



**Argo Web Trading API
v1.0.0.1**

Copyright 2015 Argo SE, Inc

1 Introduction

Argo Web Trading API is built as a RESTful web service. It can be divided into three sub-domains: order management, market data and risk info. An authorization (logon) procedure is required before any Trading API calls can be made. Typically, Trading API returns data in xml format. Requests are HTTP Get and Post messages. Some requests contain JSON formatted objects.

The following are examples of requests/responses for Argo Web Trading API.

1.1 Authorization

1.1.1 Logon

URL: POST /axis2/services/Argo/logon

Request (JSON):

```
user_id    trader1
password   trader1
```

```
function onLogon()
{
    $.post("http://192.168.1.109:9090/axis2/services/Argo/logon", { user_id: "trader1", password:
    "trader1"}, function(data){ alert("Data Loaded: " + data); }, "xml");
}
```

Response (XML):

```
<logon_result>
  <sessionIdentity>
    <user_id>trader1</user_id>
    <session_id>bc024256-f6b8-40a2-b8d2-cdef04f67817</session_id>
  </sessionIdentity>
</logon_result>
```

1.1.2 Logout

URL: POST /axis2/services/Argo/logout

Request (JSON):

```
SessionIdentity[session_id]  SOME_SESSION_ID
SessionIdentity[user_id]     SOME_USER_ID
```

```
function Logout()
{
    $.post("http://192.168.1.109:9090/axis2/services/Argo/logout", { SessionIdentity: {
    session_id: "SOME_SESSION_ID", user_id: "SOME_USER_ID"}}, function(data){ alert("Data
    Loaded: " + data); }, "xml");
}
```

Response (XML):

```
<logout_result>
</logout_result>
```

1.2 Order Management

1.2.1 New Order

URL: POST /axis2/services/Argo/new_order

Request (JSON):

Order[Account]	ACCOUNT2
Order[Currency]	USD
Order[Destination]	SOME_EXCH
Order[Exchange]	SOME_EXCH
Order[FillOrKillQty]	0
Order[ID]	1357560276276
Order[InitTime]	0
Order[LastQty]	0
Order[LastTime]	0
Order[LimitPrice]	7500
Order[OrderSide]	OS_Buy
Order[OrderState]	oRst_Suspended
Order[OrigQty]	1
Order[Qty]	1
Order[SecId]	24427
Order[StopPrice]	0
Order[Symbol]	ESZ3
Order[TIF]	TF_Day
Order[TotalFillQty]	0
Order[TrailingOffset]	0
Order[Type]	OT_Limit
Order[UserData]	MY COMMENTS
SessionIdentity[session_id]	SOME_SESSION_ID
SessionIdentity[user_id]	SOME_USER_ID

```
function newOrder()
{
  var d = new Date();
  var url = "http://192.168.1.109:9090/axis2/services/Argo/new_order";
  var json = { SessionIdentity: { session_id: "SOME_SESSION_ID", user_id:
    "SOME_USER_ID"},
    Order: { ID : d.getTime(),
      Type: "OT_Limit",
      TIF: "TF_Day",
      OrderState: "oRst_Suspended",
      OrderSide: "OS_Buy",
      Qty: 1,
      OrigQty: 1,
      LastQty: 0,
      LimitPrice: 7500,
      StopPrice: 0,
      TrailingOffset: 0,
      Destination: "SOME_EXCH",
      SecId: "24427",
      Symbol: "ESZ3",
      Account: "ACCOUNT2",
      InitTime: 0,
```

```
        LastTime: 0,  
        TotalFillQty: 0,  
        Currency: "USD",  
        FillOrKillQty: 0,  
        Exchange: "SOME_EXCH",  
        UserData: "MY COMMENTS",  
    }  
};  
$.post(url, json, function(data) {setXmlToFrame(data) }, "xml");  
}
```

Response (XML):

```
<new_order_result>  
  true  
</new_order_result>
```

1.2.2 Replace Order

URL: POST /axis2/services/Argo/replace_order

Request (JSON):

