



# Primary Risk API

V1.0

 Primary

## Historia del documento

Fecha	Versión	Descripción	Autor
22/02/2016	1.0	Versión inicial	Primary

## Índice

Introducción	3
Consulta de posiciones para una cuenta	4
Consulta de reporte de cuenta	6
Consulta de detalle de posición	8
Consulta de importaciones detalladas	12
Importación de Garantías	14
Importación simple de Garantías	17
Importación de Saldos	18
Importación simple de Saldos	21
Importación de Posiciones Iniciales	22
Importación de Crédito	25
Anexo - Integraciones en el ámbito de MatrizOMS	<b>27</b>

## Introducción

Primary Risk API es la interfaz que permite comunicarse con el sistema RIMA (Risk Manager). Soporta métodos para consultar datos de riesgo asociados a las cuentas comitentes, y

permite además importar datos de posiciones, saldos y garantías al sistema. Dichos datos deben estar en formato JSON.

La API actualmente soporta métodos para:

- Consulta de posiciones para una cuenta
- Consulta de posiciones abiertas para una cuenta
- Consulta de reporte de cuenta
- Consulta de reportes de cuentas
- Consulta de detalle de posición
- Consulta de integridad de datos
- Consulta de importaciones realizadas
- Importación de garantías\*
- Importación de saldos\*
- Importación de posiciones iniciales\*

\*Cada uno con una variante: envío **individual** o **múltiple** de los datos.

## Consulta de posiciones para una cuenta

*HTTP Request*

GET /rest/position/getPositions/{accountName}

- [{accountName}](#) = cuenta a consultar.

### Autenticación

HTTP Basic Authentication

### Authorization

- *RIMA\_ADMINISTRATOR*: Puede consultar las posiciones de cualquier cuenta del sistema.
- *RIMA\_BROKER*: Puede consultar las posiciones de las cuentas configuradas para el rol.
- *PRE\_TRADE\_RISK*: Puede consultar las posiciones de las cuentas configuradas para el rol.

### Parámetros

<b>Produce</b>	application/json
<b>Consumes</b>	application/json
<b>Response Object (template)</b>	<pre> array[   Object {     symbol(string),     instrument(Instrument),     originalBuyPrice(number),     originalSellPrice(decimal),     buyPrice(number),     buySize(number),     sellPrice(number),     sellSize(number),     totalDailyDiff(number),     totalDiff(number)   } ] Instrument {   symbolReference(string),   settlType(string) } </pre>

### Ejemplos

<b>Http Request</b>	/position/getPositions/10
---------------------	---------------------------

Objeto enviado	<pre>[   {     "symbol": "DLR/DIC22",     "instrument": {       "symbolReference": "DLR122022"     },     "originalBuyPrice": 4.775000000000,     "originalSellPrice": 6.987000000000,     "buyPrice": 9.550000000000,     "buySize": 1000.000000,     "sellPrice": 6.990000000000,     "sellSize": 10000000.000000,     "totalDailyDiff": -25600000000.00,     "totalDiff": -25625225000.00   },   {     "symbol": "DONov15",     "instrument": {       "symbolReference": "DLR122022"     },     "originalBuyPrice": 6.959000000000,     "originalSellPrice": 0.000000,     "buyPrice": 6.960000000000,     "buySize": 10000000.000000,     "sellPrice": 0.000000,     "sellSize": 0,     "totalDailyDiff": 35700000000.00,     "totalDiff": 35710000000.00   },   {     "symbol": "MERV - XMEV - TS - CI",     "instrument": {       "settlType": "CASH",       "symbolReference": "TS"     },     "originalBuyPrice": 12.000000000000,     "originalSellPrice": 0.000000,     "buyPrice": 13.000000000000,     "buySize": 10.000000,     "sellPrice": 0.000000,     "sellSize": 0,     "totalDailyDiff": 1560000.00,     "totalDiff": 1570000.00   },   {     "symbol": "MERV - XMEV - TS - 72hs",     "instrument": {       "settlType": "T_PLUS_3",       "symbolReference": "TS"     },     "originalBuyPrice": 11.000000000000,     "originalSellPrice": 0.000000,     "buyPrice": 10.000000000000,     "buySize": 1.000000,     "sellPrice": 0.000000,     "sellSize": 0,     "totalDailyDiff": 159000.00,     "totalDiff": 158000.00   } ]</pre>
----------------	--

## Consulta de reporte de cuenta

*HTTP Request*

GET /rest/accountReport/{accountName}

- {accountName} = cuenta a consultar.

