
| responsePipeLine1 | Se realiza un replace con la variable faultResponse como contenido del nodo de respuesta del servicio (variable Header del servicio).  Se elimina el contenido del nodo Body.  Se responde con bandera en *Success*. |  |
| saveHeaderResponse | Se asigna la variable “faultResponse” con xquery TransformFaultToResponseHeader ubicado en la carpeta Common/XQUERY. |  |
| invokeErrorProducer | Se realiza la asignación del mensaje de Request para el servicio ErrorProducerOSB Se utiliza el xquery TransformToErrorRequest ubicado en la carpeta Common/XQUERY.  Se invoca el servicio ErrorProducerOSB para registrar el error ocurrido a nivel de RequestPipeline, haciendo replace del mensaje Request en la variable body. |  |
| responseErrorService | Se realiza un replace con la variable faultResponse como contenido del nodo de respuesta del servicio (variable Header del servicio).  Se elimina el contenido del nodo Body.  Se responde con bandera en *Failure*. |  |

# Servicio GetLoyaltyInfoLocal

## Diagrama de servicio



## Descripción de Etapas Operación GetCardStatus

A continuación se describen las etapas que conforman las actividades del flujo GetCardStatusPipeline.

|  |  |  |
| --- | --- | --- |
| **Actividad** | **Descripción** | **Diagrama** |
| validationStage | Se realiza la validación del elemento de entrada body*, estos deben corresponder a la definición en el XSD GetLoyaltyInfoRequest.xsd*  Se asigna el Request Body del servicio a la variable “serviceRequest”. |  |
| callSetCardLogin | Se lee el Archivo de Configuración.  Se genera el Request setCardLoginRequest con xquery TransformToCRMsetCardLoginRq.xq para invocar el servicio TIGOCRMOPERACIONES.  Se invoca la operación “setCardLogin” del servicio “TIGOCRMOPERACIONES”.  Se evalúa la respuesta del servicio “TIGOCRMOPERACIONES”, en caso de error se recupera el mensaje del servicio legacy. |  |
| callGetCardData | Se genera el Request getCardDataRequest con xquery TransformToCRMgetCardDataRq.xq para invocar el servicio TIGOCRMOPERACIONES. Se envía el valor **profileId** del archivo de configuración.  Se invoca la operación “getCardData” del servicio “TIGOCRMOPERACIONES”.  Se evalúa la respuesta del servicio “TIGOCRMOPERACIONES”, en caso de error se recupera el mensaje del servicio legacy. |  |
| responseGetCardStatus | Se genera la respuesta exitosa del servicio con Xquery TransformToGetCardStatusRs.xq.  Se reemplaza la respuesta del servicio en la variable body. |  |
| replaceCodeError | Se recupera el código de error.  fn:data($fault/ctx:errorCode) |  |
| callMapResponse | Se realiza la asignación del mensaje de Request para el servicio MapResponse.  Se invoca el servicio MapResponse para obtener la descripción del código de error arrojado en RequestPipeLine.  Se realiza una validación para saber si el servicio devolvió la descripción del código de error, en caso contrario se asigna por default el código DES-999 a la variable de respuesta del servicio. |  |
| assignFaultResponse | Se asigna la variable de respuesta faultResponse (del tipo ResponseHeader) usando el xquery TransformMapResponse ubicado en la carpeta Common/XQUERY, pasando como parámetro de entrada la respuesta del servicio MapResponse |  |
| invokeErrorProducer | Se realiza la asignación del mensaje de Request para el servicio ErrorProducerOSB Se utiliza el xquery TransformToErrorRequest ubicado en la carpeta Common/XQUERY.  Se invoca el servicio ErrorProducerOSB para registrar el error ocurrido a nivel de RequestPipeline, haciendo replace del mensaje Request en la variable body. |  |
| responsePipeLine | Se realiza un replace con la variable faultResponse como contenido del nodo de respuesta del servicio (variable body del servicio).  Se responde con bandera en *Success*. |  |
| assignFaultResponse | Se asigna la variable “faultResponse” con xquery TransformFaultToResponseHeader ubicado en la carpeta Common/XQUERY. |  |
| invokeErrorProducer | Se realiza la asignación del mensaje de Request para el servicio ErrorProducerOSB Se utiliza el xquery TransformToErrorRequest ubicado en la carpeta Common/XQUERY.  Se invoca el servicio ErrorProducerOSB para registrar el error ocurrido a nivel de RequestPipeline, haciendo replace del mensaje Request en la variable body. |  |
| responseErrorServiceLocal | Se realiza un replace con la variable faultResponse como contenido del nodo de respuesta del servicio (variable body del servicio).  Se responde con bandera en *Failure*. |  |