Order[Account]	ACCOUNT2
Order[Currency]	USD
Order[Destination]	SOME_EXCH
Order[Exchange]	SOME_EXCH
Order[FillOrKillQty]	0
Order[ID]	1357560276277
Order[InitTime]	0
Order[LastQty]	0
Order[LastTime]	0
Order[LimitPrice]	7500
Order[OrderSide]	OS_Buy
Order[OrderState]	oRst_Suspended
Order[OrigQty]	1
Order[Qty]	100
Order[SecId]	24427
Order[StopPrice]	0
Order[Symbol]	ESZ3
Order[TIF]	TF_Day
Order[TotalFillQty]	0
Order[TrailingOffset]	0
Order[Type]	OT_Limit
Order[UserData]	MY COMMENTS
SessionIdentity[session_id]	SOME_SESSION_ID
SessionIdentity[user_id]	SOME_USER_ID
order_id	1357560276276

```
function replaceOrder()  
{  
    var d = new Date();  
    var url = "http://192.168.1.109:9090/axis2/services/Argo/replace_order";  
    var json = { SessionIdentity: { session_id: "SOME_SESSION_ID",  
        user_id: "SOME_USER_ID"},  
        order_id: "987654321",  
        Order: { ID : d.getTime(),
```

```
Type:"OT_Limit",
TIF: "TF_Day",
OrderState: "oRst_Suspended",
OrderSide: "OS_Buy",
Qty: 1,
OrigQty: 1,
LastQty: 0,
LimitPrice: 7500,
StopPrice: 0,
TrailingOffset: 0,
Destination: "SOME_EXCH",
SecId: "24427",
Symbol: "ESZ3",
Account: "ACCOUNT2",
InitTime: 0,
LastTime: 0,
TotalFillQty: 0,
Currency: "USD",
FillOrKillQty: 0,
Exchange: "SOME_EXCH",
UserData: "MY COMMENTS",
}
};
$.post(url, json, function(data) { setXmlToFrame(data); }, "xml");
}
```

Response (XML):

```
<replace_order_result>
  true
</replace_order_result>
```

1.2.3 Cancel Order

URL: POST /axis2/services/Argo/cancel_order

Request (JSON):

```
SessionIdentity[session_id]  SOME_SESSION_ID
SessionIdentity[user_id]     SOME_USER_ID
order_id                     1356120043526
```

```
function cancelOrder()
{
  var d = new Date();
  var url = "http://192.168.1.109:9090/axis2/services/Argo/cancel_order";
  var json = { SessionIdentity: { session_id: "SOME_SESSION_ID",
                                user_id: "SOME_USER_ID"}, order_id: "1356120043526"};
  $.post(url, json, function(data) { setXmlToFrame(data); }, "xml");
}
```

Response (XML):

```
<cancel_order_result>
  false
</cancel_order_result>
```

1.2.4 Get Orders

URL: POST /axis2/services/Argo/get_orders

Request specific arguments:

- Account – get orders only for this account (for all if *)
- symbol – get orders for specific instrument (for all if *)
- working_only – get only active orders true/false (otherwise all)
- from – get all orders with init/last time greater than this value (linux epoch)
- to – get all orders with init/last time less than this value (linux epoch)
- by_order_init_time – if true time for “from”/”to” calculation is order init time, otherwise order last time.

Request (JSON):

SessionIdentity[session_id]	SOME_SESSION_ID
SessionIdentity[user_id]	SOME_USER_ID
Account	*
by_order_init_time	true
from	0
symbol	*
to	1357573505800
working_only	false

```
function getOrders()
{
    var d = new Date();
    var url = "http://192.168.1.109:9090/axis2/services/Argo/get_orders";
    var json = { SessionIdentity: { session_id: "SOME_SESSION_ID",
        user_id: "SOME_USER_ID"},
        account: "*",
        symbol: "*",
        from: 0,
        to: d.getTime(),
        by_order_init_time: true,
        working_only: false };
    $.post(url, json, function(data) {setXmlToFrame(data) }, "xml");
}
```

Response (XML):

```
<get_orders_result>
<orders>
<order>
    <ID>1356899399335</ID>
    <OrigID></OrigID>
    <ExchID>61_306708603</ExchID>
    <Type>OT_Limit</Type>
    <TIF>TF_Day</TIF>
    <ExpirationTime>-1</ExpirationTime>
    <OrderState>oRst_Accepted</OrderState>
    <OrderSide>OS_Buy</OrderSide>
    <Qty>2</Qty>
    <OrigQty>2</OrigQty>
    <AvgPx>0</AvgPx>
    <LastQty>0</LastQty>
    <LastPrice>0</LastPrice>
    <LimitPrice>7500</LimitPrice>
```

```
<StopPrice>-2147483648</StopPrice>
<TrailingOffset>0</TrailingOffset>
<Text></Text>
<Destination></Destination>
<SecId>24427</SecId>
<Symbol>d</Symbol>
<Account>ACCOUNT2</Account>
<InitTime>0</InitTime>
<LastTime>1357572730</LastTime>
<TradeId></TradeId>
<TotalFillQty>0</TotalFillQty>
<Currency>USD</Currency>
<FillOrKillQty>0</FillOrKillQty>
<Exchange>SOME_EXCH</Exchange>
<IsReplaced>>true</IsReplaced>
<UserData>MY+COMMENTS</UserData>
<IsStopPriceTriggered>>false</IsStopPriceTriggered>
<IsWorking>>true</IsWorking>
<IsCompleted>>true</IsCompleted>
<IsUnconfirmed>>false</IsUnconfirmed>
</order>
</orders>
<get_orders_result>
```