### Autenticación

HTTP Basic Authentication

### Authorization

- *RIMA\_ADMINISTRATOR*: puede consultar el reporte de cuenta para cualquier cuenta del sistema.
- *RIMA\_BROKER*: puede consultar el reporte de cuenta para las cuentas configuradas para el permiso.
- *PRE\_TRADE\_RISK*: puede consultar el reporte de cuenta de las cuentas configuradas para el permiso.

### Parámetros

<b>Produce</b>	application/json
<b>Consumes</b>	application/json
<b>Response Object (template)</b>	<pre> <b>object</b> {   accountName (<b>string</b>),   availableToCollateral (<b>number</b>),   collateral (<b>number</b>),   credit (<b>number</b>),   dailyDiff (<b>number</b>),   margin (<b>number</b>),   lastCalculation (<b>number</b>),   hasError (<b>boolean</b>),   <b>object</b>[     settleType (<b>string</b>) :     detailedAccountReports (<b>DetailedAccountReport</b>)   ] }  <b>DetailedAccountReport</b> {   cash (<b>number</b>),   movement (<b>number</b>),   portfolio (<b>number</b>),   accountValue (<b>number</b>),   availableToOperate (<b>number</b>) }  Valores permitidos settleType ("CASH", "NEXT_DAY", "T_PLUS_2", "T_PLUS_3") </pre>

### Ejemplos

<b>Http Request</b>	/accountReport/10
<b>Response Object (ejemplo)</b>	<pre> {   "accountName": "10",   "availableToCollateral": 10000.000000000000,   "collateral": 0,   "credit": 10000.000000000000,   "dailyDiff": 0,   "margin": 0,   "lastCalculation": 1433169526058,   "hasError": false,   "detailedAccountReports": {     "CASH": {       "cash": 0,       "movement": 0,       "portfolio": 0,       "accountValue": 10000.000000000000,       "availableToOperate": 10000.000000000000     },     "NEXT_DAY": {       "cash": 0,       "movement": 0,       "portfolio": 0,       "accountValue": 10000.000000000000,       "availableToOperate": 10000.000000000000     },     "T_PLUS_2": {       "cash": 0,       "movement": 0,       "portfolio": 0,       "accountValue": 10000.000000000000,       "availableToOperate": 10000.000000000000     },     "T_PLUS_3": {       "cash": 0,       "movement": 0,       "portfolio": 0,       "accountValue": 10000.000000000000,       "availableToOperate": 10000.000000000000     }   } } </pre>

## Consulta de detalle de posición

HTTP Request

GET /rest/detailedPosition/{accountName}



- *{accountName}* = cuenta a consultar.

### Autenticación

HTTP Basic Authentication

### Authorization

- *RIMA\_ADMINISTRATOR*: puede consultar el detalle de la posición de cualquier cuenta del sistema.
- *RIMA\_BROKER*: puede consultar el detalle de la posición de las cuentas configuradas para el permiso.
- *PRE\_TRADE\_RISK*: puede consultar el detalle de las cuentas configuradas para el permiso.

### Parámetros

<b>Produce</b>	application/json
<b>Consumes</b>	application/json
<b>Response Object (template)</b>	<pre> Object {   lastCalculation(number),   account(string),   totalDailyDiffPlain(number),   totalMarketValue(number),   report(     object(       ContractType:detailedPositions(DetailedPositions)     )   ) }  DetailedPositions {   object(string:Array[DetailedPositionItem]),   instrumentMarketValue(number),   instrumentInitialSize(number),   instrumentFilledSize(number),   instrumentCurrentSize(number) }  DetailedPositionItem {   currency(string)   symbolReference(string)   exchangeRate(number)   contractType(string),   contractSize(number),   priceConversionFactor(number),   totalInitialSize(number),   totalFilledSize(number),   buyInitialPrice(number),   buyInitialSize(number),   buyFilledPrice(number),   buyFilledSize(number), </pre>