## Descripción de Etapas Operación GetBenefits

A continuación se describen las etapas que conforman las actividades del flujo GetBenefitsPipeline.

|  |  |  |
| --- | --- | --- |
| **Actividad** | **Descripción** | **Diagrama** |
| validationStage | Se realiza la validación del elemento de entrada body*, estos deben corresponder a la definición en el XSD GetLoyaltyInfoRequest.xsd*  Se asigna el Request Body del servicio a la variable “serviceRequest”. |  |
| callSetCardLogin | Se lee el Archivo de Configuración.  Se genera el Request setCardLoginRequest con xquery TransformToCRMsetCardLoginRq.xq para invocar el servicio TIGOCRMOPERACIONES.  Se invoca la operación “setCardLogin” del servicio “TIGOCRMOPERACIONES”.  Se evalúa la respuesta del servicio “TIGOCRMOPERACIONES”, en caso de error se recupera el mensaje del servicio legacy. |  |
| callGetCardData | Se genera el Request getCardDataRequest con xquery TransformToCRMgetCardDataRq.xq para invocar el servicio TIGOCRMOPERACIONES. Se envía el valor **benefitsId** del archivo de configuración.  Se invoca la operación “getCardData” del servicio “TIGOCRMOPERACIONES”.  Se evalúa la respuesta del servicio “TIGOCRMOPERACIONES”, en caso de error se recupera el mensaje del servicio legacy. |  |
| responseGetBenefits | Se genera la respuesta exitosa del servicio con Xquery TransformToGetBenefitsRs.xq.  Se reemplaza la respuesta del servicio en la variable body. |  |
| replaceCodeError | Se recupera el código de error.  fn:data($fault/ctx:errorCode) |  |
| callMapResponse | Se realiza la asignación del mensaje de Request para el servicio MapResponse.  Se invoca el servicio MapResponse para obtener la descripción del código de error arrojado en RequestPipeLine.  Se realiza una validación para saber si el servicio devolvió la descripción del código de error, en caso contrario se asigna por default el código DES-999 a la variable de respuesta del servicio. |  |
| assignFaultResponse | Se asigna la variable de respuesta faultResponse (del tipo ResponseHeader) usando el xquery TransformMapResponse ubicado en la carpeta Common/XQUERY, pasando como parámetro de entrada la respuesta del servicio MapResponse |  |
| invokeErrorProducer | Se realiza la asignación del mensaje de Request para el servicio ErrorProducerOSB Se utiliza el xquery TransformToErrorRequest ubicado en la carpeta Common/XQUERY.  Se invoca el servicio ErrorProducerOSB para registrar el error ocurrido a nivel de RequestPipeline, haciendo replace del mensaje Request en la variable body. |  |
| responsePipeLine | Se realiza un replace con la variable faultResponse como contenido del nodo de respuesta del servicio (variable body del servicio).  Se responde con bandera en *Success*. |  |
| assignFaultResponse | Se asigna la variable “faultResponse” con xquery TransformFaultToResponseHeader ubicado en la carpeta Common/XQUERY. |  |
| invokeErrorProducer | Se realiza la asignación del mensaje de Request para el servicio ErrorProducerOSB Se utiliza el xquery TransformToErrorRequest ubicado en la carpeta Common/XQUERY.  Se invoca el servicio ErrorProducerOSB para registrar el error ocurrido a nivel de RequestPipeline, haciendo replace del mensaje Request en la variable body. |  |
| responseErrorServiceLocal | Se realiza un replace con la variable faultResponse como contenido del nodo de respuesta del servicio (variable body del servicio).  Se responde con bandera en *Failure*. |  |