1.2.5 Get Trades

URL: POST /axis2/services/Argo/get_trades

Request (JSON):

```
SessionIdentity[session_id]  SOME_SESSION_ID
SessionIdentity[user_id]     SOME_USER_ID
account                       *
from                           0
symbol                         *
to 1357573863202
```

```
function getTrades()
{
    var d = new Date();
    var url = "http://192.168.1.109:9090/axis2/services/Argo/get_trades";
    var json = { SessionIdentity: { session_id: "SOME_SESSION_ID", user_id: "SOME_USER_ID"},
                account: "*",
                symbol: "*",
                from: 0,
                to: d.getTime()};
    $.post(url, json, function(data) {setXmlToFrame(data) }, "xml");
}
```

Response (XML):

```
<get_trades_result>
<trades>
<trade>
<OrderID>830976250517</OrderID>
<ExchID>77_306708641</ExchID>
<ID>2047</ID>
```

```
<InitTime>1357484098</InitTime>
<OrderSide>OS_Sell</OrderSide>
<Price>15000</Price>
<LimitPrice>15000</LimitPrice>
<Qty>1</Qty>
<Symbol>ESZ3</Symbol>
<SecId>24427</SecId>
<Currency>USD</Currency>
<Account>RT_ACCOUNT</Account>
<ExchangeDestination>SOME_EXCH</ExchangeDestination>
<TradeState></TradeState>
<IsOtherDayFill>>false</IsOtherDayFill>
<ParentTradeId></ParentTradeId>
<ExecRepFlags>0</ExecRepFlags>
<IsImplied>>false</IsImplied>
<IsEts>>false</IsEts>
<IsCorrected>>false</IsCorrected>
<IsMakingPosition>>false</IsMakingPosition>
<MatchedQty>0</MatchedQty>
<PosQty>0</PosQty>
<Text></Text>
</trade>
</trades>
</get_trades_result>
```

1.3 Market Data

1.3.1 Get Instruments

URL: POST /axis2/services/Argo/get_instrument

Request (JSON):

```
SessionIdentity[session_id]  SOME_SESSION_ID
SessionIdentity[user_id]     SOME_USER_ID
regex_symbol                 ES.*
```

```
function getInstruments()
{
  var url = "http://192.168.1.109:9090/axis2/services/Argo/get_instrument";
  var json = { SessionIdentity: { session_id: "SOME_SESSION_ID", user_id: "SOME_USER_ID"},
    regex_symbol: "ES.*"};
  $.post(url, json, function(data) {setXmlToFrame(data) }, "xml");
}
```