	<pre> sellInitialPrice (number) , sellInitialSize (number) , sellFilledPrice (number) , sellFilledSize (number) , marketPrice (number) , buyCurrentSize (number) , sellCurrentSize (number) , totalCurrentSize (number) , detailedDailyDiff (DetailedDailyDiff) }  DetailedDailyDiff { totalDailyDiff (number) , buyDailyDiffPlain (number) , sellDailyDiffPlain (number) , buyPricePPPDiff (number) , sellPricePPPDiff (number) , buyDailyDiff (number) , sellDailyDiff (number) , totalDailyDiffPlain (number) , }  Valores permitidos currency -&gt; <ul style="list-style-type: none"> <li>● "USD G": Dólar Matba Rofex</li> <li>● "USD R": Dólar Ref. CA3500</li> <li>● "USD D": Dólar MEP</li> <li>● "USD C": Dólar Cable</li> <li>● "U\$S": Dólar</li> <li>● "ARS": Pesos</li> </ul> contractType -&gt; STOCK, BOND, FUTURE, OPTION_PUT, OPTION_CALL, CEDEAR) </pre>
--	---

### Ejemplos

<b>Http Request</b>	/detailedPosition/10
<b>Response Object (ejemplo)</b>	<pre> {   "lastCalculation":1430320921800,   "account":"10",   "totalDailyDiffPlain":60500.0000,   "totalMarketValue":-72.0000,   "report":{     "FUTURE":{       "DLR112015":{         "detailedPositions":[           {             "currency":"ARS",             "symbolReference":"DLR122022",             "exchangeRate":1.000000,             "contractType":"FUTURE",             "contractSize":1000.000000,             "priceConversionFactor":1.000000,             "totalInitialSize":10.000000,             "totalFilledSize":0,             "buyInitialPrice":6.960000,             "buyInitialSize":10.000000,             "buyFilledPrice":0,             "buyFilledSize":0,             "sellInitialPrice":0,             "sellInitialSize":0,           }         ]       }     }   } } </pre>

```

        "sellFilledPrice":0,
        "sellFilledSize":0,
        "marketPrice":10.000000000000,
        "buyCurrentSize":10.000000,
        "sellCurrentSize":0,
        "detailedDailyDiff":{
          "totalDailyDiff":30400.00000,
          "buyDailyDiffPlain":30400.00000,
          "sellDailyDiffPlain":0,
          "buyPricePPPDiff":6.960000,
          "sellPricePPPDiff":0,
          "buyDailyDiff":30400.00000,
          "sellDailyDiff":0,
          "totalDailyDiffPlain":30400.0000
        },
        "totalCurrentSize":10.000000
      }
    ],
    "instrumentInitialSize":10.000000,
    "instrumentFilledSize":0,
    "instrumentCurrentSize":10.000000
  },
  "DLR122012":{
    "detailedPositions":[
      {
        "currency":"ARS",
        "symbolReference":"DLR122015",
        "exchangeRate":1.000000,
        "contractType":"FUTURE",
        "contractSize":1000.000000,
        "priceConversionFactor":1.000000,
        "totalInitialSize":-10.000000,
        "totalFilledSize":0,
        "buyInitialPrice":0,
        "buyInitialSize":0,
        "buyFilledPrice":0,
        "buyFilledSize":0,
        "sellInitialPrice":6.990000,
        "sellInitialSize":10.000000,
        "sellFilledPrice":0,
        "sellFilledSize":0,
        "marketPrice":10.000000000000,
        "buyCurrentSize":0,
        "sellCurrentSize":10.000000,
        "detailedDailyDiff":{
          "totalDailyDiff":30100.00000,
          "buyDailyDiffPlain":0,
          "sellDailyDiffPlain":30100.0000,
          "buyPricePPPDiff":0,
          "sellPricePPPDiff":6.990000,
          "buyDailyDiff":0,
          "sellDailyDiff":30100.00000,
          "totalDailyDiffPlain":30100.0000
        },
        "totalCurrentSize":-10.000000
      }
    ],
    "instrumentInitialSize":-10.000000,
    "instrumentFilledSize":0,
    "instrumentCurrentSize":-10.000000
  }
},
"STOCK":{
  "TS":{
    "detailedPositions":[
      {
        "currency":"ARS",
        "settlType":"T_PLUS_3",
        "symbolReference":"TS",