## Descripción de Etapas Operación GetCashbackInfo

A continuación se describen las etapas que conforman las actividades del flujo GetCashbackInfoPipeline.

|  |  |  |
| --- | --- | --- |
| **Actividad** | **Descripción** | **Diagrama** |
| validationStage | Se realiza la validación del elemento de entrada body*, estos deben corresponder a la definición en el XSD GetLoyaltyInfoRequest.xsd*  Se asigna el Request Body del servicio a la variable “serviceRequest”. |  |
| callGetCardBalance | Se lee el Archivo de Configuración.  Se genera el Request getCardBalanceRequest con xquery TransformToCRMgetCardBalanceRq.xq para invocar el servicio TIGOCRMOPERACIONES.  Se invoca la operación “getCardBalance” del servicio “TIGOCRMOPERACIONES”.  Se evalúa la respuesta del servicio “TIGOCRMOPERACIONES”, en caso de error se recupera el mensaje del servicio legacy. |  |
| responseGetCashbackInfo | Se genera la respuesta exitosa del servicio con Xquery TransformToGetCashbackInfoRs.xq.  Se reemplaza la respuesta del servicio en la variable body. |  |
| invokeAuditProducer | Se realiza la asignación del mensaje de Request para el servicio AuditProducerOSB Se utiliza el xquery TransformToAuditRequest ubicado en la carpeta Common/XQUERY.  Se invoca el servicio AuditProducerOSB para registrar el log del servicio, en caso que ocurra un error en el llamado a este servicio se manejara la excepción continuando con el flujo normal. |  |
| replaceCodeError | Se recupera el código de error.  fn:data($fault/ctx:errorCode) |  |
| callMapResponse | Se realiza la asignación del mensaje de Request para el servicio MapResponse.  Se invoca el servicio MapResponse para obtener la descripción del código de error arrojado en RequestPipeLine.  Se realiza una validación para saber si el servicio devolvió la descripción del código de error, en caso contrario se asigna por default el código DES-999 a la variable de respuesta del servicio. |  |
| assignFaultResponse | Se asigna la variable de respuesta faultResponse (del tipo ResponseHeader) usando el xquery TransformMapResponse ubicado en la carpeta Common/XQUERY, pasando como parámetro de entrada la respuesta del servicio MapResponse |  |
| invokeErrorProducer | Se realiza la asignación del mensaje de Request para el servicio ErrorProducerOSB Se utiliza el xquery TransformToErrorRequest ubicado en la carpeta Common/XQUERY.  Se invoca el servicio ErrorProducerOSB para registrar el error ocurrido a nivel de RequestPipeline, haciendo replace del mensaje Request en la variable body. |  |
| responsePipeLine | Se realiza un replace con la variable faultResponse como contenido del nodo de respuesta del servicio (variable body del servicio).  Se responde con bandera en *Success*. |  |
| assignFaultResponse | Se asigna la variable “faultResponse” con xquery TransformFaultToResponseHeader ubicado en la carpeta Common/XQUERY. |  |
| invokeErrorProducer | Se realiza la asignación del mensaje de Request para el servicio ErrorProducerOSB Se utiliza el xquery TransformToErrorRequest ubicado en la carpeta Common/XQUERY.  Se invoca el servicio ErrorProducerOSB para registrar el error ocurrido a nivel de RequestPipeline, haciendo replace del mensaje Request en la variable body. |  |
| responseErrorServiceLocal | Se realiza un replace con la variable faultResponse como contenido del nodo de respuesta del servicio (variable body del servicio).  Se responde con bandera en *Failure*. |  |

## Descripción de Etapas Operación GetCashbackDetails

A continuación se describen las etapas que conforman las actividades del flujo GetCashbackDetailsPipeline.