Response (XML):

```
<get_instrument_result>
<instruments>
<instrument>
  <TickSize>2500</TickSize>
  <TickValue>12.5</TickValue>
  <RoundLot>1</RoundLot>
  <Currency>USD</Currency>
  <Numerator>2500</Numerator>
  <Denominator>10000</Denominator>
```



```
<Symbol>ESM3</Symbol>
<SecurityId>19696</SecurityId>
<Exchange>SOME_EXCH</Exchange>
<EtsType>ets_none</EtsType>
<MaturityMonthYear>06/2013</MaturityMonthYear>
<LabelMaturityMonthYear></LabelMaturityMonthYear>
<MaturityDay>1</MaturityDay>
<PriceFormat>pf_none</PriceFormat>
<PutOrCall>en_without</PutOrCall>
<Exercise>ex_american</Exercise>
<Description></Description>
<Underlying>ES</Underlying>
<Expiration>1371821400</Expiration>
<Activation>1331904600</Activation>
<SecurityType>st_fut</SecurityType>
<ISIN></ISIN>
<CUSIP></CUSIP>
<StrikePrice>0</StrikePrice>
<NotionalAmount>0</NotionalAmount>
<ClearingFirmId>0</ClearingFirmId>
<UpfrontPayment>0</UpfrontPayment>
<DisplaySymbol>0</DisplaySymbol>
<Coupon>0</Coupon>
<ClearingId>0</ClearingId>
<StartDate>0</StartDate>
<EndDate>0</EndDate>
<CmeMarketCode></CmeMarketCode>
<MinLotSize>1</MinLotSize>
<UnitOfMeasureQty>1</UnitOfMeasureQty>
<ReferencePrice>10</ReferencePrice>
<TransferSecretary></TransferSecretary>
<Remarks></Remarks>
<IndustrySector></IndustrySector>
<IPOPrice>10.0</IPOPrice>
<Issuer>test</Issuer>
<SettlementTplus>0</SettlementTplus>
<IssuedShares>100</IssuedShares>
<RestrictedShares>1</RestrictedShares>
<IssueDate>1371821400</IssueDate>
</instrument>
</instruments>
</get_instrument_result>
```

1.3.2 Get Depth

URL: POST /axis2/services/Argo/get_depth

Request (JSON):

```
SessionIdentity[session_id] SOME_SESSION_ID
SessionIdentity[user_id] SOME_USER_ID
Symbol ESZ3
```

```
function getDepth()
{
  var url = "http://192.168.1.109:9090/axis2/services/Argo/get_depth";
```

```
var json = { SessionIdentity: { session_id: "SOME_SESSION_ID", user_id: "SOME_USER_ID"},  
symbol: "ESZ3"};  
$.post(url, json, function(data) {setXmlToFrame(data) }, "xml");  
}
```

Response (XML):

```
<get_depth_result>  
  <depth>  
    <buy_levels>  
      <depth_level>  
        <Price>7500</Price>  
        <Qty>295</Qty>  
        <NumOfOrders>20</NumOfOrders>  
      </depth_level>  
    </buy_levels>  
    <sell_levels>  
      <depth_level>  
        <Price>17500</Price>  
        <Qty>2</Qty>  
        <NumOfOrders>2</NumOfOrders>  
      </depth_level>  
    </sell_levels>  
  </depth>  
</get_depth_result>
```

1.3.3 Get Price Info

URL: POST /axis2/services/Argo/get_price_info

Request (JSON):

SessionIdentity[session_id]	SOME_SESSION_ID
SessionIdentity[user_id]	SOME_USER_ID
symbol	ESZ3

```
function getPriceInfo()  
{  
  var url = "http://192.168.1.109:9090/axis2/services/Argo/get_price_info";  
  var json = { SessionIdentity: { session_id: "SOME_SESSION_ID", user_id: "SOME_USER_ID"},  
symbol: "ESZ3"};  
  $.post(url, json, function(data) {setXmlToFrame(data) }, "xml");  
}
```

Response (XML):

```
<get_price_info_result>  
  <price_info>  
    <LastTradePrice>0</LastTradePrice>  
    <BestBidPrice>0</BestBidPrice>  
    <BestBidQty>0</BestBidQty>  
    <BestOfferPrice>0</BestOfferPrice>  
    <BestOfferQty>0</BestOfferQty>  
    <LastTradeQty>0</LastTradeQty>  
    <OpenPrice>0</OpenPrice>  
    <HighPrice>0</HighPrice>  
    <LowPrice>0</LowPrice>  
    <ClosePrice>0</ClosePrice>
```

```
<TotalTradedVolume>0</TotalTradedVolume>
</price_info>
</get_price_info_result>
```

1.3.4 Get News

URL: POST /axis2/services/Argo/get_news

Request (JSON):

```
SessionIdentity[session_id] SOME_SESSION_ID
SessionIdentity[user_id]    SOME_USER_ID
```

```
function getNewsInfo()
{
  var url = "http://192.168.1.109:9090/axis2/services/Argo/get_news";
  var json = { SessionIdentity: { session_id: "SOME_SESSION_ID", user_id: "SOME_USER_ID"} };
  $.post(url, json, function(data) {setXmlToFrame(data) }, "xml");
}
```

Response (XML):

```
<get_news_result>
<news_list>
<news>
  <Text>txt</Text>
  <Headline>hdl</Headline>
  <Symbol>ESZ3</Symbol>
  <Channel>FX</Channel>
  <EventTime>1371821400</EventTime>
</news>
</news_list>
</get_news_result>
```