```

```

"exchangeRate":1.000000,
"contractType":"STOCK",
"contractSize":1000.000000,
"priceConversionFactor":1.000000,
"totalInitialSize":1.000000,
"totalFilledSize":0,
"buyInitialPrice":10.000000,
"buyInitialSize":1.000000,
"buyFilledPrice":0,
"buyFilledSize":0,
"sellInitialPrice":0,
"sellInitialSize":0,
"sellFilledPrice":0,
"sellFilledSize":0,
"marketPrice":9.0000000000,
"buyCurrentSize":1.000000,
"sellCurrentSize":0,
"totalCurrentSize":1.000000,
"marketValue":9.0000
},
{
"currency":"ARS",
"settlType":"CASH",
"symbolReference":"TS",
"exchangeRate":1.000000,
"contractType":"STOCK",
"contractSize":1000.000000,
"priceConversionFactor":1.000000,
"totalInitialSize":-10.000000,
"totalFilledSize":1.000000,
"buyInitialPrice":0,
"buyInitialSize":0,
"buyFilledPrice":9.000000,
"buyFilledSize":1.000000,
"sellInitialPrice":10.000000,
"sellInitialSize":10.000000,
"sellFilledPrice":0,
"sellFilledSize":0,
"marketPrice":9.0000000000,
"buyCurrentSize":1.000000,
"sellCurrentSize":10.000000,
"totalCurrentSize":-9.000000,
"marketValue":-81.0000
}
],
"instrumentMarketValue":-72.0000,
"instrumentInitialSize":-9.000000,
"instrumentFilledSize":1.000000,
"instrumentCurrentSize":-8.000000
}
}
}
}

```

## Consulta de importaciones detalladas

HTTP Request

GET /rest/importLogReport/getAll?userName={value1}&importType={value2}

- {value1} = nombre de usuario a consultar. No es obligatorio.
- {value2} = tipo de importación a consultar. No es obligatorio.

Valores permitidos para importType:

- COLLATERALS
- COLLATERALS\_MULTIPLE
- IMPORT\_CREDITS
- IMPORT\_INITIAL\_POSITIONS
- IMPORT\_INITIAL\_POSITIONS\_MULTIPLE
- IMPORT\_MOVEMENTS
- IMPORT\_MOVEMENTS\_MULTIPLE
- SETTLEMENTS
- SETTLEMENT
- MULTIPLE\_MARGINS
- MARGINS
- EXCHANGE\_RATES
- INSTRUMENT
- INSTRUMENTS

Autenticación

HTTP Basic Authentication

Authorization

- *RIMA\_ADMINISTRATOR*: puede realizar consultas de importaciones que hizo cualquier usuario.
- *RIMA\_BROKER*: puede realizar la consulta de las importaciones realizadas por el broker autenticado.

Parámetros

<b>Produce</b>	application/json
<b>Consumes</b>	application/json
<b>Response Object (template)</b>	<pre>Object {   importLogReports (array [ImportLogData]) } ImportLogData {   username (string),   datetime (number),   successfullyCompleted (boolean),   importType (string),   resultOutput (string),   input (string)</pre>

	<pre>}  Valores permitidos para importType:  • COLLATERALS • COLLATERALS_MULTI • IMPORT_CREDITS • IMPORT_INITIAL_POSITIONS • IMPORT_INITIAL_POSITIONS_MULTIPLE • IMPORT_MOVEMENTS • IMPORT_MOVEMENTS_MULTIPLE • SETTLEMENTS • SETTLEMENT • MULTIPLE MARGINS • MARGINS • EXCHANGE PLERATES • INSTRUMENT • INSTRUMENTS</pre>
--	--