|  |  |  |
| --- | --- | --- |
| **Actividad** | **Descripción** | **Diagrama** |
| validationStage | Se realiza la validación del elemento de entrada body*, estos deben corresponder a la definición en el XSD GetLoyaltyInfoRequest.xsd*  Se asigna el Request Body del servicio a la variable “serviceRequest”. |  |
| callGetCardBalanceSheet | Se lee el Archivo de Configuración.  Se genera el Request getCardBalanceSheetRequest con xquery TransformToCRMgetCardBalanceSheetRq.xq para invocar el servicio TIGOCRMOPERACIONES.  Se invoca la operación “getCardBalanceSheet” del servicio “TIGOCRMOPERACIONES”.  Se evalúa la respuesta del servicio “TIGOCRMOPERACIONES”, en caso de error se recupera el mensaje del servicio legacy. |  |
| responseGetCashbackDetails | Se genera la respuesta exitosa del servicio con Xquery TransformToGetCashbackInfoRs.xq.  Se reemplaza la respuesta del servicio en la variable body. |  |
| invokeAuditProducer | Se realiza la asignación del mensaje de Request para el servicio AuditProducerOSB Se utiliza el xquery TransformToAuditRequest ubicado en la carpeta Common/XQUERY.  Se invoca el servicio AuditProducerOSB para registrar el log del servicio, en caso que ocurra un error en el llamado a este servicio se manejara la excepción continuando con el flujo normal. |  |
| replaceCodeError | Se recupera el código de error.  fn:data($fault/ctx:errorCode) |  |
| callMapResponse | Se realiza la asignación del mensaje de Request para el servicio MapResponse.  Se invoca el servicio MapResponse para obtener la descripción del código de error arrojado en RequestPipeLine.  Se realiza una validación para saber si el servicio devolvió la descripción del código de error, en caso contrario se asigna por default el código DES-999 a la variable de respuesta del servicio. |  |
| assignFaultResponse | Se asigna la variable de respuesta faultResponse (del tipo ResponseHeader) usando el xquery TransformMapResponse ubicado en la carpeta Common/XQUERY, pasando como parámetro de entrada la respuesta del servicio MapResponse |  |
| invokeErrorProducer | Se realiza la asignación del mensaje de Request para el servicio ErrorProducerOSB Se utiliza el xquery TransformToErrorRequest ubicado en la carpeta Common/XQUERY.  Se invoca el servicio ErrorProducerOSB para registrar el error ocurrido a nivel de RequestPipeline, haciendo replace del mensaje Request en la variable body. |  |
| responsePipeLine | Se realiza un replace con la variable faultResponse como contenido del nodo de respuesta del servicio (variable body del servicio).  Se responde con bandera en *Success*. |  |
| assignFaultResponse | Se asigna la variable “faultResponse” con xquery TransformFaultToResponseHeader ubicado en la carpeta Common/XQUERY. |  |
| invokeErrorProducer | Se realiza la asignación del mensaje de Request para el servicio ErrorProducerOSB Se utiliza el xquery TransformToErrorRequest ubicado en la carpeta Common/XQUERY.  Se invoca el servicio ErrorProducerOSB para registrar el error ocurrido a nivel de RequestPipeline, haciendo replace del mensaje Request en la variable body. |  |
| responseErrorServiceLocal | Se realiza un replace con la variable faultResponse como contenido del nodo de respuesta del servicio (variable body del servicio).  Se responde con bandera en *Failure*. |  |

# Recursos del Servicio

## Esquemas XSD

Los esquemas que son utilizados directamente para la correcta ejecución del servicio son:

|  |  |  |
| --- | --- | --- |
| **Esquema** | **Descripción** | **Ubicación** |
| GetLoyaltyInfoRequest.xsd | Esquema donde se describe el elemento GetLoyaltyInfoRequestType el cual es el mensaje request para el servicio GetLoyaltyInfoLocal y GetLoyaltyInfo | /TigoApp/GetLoyaltyInfo/XSD/GetLoyaltyInfoRequest.xsd |
| GetLoyaltyInfoResponse.xsd | Esquema donde se describe el elemento GetLoyaltyInfoResponseType el cual es el mensaje response para el servicio GetLoyaltyInfoLocal y GetLoyaltyInfo | /TigoApp/GetLoyaltyInfo/XSD/GetLoyaltyInfoResponse.xsd |
| AuditRequest.xsd | Esquema donde se describe el mensaje AuditAdapterInput, el cual es el mensaje request del servicio AuditProducerOSB | /TigoApp/Common/XSD/AuditRequest.xsd |
| ErrorRequest.xsd | Esquema donde se describe el mensaje ErrorAdapterInput, el cual es el mensaje request del servicio ErrorProducerOSB | /TigoApp/Common/XSD/ErrorRequest.xsd |
| MapResponse.xsd | Esquema donde se describe los mensajes del servicio ErrorProducerOSB. | /TigoApp/Common/XSD/MapResponse.xsd |
| ParameterTypeV2.xsd | Contiene la definición de los campos adicionales para los mensajes de entrada y salida del servicio | /TigoApp/Common/XSD/ParameterTypeV2.xsd |
| RequestHeaderV3.xsd | Esquema donde se definen los campos de entrada para el header. | /TigoApp/Common/XSD/RequestHeaderV3.xsd |
| ResponseHeaderV3.xsd | Esquema donde se definen los campos de salida para el header. | /TigoApp/Common/XSD/ResponseHeaderV3.xsd |

## WSDL

|  |  |  |
| --- | --- | --- |
| **Nombre** | **Descripción** | **Ubicación** |
| PS\_GetLoyaltyInfoLocal.wsdl | Archivo WSDL que define la invocación de la operación GetLoyaltyInfo del proxy GetLoyaltyInfoLocal\_PUB.proxy. | /TigoApp/GetLoyaltyInfo/WSDL/PS\_GetLoyaltyInfoLocal.wsdl |
| PS\_GetLoyaltyInfo.wsdl | Archivo WSDL que define la invocación de la operación GetLoyaltyInfo del proxy GetLoyaltyInfo\_PUB.proxy. | /TigoApp/GetLoyaltyInfo/WSDL/PS\_GetLoyaltyInfo.wsdl |
| BS\_TIGOCRMOPERACIONES.wsdl | Archivo WSDL que define la invocación de la operación setCardAffiliation del business service TIGOCRMOPERACIONES.biz. | /TigoApp/GetLoyaltyInfo/WSDL/BS\_TIGOCRMOPERACIONES.wsdl |
| BS\_MapResponse.wsdl | Archivo WSDL que define la invocación de la operación process de MapResponse, y da origen al Business Service MapResponse.biz | /TigoApp/Common/WSDL/BS\_MapResponse.wsdl |
| BS\_AuditProducerOSB.wsdl | Archivo WSDL que define la invocación de la operación execute de AuditProducerOSB, y da origen al Business Service AuditProducerOSB.biz | /TigoApp/Common/WSDL/BS\_AuditProducerOSB.wsdl |
| BS\_ErrorProducerOSB.wsdl | Archivo WSDL que define la invocación de la operación execute de ErrorProducerOSB, y da origen al Business Service ErrorProducerOSB.biz | /TigoApp/Common/WSDL/BS\_ErrorProducerOSB.wsdl |

