



# **Market Data Web Services Specification**

**Version 2.3**

**2-June-2014**

## Revision History

Revised Date	Effective Date	Version	Description
1-Jun-2006	TBD	1.0	Create document (for internal used only)
31-Jul-2006	01-Aug-2006	1.0a	Add more descriptions
15-Jun-2007	03-Sep-2007	1.1	<ul style="list-style-type: none"> <li>- Provide on Internet channel; Apply security policy</li> <li>- Add new service, <i>PasswordServices</i></li> <li>- Add sections of Data Format, System Architecture and add more descriptions</li> </ul>
14-Mar-2011	21-Mar-2011	1.2	<ul style="list-style-type: none"> <li>- Add new service, <i>INAVServices</i></li> <li>- Add authorization for calling service</li> <li>- Add return latest INAV of ETF in <i>StockQuotationServices</i></li> </ul>
10-May-2011	13-Jun-2011	2.0	Add new service, <i>TradeSummaryServices</i> , for TFEX trading data
21-May-2012	03-Sep-2012	2.1	Support New Equity Trading System <ul style="list-style-type: none"> <li>- Update <i>StockQuotationServices</i> <ul style="list-style-type: none"> <li>• Add <i>isOddLot</i> parameter in <i>StockQuotationServices</i></li> <li>• Add <i>isOddLot</i> field in <i>StockQuotation</i></li> <li>• Extend Symbol length from 8 to 20</li> </ul> </li> <li>- Cancel Open field in <i>IndexQuotation</i></li> </ul>
12-Mar-2014	06-May-2014	2.2	<ul style="list-style-type: none"> <li>- Support new Derivatives Trading System               <ul style="list-style-type: none"> <li>• Cancel <i>getInstrumentClassSummary</i> method of <i>TradeSummaryServices</i></li> <li>• Modify <i>TFEXTradeSummary</i> class                   <ul style="list-style-type: none"> <li>▶ Cancel <i>InstrumentType</i>, <i>InstrumentGroupType</i>, <i>InstrumentClass</i>, <i>Underlying</i> and <i>TradingDate</i> fields</li> <li>▶ Add <i>List</i>, <i>Segment</i> and <i>TotalTrade</i> fields</li> </ul> </li> <li>• Cancel <i>TradingDate</i> field in <i>TFEXMostActive</i> class</li> </ul> </li> <li>- Change service name to <b>Market Data Web Services</b></li> <li>- Update document template</li> </ul>
20-May-2014	02-Jun-2014	2.3	<ul style="list-style-type: none"> <li>- Add new URL for market data delayed 15 minutes</li> <li>- Update start of service time to 6.30 a.m. (BKK time)</li> </ul>

## Table of Contents

<b>1</b>	<b>Introduction.....</b>	<b>5</b>
<b>2</b>	<b>System Architecture .....</b>	<b>6</b>
<b>3</b>	<b>Provided Services .....</b>	<b>7</b>
3.1	StockQuotationServices .....	8
3.2	IndexQuotationServices .....	10
3.3	TradeSummaryServices.....	11
3.4	PasswordServices.....	12
3.5	INAVServices .....	13
<b>4</b>	<b>Data Format.....</b>	<b>15</b>
4.1	StockQuotation .....	15
4.2	IndexQuotation .....	17
4.3	TFEXTradeSummary .....	18
4.4	TFEXMostActive.....	19
<b>5</b>	<b>WSDL (Web Services Description Language).....</b>	<b>21</b>
<b>6</b>	<b>System Flow Diagram.....</b>	<b>22</b>
<b>7</b>	<b>Development Guidelines .....</b>	<b>23</b>
7.1	Authentication.....	23
7.2	Error Detection .....	24
7.3	Returned Result .....	24
7.4	Client Flow Chart .....	25
<b>Appendix A: XML Schema for the returned objects .....</b>		<b>28</b>
	XML Schema for StockQuotation.....	28
	XML Schema for IndexQuotation.....	29
	XML Schema for TFEXTradeSummary .....	29
<b>Appendix B: Example Requests .....</b>		<b>31</b>
	Example Requests for StockQuotationServices .....	31
	Example Requests for IndexQuotationServices .....	33
	Example Requests for TradeSummaryServices .....	34
	Example Requests for PasswordServices .....	35
	Example Requests for INAVServices.....	35
<b>Appendix C: Example Responses .....</b>		<b>36</b>
	Example Responses for StockQuotationServices .....	36
	Example Responses for IndexQuotationServices.....	39
	Example Responses for TradeSummaryServices .....	42
	Example Empty Object Response (in case of no data found).....	46