*Ejemplos*

<b>Http Request</b>	/importLogReport/getAll?userName=user1&importType=COLLATERALS
<b>Response Object (ejemplo)</b>	<pre>{   "importLogReports": [     {       "input": {"account": "", "collaterals": [{"currency": "ARS", "account": "11000", "amount": 3000}, {"currency": "USD R", "account": "11000", "amount": 3000}, {"currency": "USD C", "account": "11000", "amount": 3000}]},       "username": "user1",       "datetime": 1447935883000,       "importType": "COLLATERALS",       "resultOutput": {"status": "OK"},       "successfullyCompleted": true     },     {       "input": {"margins": [{"symbolReference": "DLR012016", "marginBid": 865.0, "marginOffer": 865.0, "settlementType": null}, {"symbolReference": "DLR022016", "marginBid": 905.0, "marginOffer": 905.0, "settlementType": null}, {"symbolReference": "DLR032016", "marginBid": 945.0, "marginOffer": 945.0, "settlementType": null}, {"symbolReference": "XVTEST", "marginBid": 300.0, "marginOffer": 300.0, "settlementType": null}]},       "username": "user1",       "datetime": 1447870480000,       "importType": "MARGINS",       "resultOutput": "Importación de márgenes realizada con éxito",       "successfullyCompleted": true     }   ] }</pre>

## Importación de Garantías

HTTP Request

POST /rest/import/collaterals

Autenticación

HTTP Basic Authentication

Parámetros

<b>Produce</b>	application/json
<b>Consumes</b>	application/json
<b>Body Object (template)</b>	<pre>Object{     account(String) ,     collaterals(array[Collateral]) } Collateral{     account(String) ,     currency(String) ,     amount(number) } </pre> <p><u>Valores permitidos:</u>  <b>currency:</b></p> <ul style="list-style-type: none"> <li>• "USD G": Dólar Matba Rofex</li> <li>• "USD R": Dólar Ref. CA3500</li> <li>• "USD D": Dólar MEP</li> <li>• "USD C": Dólar Cable</li> <li>• "U\$\$": Dólar</li> <li>• "ARS": Pesos</li> </ul>
<b>Response Object (template)</b>	<pre>Object{     status(String) ,     message(String) ,     data(array[String]) } </pre> <p><u>Valores permitidos:</u>  <b>status:</b></p> <ul style="list-style-type: none"> <li>• OK</li> <li>• ERROR</li> <li>• FAILED</li> </ul> <p><b>message:</b></p> <ul style="list-style-type: none"> <li>• DIFFERENT_ACCOUNTS</li> <li>• INVALID_ACCOUNTS</li> <li>• UNEXPECTED_ACCOUNT</li> <li>• UNKNOWN_CURENC</li> </ul>

*Ejemplos*

<b>Http Request</b>	/import/collaterals
<b>Objeto Enviado</b>	<p>1. Elimina todas las garantías cargadas para la cuenta 10 y crea las nuevas.</p> <pre> {   "account": "10",   "collaterals": [     {       "account": "10",       "currency": "ARS",       "amount": 13000     },     {       "account": "10",       "currency": "USD G",       "amount": 13000     }   ] } </pre> <p>2. Elimina todas las garantías cargadas para todas las cuentas asignadas al usuario autenticado y crea las nuevas.</p> <pre> {   "account": "",   "collaterals": [     {       "account": "10",       "currency": "ARS",       "amount": 165     },     {       "account": "50",       "currency": "USD G",       "amount": 13000     },     {       "account": "60",       "currency": "USD G",       "amount": 7000     }   ] } </pre>
<b>Objeto Recibido</b>	<p>1. Importación de garantías realizada con éxito:</p> <pre> {   "status": " OK " } </pre> <p>2. Currency no encontrado:</p> <pre> {   "status": "ERROR",   "message": "UNKNOWN_CURRENCY", } </pre>



	<pre>       "data": [         "USD T"       ]     }   </pre> <p>3. Cuentas mal ingresadas:</p> <pre>     {       "status": "ERROR",       "message": "UNEXPECTED_ACCOUNT",       "data": [         "120"       ]     }   </pre> <p>4. Cuentas no válidas para el usuario:</p> <pre>     {       "status": "ERROR",       "message": "INVALID_ACCOUNTS",       "data": [         "999",         "888",         "777"       ]     }   </pre>
--	--

## Importación simple de Garantías

Permite la importación simple de garantías. Lo que permite enviar movimientos de a un registro para minimizar la latencia en la importación.