## XQUERY

|  |  |  |
| --- | --- | --- |
| **Xquery** | **Descripción** | **Ubicación** |
| TransformToCRMgetCardBalanceRq.xq | Genera el elemento getCardBalance de Request | /TigoApp/GetLoyaltyInfo/XQUERY/TransformToCRMgetCardBalanceRq.xq |
| TransformToCRMgetCardBalanceSheetRq.xq | Genera el elemento getCardBalanceSheet de Request | /TigoApp/GetLoyaltyInfo/XQUERY/TransformToCRMgetCardBalanceSheetRq.xq |
| TransformToCRMgetCardDataRq.xq | Genera el elemento getCardData de Request | /TigoApp/GetLoyaltyInfo/XQUERY/TransformToCRMgetCardDataRq.xq |
| TransformToCRMsetCardLoginRq.xq | Genera el elemento setCardLogin de Request | /TigoApp/GetLoyaltyInfo/XQUERY/TransformToCRMsetCardLoginRq.xq |
| TransformToGetBenefitsRs.xq | Genera el elemento getBenefitsResponse de Response | /TigoApp/GetLoyaltyInfo/XQUERY/TransformToGetBenefitsRs.xq |
| TransformToGetCashbackInfoRs.xq | Genera el elemento getCashbackInfoResponse de Response | /TigoApp/GetLoyaltyInfo/XQUERY/TransformToGetCashbackInfoRs.xq |
| TransformToGetCardStatusRs.xq | Genera el elemento getCardStatusResponse de Response | /TigoApp/GetLoyaltyInfo/XQUERY/TransformToGetCardStatusRs.xq |
| TransformToGetCashbackDetailsRs.xq | Genera el elemento getCashbackDetailsResponse de Response | /TigoApp/GetLoyaltyInfo/XQUERY/TransformToGetCashbackDetailsRs.xq |
| ConfigurationFile.xq | Archivo xml de configuración para valores fijos utilizados por el servicio en tiempo de ejecución. | /TigoApp/GetLoyaltyInfo/XQUERY/ConfigurationFile.xq |
| TransformFaultToResponseHeader.xq | Genera el elemento de respuesta ResponseHeader cuando ocurre un error. Teniendo como entrada la variable “fault” | /TigoApp/Common/XQUERY/TransformFaultToResponseHeader.xq |
| TransformMapResponse.xq | Genera el elemento de respuesta ResponseHeader cuando ocurre un error. Teniendo como entrada la respuesta del servicio MapResponse | /TigoApp/Common/XQUERY/TransformMapResponse.xq |
| TransformOKToResponseHeader.xq | Genera el elemento de respuesta ResponseHeader cuando el servicio con respuesta en header se ejecuto exitosamente. | /TigoApp/Common/XQUERY/TransformOKToResponseHeader.xq |
| TransformToErrorRequest.xq | Genera el elemento Request ErrorAdapterInputpara la invocación del servicio ErrorProducerOSB | /TigoApp/Common/XQUERY/TransformToErrorRequest.xq |

## Business Service

|  |  |  |
| --- | --- | --- |
| **Business Service** | **Descripción** | **Ubicación** |
| GetLoyaltyInfoLocal.biz | Business Service correspondiente al servicio GetLoyaltyInfoLocal | /TigoApp/GetLoyaltyInfo/BS/GetLoyaltyInfoLocal.biz |
| TIGOCRMOPERACIONES.biz | Business Service a través del cual se invoca el servicio TIGOCRMOPERACIONES | /TigoApp/GetLoyaltyInfo/BS/TIGOCRMOPERACIONES.biz |
| AuditProducerOSB.biz | Business Service a través del cual se invoca el servicio AuditProducerOSB/proxy/AuditProducer | /TigoApp/Common/BS/AuditProducerOSB.biz |
| ErrorProducerOSB.biz | Business Service a través del cual se invoca el servicio ErrorProducerOSB/proxy/ErrorProducer | /TigoApp/Common/BS/ErrorProducerOSB.biz |
| MapResponse.biz | Business Service a través del cual se invoca el servicio /soa-infra/services/ManageWorkOrders/MapResponse/mappingresponse\_client\_ep | /TigoApp/Common/BS/MapResponse.biz |

## Service Account

|  |  |  |
| --- | --- | --- |
| **Service** | **Descripción** | **Ubicación** |
| GetLoyaltyInfoLocal.sa | Recurso que persiste la configuración de credenciales de una cuenta. | /TigoApp/GetLoyaltyInfo/BS/GetLoyaltyInfoLocal.sa |

## AD

|  |  |  |
| --- | --- | --- |
| **Alerta** | **Descripción** | **Ubicación** |
| GetLoyaltyInfo.alert | Archivo alert destination, en el cual se configura como sera tratada la alerta (SLA) del servicio. | /TigoApp/GetLoyaltyInfo/AD/GetLoyaltyInfo.alert |