Example Exception Response .....46

**Appendix D: Application Error Codes.....47**

## Market Data Web Services

### 1 Introduction

The “**Market Data Web Services**” is used for the members to query real-time or delayed trading data including stocks and indices of SET, mai, BEX and TFEX. This service will facilitate the clients who do not need to receive all market data feed and manipulate them but just calling Web Services to retrieve only the required information.

The members can call the different URLs for different market data type including real-time or delay, but with the same service name. The server will check authorization of the client to call only permitted services.

- Real-time market data: <http://www.prs.set.or.th/webservice>
- Delayed market data: <http://www.prs.set.or.th/delayservice>

The service is available on Monday to Friday, between 6:30 a.m. and 22:30 p.m. BKK time.

## 2 System Architecture

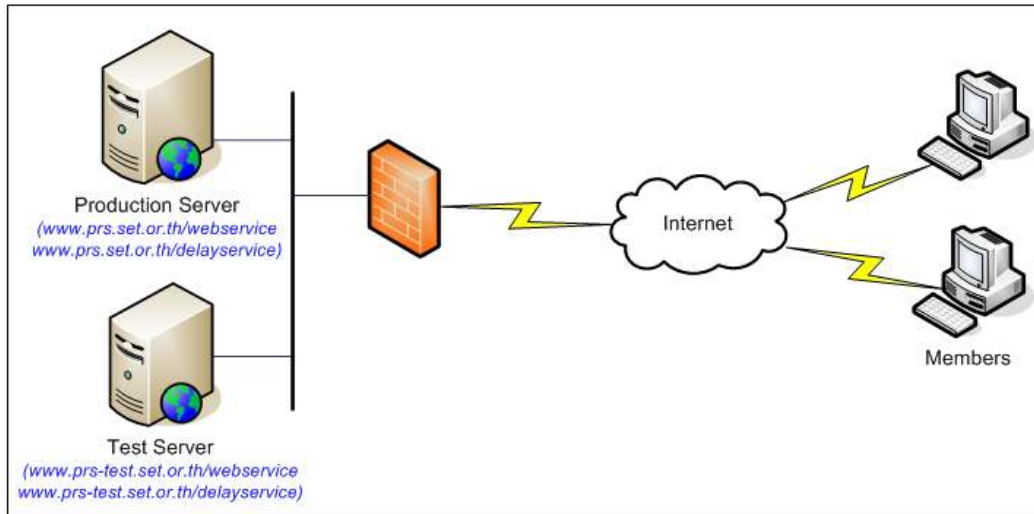


Figure 1 – System Architecture

The clients can access Market Data Web Services via Internet channel. Market Data Web Services provides 2 types of server, Production Server and Test Server. The member can call services from each server by specifying the different URLs. The URL of each server will be described as following;

- ▶ **Production Server**

Real-time market data: <http://www.prs.set.or.th/webservice>

Delayed market data: <http://www.prs.set.or.th/delayservice>

- ▶ **Test Server:** For testing purpose, before the clients live on Production Server.

Real-time market data: <http://www.prs-test.set.or.th/webservice>

Delayed market data: <http://www.prs-test.set.or.th/delayservice>

### 3 Provided Services

Market Data Web Services provide 5 services as the followings:

- ▶ StockQuotationServices
- ▶ IndexQuotationServices
- ▶ TradeSummaryServices
- ▶ PasswordServices
- ▶ INAVServices

Each service contains the operations as below.

Service Name	Operation Name
StockQuotationServices	getStockQuotation
	getStockQuotationByTime
	getStockQuotationByMarket
	getStockQuotationByMarketTime
IndexQuotationServices	getIndexQuotation
	getIndexQuotationByTime
TradeSummaryServices	getMarketSummary
	getMostActiveVolume
PasswordServices	changePassword
INAVServices	updateINAV

The authorization to use any service will be granted by SET. Therefore, the clients can call only permitted services except for PasswordServices which are permitted for all clients. Especially for INAVServices, only AMC clients are permitted to call INAV service to send the updated INAV of their own corresponding ETF funds to the system.