*HTTP Request*

POST /rest/import/singleCollateral

*Autenticación*

HTTP Basic Authentication

*Parámetros*

<b>Produces</b>	application/json
<b>Consumes</b>	application/json

<b>Body Object (template)</b>	<pre>Collateral{     account(String),     currency(String), amount(number) }</pre> <p><u>Valores permitidos:</u> <u>currency:</u></p> <ul style="list-style-type: none"> <li>• "USD G": Dólar Matba Rofex</li> <li>• "USD R": Dólar Ref. CA3500</li> <li>• "USD D": Dólar MEP</li> <li>• "USD C": Dólar Cable</li> <li>• "U\$S": Dólar</li> <li>• "ARS": Pesos</li> </ul> <p>Campo Amount tiene que ser mayor a cero.</p>
<b>Response Object (template)</b>	<pre>Object{     status(String),     message(String),     data(array[String]) }</pre> <p><u>Valores permitidos:</u> <u>status:</u></p> <ul style="list-style-type: none"> <li>• OK</li> <li>• ERROR</li> <li>• FAILED</li> <li>• WARNING</li> </ul>

### Ejemplo

<b>Http Request</b>	/import/singleCollateral
<b>Objeto Enviado</b>	<p>1. Agrega una nueva garantía</p> <pre>{   "account": "10",   "currency": "ARS",   "amount": 13000 }</pre>
<b>Objeto Recibido</b>	<p>1. Importación de garantías realizada con éxito:</p> <pre>{   "status": " OK " }</pre>

## Importación de Saldos

Permite la importación de saldos.

*HTTP Request*

POST /rest/import/movements

*Autenticación*

HTTP Basic Authentication

*Parámetros*

<b>Produces</b>	application/json
<b>Consumes</b>	application/json
<b>Body Object (template)</b>	<pre> Object{     account(String) ,     keepPrevious(boolean) (default =     false) ,     movements(array[Movement]) } Movement{     account(String) ,     currency(String) ,     cash(number) ,     settlType(String) } </pre> <p><u>Valores permitidos:</u>  <b>currency:</b></p> <ul style="list-style-type: none"> <li>● "USD G": Dólar Matba Rofex</li> <li>● "USD R": Dólar Ref. CA3500</li> <li>● "USD D": Dólar MEP</li> <li>● "USD C": Dólar Cable</li> <li>● "U\$S": Dólar</li> <li>● "ARS": Pesos</li> </ul> <p><b>settlType:</b></p> <ul style="list-style-type: none"> <li>● CASH</li> <li>● NEXT_DAY</li> </ul>
<b>Response Object (template)</b>	<pre> Object{     status(String) ,     message(String) ,     data(array[String]) } </pre> <p><u>Valores permitidos:</u>  <b>status:</b></p> <ul style="list-style-type: none"> <li>● OK</li> <li>● ERROR</li> <li>● FAILED</li> </ul> <p><b>message:</b></p> <ul style="list-style-type: none"> <li>● DIFFERENT_ACCOUNTS</li> <li>● INVALID_ACCOUNTS</li> </ul>

	<ul style="list-style-type: none"> <li>• UNEXPECTED_ACCOUNT</li> <li>• UNKNOWN_CURRENC</li> </ul>
--	---