## Proxy Service

|  |  |  |
| --- | --- | --- |
| **Proxy Service** | **Descripción** | **Ubicación** |
| GetLoyaltyInfo\_PUB.proxy | Proxy Service a través del cual se encapsula la funcionalidad del proxy GetLoyaltyInfoLocal. Agregando Header a sus mensajes de entrada y salida | /TigoApp/GetLoyaltyInfo/PS/GetLoyaltyInfo\_PUB.proxy |
| GetLoyaltyInfoLocal\_PUB.proxy | Proxy Service a través del cual se realizan las integraciones entre los servicios legacy, validaciones, transformaciones y reglas de negocio para llevar a cabo el proceso del servicio. | /TigoApp/GetLoyaltyInfo/PS/GetLoyaltyInfoLocal\_PUB.proxy |

## Contrato de Servicio

**Parámetros de entrada Body**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Campo** | **Tipo** | **Longitud** | **Requerido** | **Descripción** |
| msisdn | String | N/A | Si | Número telefónico. Formato internacional 502+número |

**Parámetros de entrada con Header**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Campo** | **Tipo** | **Longitud** | **Requerido** | **Descripción** |
| consumerID | String | N/A | No |  |
| transactionID | String | N/A | No |  |
| country | Enum | N/A | No | SLV,GTM,HND,BOL,COL,PRY,TCD,COG,GHA,MUS,RWA,SEN,TZA |
| correlationID | String | N/A | Si |  |

**Parámetros de salida Body**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Campo** | **Tipo** | **Longitud** | **Requerido** | **Descripción** |
| prize | String | N/A | No |  |
| benefit | String | N/A | No |  |
| date | Date | N/A | No |  |
| expireDate | Date | N/A | No |  |
| name | String | N/A | No |  |
| segment | String | N/A | No |  |
| membershipDate | Date | N/A | No |  |
| contract | String | N/A | No |  |
| birthDate | Date | N/A | No |  |
| availableBalance | Decimal | N/A | No |  |
| usedBalance | Decimal | N/A | No |  |
| accumulatedBalance | Decimal | N/A | No |  |
| expiredBalance | Decimal | N/A | No |  |
| nextToExpireBalance1 | Decimal | N/A | No |  |
| expireDate1 | String | N/A | No |  |
| nextToExpireBalance2 | Decimal | N/A | No |  |
| expireDate2 | String | N/A | No |  |
| lastMonthBalance | Decimal | N/A | No |  |
| lastMonth | String | N/A | No |  |
| cutoffDate | String | N/A | No |  |
| date | Date | N/A | No |  |
| transaction | String | N/A | No |  |
| amount | Decimal | N/A | No |  |
| balance | Decimal | N/A | No |  |
| reference | String | N/A | No |  |

**Parámetros de salida con Header**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Campo** | **Tipo** | **Longitud** | **Requerido** | **Descripción** |
| correlationID | String | N/A | No |  |
| status | StatusContentType | N/A | No |  |

**Parámetros de salida en caso de error**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Campo** | **Tipo** | **Longitud** | **Requerido** | **Descricpión** |
| status | StatusContentType | N/A | No |  |
| code | string | N/A | No |  |
| codeType | string | N/A | No |  |
| description | string | N/A | No |  |

**Detalle de Invocación**

|  |  |
| --- | --- |
| **Campo** | **Detalle** |
| **Protocolo de Transporte** | HTTP/SOAP |
| **Modelado de Invocación** | Request/Response |
| **URI de Wsdl** | [http://172.30.9.9:8011/TigoApp/GetLoyaltyInfo/V1?wsdl](http://172.30.9.9:8011/TigoApp/GetDeezerAvailability/V1?wsdl) |
| **URI de Wsdl Local** | [http://172.30.9.9:8011/TigoApp/GetLoyaltyInfoLocal/V1?wsdl](http://172.30.9.9:8011/TigoApp/GetDeezerAvailabilityLocal/V1?wsdl) |
| **Operación** | GetLoyaltyInfo |