1.4 Risk Management Calls

1.4.1 Get Accounts

URL: POST /axis2/services/Argo/get_accounts

Request (JSON):

```
SessionIdentity[session_id] SOME_SESSION_ID
SessionIdentity[user_id]    SOME_USER_ID
```

```
function getAccounts()
{
  var url = "http://192.168.1.109:9090/axis2/services/Argo/get_accounts";
  var json = { SessionIdentity: { session_id: "SOME_SESSION_ID", user_id:
  "SOME_USER_ID"} };
  $.post(url, json, function(data) {setXmlToFrame(data) }, "xml");
}
```

Response (XML):

```
<get_accounts_result>
<accounts>
  <account>
    <Id>ACCOUNT1</Id>
```

```
<Name>ACCOUNT1</Name>
<Currency>USD</Currency>
<Status>1</Status>
<RealizedPNL>0</RealizedPNL>
<UnrealizedPNL>0</UnrealizedPNL>
<Balance>100000</Balance>
<NetAssetValue>100000</NetAssetValue>
<UnusedMargin>100000</UnusedMargin>
<UsedMargin>0</UsedMargin>
<TotalValue>0</TotalValue>
<RiskProfileId>rlp2</RiskProfileId>
<SettlementRealizedPNL>0</SettlementRealizedPNL>
<SettlementUnrealizedPNL>0</SettlementUnrealizedPNL>
<ExchangeAccountId>ACCOUNT1</ExchangeAccountId>
<DisplayName></DisplayName>
</account>
</accounts>
</get_accounts_result>
```

1.4.2 Get Positions:

URL: POST /axis2/services/Argo/get_positions

Request (JSON):

```
SessionIdentity[session_id]  SOME_SESSION_ID
SessionIdentity[user_id]    SOME_USER_ID
```

```
function getPositions()
{
  var url = "http://192.168.1.109:9090/axis2/services/Argo/get_positions";
  var json = { SessionIdentity: { session_id: "SOME_SESSION_ID", user_id:
    "SOME_USER_ID" }};
  $.post(url, json, function(data) {setXmlToFrame(data) }, "xml");
}
```

Response (XML):

```
<get_accounts_result>
<positions>
  <position>
    <Id>ACCOUNT2|ESZ3|SOME_EXCH</Id>
    <Account>ACCOUNT2</Account>
    <SecurityId>24427</SecurityId>
    <Symbol>ESZ3</Symbol>
    <Exchange>SOME_EXCH</Exchange>
    <TradePrice>0</TradePrice>
    <AveragePrice>0</AveragePrice>
    <Value>0</Value>
    <RealizedPNL>0</RealizedPNL>
    <UnrealizedPNL>0</UnrealizedPNL>
    <Qty>0</Qty>
    <SettlementUnrealizedPNL>0</SettlementUnrealizedPNL>
    <SettlementRealizedPNL>0</SettlementRealizedPNL>
    <MaintenanceQty>0</MaintenanceQty>
    <InitialQty>0</InitialQty>
    <InitialMargin>0</InitialMargin>
```

Argo Web Trading API

```
<MaintenanceMargin>0</MaintenanceMargin>
<ParentSecurityId/>
<WorkingBuys>292</WorkingBuys>
<WorkingSells>0</WorkingSells>
</position>
<position>
  <Id>RT_ACCOUNT|ESZ3|SOME_EXCH</Id>
  <Account>RT_ACCOUNT</Account>
  <SecurityId>24427</SecurityId>
  <Symbol>ESZ3</Symbol>
  <Exchange>SOME_EXCH</Exchange>
  <TradePrice>0</TradePrice>
  <AveragePrice>0</AveragePrice>
  <Value>0</Value>
  <RealizedPNL>0</RealizedPNL>
  <UnrealizedPNL>0</UnrealizedPNL>
  <Qty>0</Qty>
  <SettlementUnrealizedPNL>0</SettlementUnrealizedPNL>
  <SettlementRealizedPNL>0</SettlementRealizedPNL>
  <MaintenanceQty>0</MaintenanceQty>
  <InitialQty>0</InitialQty>
  <InitialMargin>0</InitialMargin>
  <MaintenanceMargin>0</MaintenanceMargin>
  <ParentSecurityId/>
  <WorkingBuys>3</WorkingBuys>
  <WorkingSells>2</WorkingSells>
</position>
</positions>
</get_accounts_result>
```