Remark:

- Stock means the stocks listed on SET, mai and BEX
- Index means SET Index, SET100/SET50 Index and mai Index including Industry and Sector Index.
- Market means SET, mai and BEX.
- Time means any time. If there is no information at the time of request, the server will return the latest information before the request time to the client. Please refer to the details of each service.
- Quotation information means Prior, Open, High, Low, 3 Bids, 3 Offers, Last, Projected Open and Average Price.

- If the client requests for a stock quotation by specifying the requested time, the client will receive the quotation information except Bid and Offer prices.

### 3.1 *StockQuotationServices*

There are 4 provided operations for StockQuotationServices as the followings:

- ▶ ***getStockQuotation*** – To get latest quotations of specific stock symbols.
  - If there is no trade, it will return the object containing zero values. The Client may refer to the field Time's value. If the system returns the value of field Time as Zero value, meaning that there is no trade of that stock.
  - INAV information is available only for ETF securities.
- ▶ ***getStockQuotationByTime*** – To get quotations (except bid prices, offer prices and INAV) of specific stock symbols at a specific time.
  - If time is not specified, it will return quotations of all existing execution times.
  - If there is no any trade, it will return empty object.
  - If there is no transaction executed at the time of request, the server will return the quotation of the latest executed transaction before the request time to the client. For example, if the client requests the quotation information at 10:30:00am but at that time there is no transaction executed, the system will return the quotation information of latest executed transaction before the time of request, for example, the one executed at 09:59:30am.
- ▶ ***getStockQuotationByMarket*** – To get latest quotations of all stocks for the specific markets. For example, all stocks of SET.
  - If there is no transaction executed, it will return the quotation containing zero values. The Client may refer to the field Time's value. If the system returns the value of field Time as Zero value, meaning that there is no trade of that stock.
  - INAV information is available only for ETF securities.
- ▶ ***getStockQuotationByMarketTime*** – To get quotations (except bid prices, offer prices and INAV) of all stocks for the specific markets at a specific time.
  - Time is mandatory field. Clients have to specify requested time.
  - If there is no trade, it will return empty object.
  - If there is no transaction executed at the time of request, the server will return the quotation of the latest executed transaction before the request time to the client. For example, if the client requests the quotation information at 10:30:00am but at that time



there is no transaction executed, the system will return the quotation information of latest executed transaction before the time of request, for example, the one executed at 09:59:30am.

Remark: You can see the example of returned result with empty object in Appendix C – Example Responses.

The parameters and returned results of each operation can be described as below table.

Operation Name	Parameter	Returned Results
getStockQuotation	listOfStockSymbol (string), isOddLot (string)	Vector of StockQuotation List
getStockQuotationByTime	listOfStockSymbol (string), isOddLot (string), time (string)	Vector of StockQuotation List
getStockQuotationByMarket	listOfMarket (string), isOddLot (string)	Vector of StockQuotation List
getStockQuotationByMarketTime	listOfMarket (string), isOddLot (string), time (string)	Vector of StockQuotation List

Remark:

- The parameters including Symbols and Markets are separated with semicolon (;) i.e. BBL;ADVANC for symbols and SET;MAI for markets. Symbols and markets are not case-sensitive, for example, you can specify Symbol as “BBL” or “bbl” to retrieve the information of BBL security. You can see examples in Appendix B – Example Requests. In order to get all securities in the specific security type, for example, common stock, DW, ETF, you can also specify Security Type via the parameter of listOfStockSymbol
- There are 3 possible values of isOddLot parameter including “Y”, “N” and “B” as follows:
  - This parameter is **optional**. If you do not specify, the default value is “N”. This means the server will return data only for Round lot market.
  - If you specify isOddLot=Y. The server will return data of Odd lot market.
  - If you specify isOddLot=B. The server will return data of both Round lot and Odd lot market.
- You can see the details of StockQuotation from Appendix A – XML Schema for the returned objects, and examples from Appendix C – Example Responses.
- The possible values of Market ID, Security Type and isOddLot are shown in WSDL section of MarketEnum, SecurityTypeEnum and IsOddLotEnum data type respectively.
- The format of parameter “time” is “hhmmss” i.e. 103000.