*Ejemplo*

<b>Http Request</b>	/import/movements
<b>Objeto Enviado</b>	<p>2. Elimina todos los saldos cargados para la cuenta 10 y crea los nuevos</p> <pre> {   "account": "10",   "movements": [     {       "account": "10",       "currency": "ARS",       "cash": 6000,       "settlType": "NEXT_DAY"     },     {       "account": "10",       "currency": "ARS",       "cash": 5000,       "settlType": "CASH"     }   ] } </pre> <p>3. Mantiene todos los saldos cargados para todas las cuentas asignadas al usuario autenticado y agrega los nuevos</p> <pre> {   "account": "",   "movements": [     {       "account": "10",       "currency": "ARS",       "cash": 8900,       "settlType": "NEXT_DAY"     },     {       "account": "30",       "currency": "ARS",       "cash": 98000,       "settlType": "CASH"     }   ],   "keepPrevious": true } </pre>
<b>Objeto Recibido</b>	<p>2. Importación de garantías realizada con éxito:</p> <pre> {   "status": " OK " } </pre>

## 3. Currency no encontrado:

```
{
  "status": "ERROR",
  "message": "UNKNOWN_CURRENCY",
  "data": [
    "USD T"
  ]
}
```

## 4. Cuentas mal ingresadas:

```
{
  "status": "ERROR",
  "message": "UNEXPECTED_ACCOUNT",
  "data": [
    "120"
  ]
}
```

## 5. Cuentas no válidas para el usuario:

```
{
  "status": "ERROR",
  "message": "INVALID_ACCOUNTS",
  "data": [
    "999",
    "888",
    "777"
  ]
}
```

## Importación simple de Saldos

Permite la importación simple de saldos. Lo que permite enviar movimientos de un registro para minimizar la latencia en la importación.

### HTTP Request

POST /rest/import/singleMovement

### Autenticación

HTTP Basic Authentication

### Parámetros

<b>Produce</b>	application/json
<b>Consumes</b>	application/json
<b>Body Object (template)</b>	<pre>Movement {   account(string),   currency(string),   cash(number),   settlType(integer) }</pre> <p><u>Valores permitidos:</u>  <b>currency:</b></p> <ul style="list-style-type: none"> <li>• "USD G": Dólar Matba Rofex</li> <li>• "USD R": Dólar Ref. CA3500</li> <li>• "USD D": Dólar MEP</li> <li>• "USD C": Dólar Cable</li> <li>• "U\$S": Dólar</li> <li>• "ARS": Pesos</li> </ul>
<b>Response Object (template)</b>	<pre>Object{   status(String),   message(String), }</pre> <p><u>Valores permitidos:</u>  <b>status:</b></p> <ul style="list-style-type: none"> <li>• OK</li> <li>• ERROR</li> <li>• FAILED</li> <li>• WARNING</li> </ul>

### Ejemplo

<b>Http Request</b>	/import/singleMovement
<b>Objeto Enviado</b>	<p>1. Agrega saldo para la cuenta 10</p> <pre>{   "account": "10",   "currency": "ARS",   "cash": 6000,   "settlType": "1" }</pre>

<b>Objeto Recibido</b>	<p>1. Importación de saldos realizada con éxito:</p> <pre>{   "status": " OK " }</pre>
------------------------	--

## Importación de Posiciones Iniciales

Permite la importación de posiciones iniciales.

*HTTP Request*

POST /rest/import/initialpositions

*Autenticación*

HTTP Basic Authentication

*Parámetros*

<b>Produces</b>	application/json
<b>Consumes</b>	application/json
<b>Body Object (template)</b>	<pre>Object{   account (String) ,   keepPrevious (boolean) (default =   false) ,   positions (array[InitialPosition]) }  InitialPosition{   account (String) ,   symbolReference (String) ,   size (number) ,   price (number) ,   originalPrice (number) ,   settlType (String) ,   daysParked (number) }  Valores permitidos: settlType: <ul style="list-style-type: none"> <li>• CASH</li> <li>• NEXT_DAY</li> </ul> </pre>

	<ul style="list-style-type: none"> <li>• T_PLUS_2</li> <li>• T_PLUS_3</li> </ul>
<b>Response Object</b>  (template)	<pre>Object{   status(String),   message(String),   data (array[String]) }</pre> <p><u>Valores permitidos:</u></p> <p><u>status:</u></p> <ul style="list-style-type: none"> <li>• OK</li> <li>• ERROR</li> <li>• FAILED</li> </ul> <p><u>message:</u></p> <ul style="list-style-type: none"> <li>• DIFFERENT_ACCOUNTS</li> <li>• INVALID_ACCOUNTS</li> <li>• UNEXPECTED_ACCOUNT</li> <li>• UNKNOWN_CURRENC</li> </ul>

*Ejemplo*

<b>Http Request</b>	/import/initialpositions
<b>Objeto Enviado</b>	<p>1. Elimina todas las posiciones iniciales para la cuenta 10 y crea las nuevas.</p> <pre>{   "account": "10",   "positions": [     {       "account": "10",       "symbolReference": "DLR052024",       "settlType": null,       "size": 500,       "price": 11.65,       "originalPrice": 11.75     },     {       "account": "10",       "symbolReference": "GD30",       "settlType": "T_PLUS_3",       "size": 1,       "price": 189,       "originalPrice": 192,       "daysParked": 3     }   ] }</pre> <p>2. Mantiene todas las posiciones iniciales para todas las cuentas asignadas al usuario autenticado e importa las nuevas.</p> <pre>{   "account": "",   "positions": [     {</pre>



	<pre>                 "account": "10",                 "symbolReference": "DLR052024",                 "settleType": null,                 "size": 500,                 "price": 11.65,                 "originalPrice": 11.75             },             {                 "account": "50",                 "symbolReference": "GD30",                 "settleType": "T_PLUS_3",                 "size": 1,                 "price": 189,                 "originalPrice": 192,                 "daysParked": 3             }         ],         "keepPrevious": true     }         </pre>
<b>Objeto Recibido</b>	<p>1. Importación de posiciones iniciales realizada con éxito:</p> <pre> {   "status": " OK " }         </pre> <p>2. Cuentas mal ingresadas:</p> <pre> {   "status": "ERROR",   "message": "UNEXPECTED_ACCOUNT",   "data": [     "120"   ] }         </pre> <p>3. Cuentas no válidas para el usuario:</p> <pre> {   "status": "ERROR",   "message": "INVALID_ACCOUNTS",   "data": [     "999",     "888",     "777"   ] }         </pre> <p>4. Posiciones repetidas (misma combinación account-symbolReference-settleType):</p> <pre> {   "status": "ERROR",   "message": "INVALID_POSITIONS",   "data": [     "DRL072015 CASH",     "TS"   ] }         </pre> <p>5. Instrumentos no válidos:</p> <pre> {         </pre>

	<pre> "status": "ERROR", "message": "INVALID_INSTRUMENTS", "data": [   "DRL072015_CASH",   "PAMP T_PLUS_2*" ] </pre>
--	--

## Importación de Crédito

Permite la importación de crédito.

*HTTP Request*

POST /rest/import/credits

*Autenticación*

HTTP Basic Authentication

*Parámetros*

<b>Produce</b>	application/json
<b>Consumes</b>	application/json
<b>Body Object (template)</b>	<pre> Object{   account (String) ,   credits (array[credits]) } credits{   account (String) ,   currency (String) ,   amount (number) } </pre>

<b>Response Object</b>  <b>(template)</b>	<pre>Object{     status (String) ,     message (String) ,     data (array[String]) }</pre> <p><u>Valores permitidos:</u></p> <p><i>status:</i></p> <ul style="list-style-type: none"><li>• OK</li><li>• ERROR</li><li>• FAILED</li></ul> <p><i>message:</i></p> <ul style="list-style-type: none"><li>• DIFFERENT_ACCOUNTS</li><li>• INVALID_ACCOUNTS</li><li>• UNEXPECTED_ACCOUNT</li><li>• UNKNOWN_CURENC</li></ul>
---	---

## Anexo - Integraciones en el ámbito de MatrizOMS

Dentro del entorno de MatrizOMS, los endpoints de la API siguen una estructura específica con el prefijo “oms-webservices” en las solicitudes, como se detalla a continuación:

**`/oms-webservices/rest/{recurso}`**

Por ejemplo:

**`/oms-webservices/rest/position/getPositions`**

**`/oms-webservices/rest/import/initialPositions`**