### 3.2 IndexQuotationServices

There are 2 provided operations for IndexQuotationServices as the followings:

- ▶ **getIndexQuotation** -- To get latest quotations of specific index symbols including index of market, industry and sector.
  - If there is no trade, it still returns the object containing zero values.
- ▶ **getIndexQuotationByTime** – To get quotations of specific index symbols including index of market, industry and sector at a specific time.
  - If time is not specified, it will return index quotations of all existing times.
  - Before opening market, there is no Index Value, the system will return empty object to the client.
  - If there is no information at the time of request, the server will return Index quotation of the latest one before the request time to the client. For example, if the client request Index quotation information at 10:30:00am but at that time there is no information, the system will return the quotation information of latest one before the time of request, for example, the one at 09:59:30am.

The parameters and returned results of each operation can be described as below table.

Operation Name	Parameter	Returned Results
getIndexQuotation	listOfIndex (string)	Vector of IndexQuotation List
getIndexQuotationByTime	listOfIndex (string), time (string)	Vector of IndexQuotation List

Remark:

- The parameters are separated with semicolon ( ; ) i.e. BANK;ENERG. They are case-insensitive, for example, you can specify Index Name as “BANK” or “bank” to retrieve the information of Bank Sectoral Index. You can see examples in Appendix B – Example Requests.
- You can see the details of IndexQuotation from Appendix A – XML Schema for the returned objects, and examples from Appendix C – Example Responses.
- The possible values of Index Name including market ID, industry name and sector name are shown in WSDL section of IndexEnum, IndustryEnum and SectorEnum data type respectively.
- The format of parameter “time” is “hhmmss” i.e. 103000.

### 3.3 TradeSummaryServices

There are 2 provided operations for TradeSummaryServices which are available only for TFEX trading data as the followings:

- ▶ **getMarketSummary** – To get trading summary for the specific IDs including MarketCode, List and Segment.
  - If the client request trade summary for MarketCode, returned values for List and Segment field will be blank.
  - If the client request trade summary for List, returned values for Segment field will be blank.
- ▶ **getMostActiveVolume** – To get most active by volume for the specific IDs including MarketCode, List, Segment and its ranking.

The parameters and returned results of each operation can be described as below table.

Operation Name	Parameter	Returned Results
getMarketSummary	listOfID (string), session(string)	Vector of TFEXTradeSummary
getMostActiveVolume	listOfRanking (string in JSON <sup>1</sup> format) including the following fields: <ul style="list-style-type: none"> <li>- ID (string)</li> <li>- rank (number)</li> </ul> , session(string)	Vector of MostActive

Remark:

- The listOfID parameters are separated with semicolon ( ; ) i.e. S50IF;SF. They are case-insensitive.
- The format of listOfRanking parameter can be defined as below. The clients can request more than one by separated each ID with brace ( { ... } ). For example, listOfRanking=[{ID:"S50IF",rank:1},{ID:"SF",rank:2}]
- The possible values of session parameter is 'D' (Day), 'N' (Night) and ' ' (Current session)

---

<sup>1</sup> JSON (JavaScript Object Notation) is a lightweight data-interchange format. It is based on a subset of the JavaScript Programming Language, Standard ECMA-262 3rd Edition - December 1999. JSON is a text format that is completely language independent but uses conventions that are familiar to programmers of the C-family of languages, including C, C++, C#, Java, JavaScript, Perl, Python, and many others.

- If the specific ID does not trade in the specific session, for example, TXI does not trade in the Night session, the system will return the object containing zero values.

### 3.4 PasswordServices

- ▶ **changePassword** -- Clients' password can be changed in accordance with SET's password policy as follows:

- The password length must be 8 to 16 characters.
- The password must be different from login name and reversible of login name.  
For example, if your login name is "12345", your new password must not be "12345" and "54321".
- The password must be different from 5 previous ones and must not be the one used within the past 6 months.
- The password must contain characters, numbers and special character (e.g.: \$, #, ? or !).

Please see more details about SET security policy in item "6.1 Authentication" in "Development Guidelines" section.

The parameters and returned results of each operation can be described as below table.

Operation Name	Parameter	Returned Results
changePassword	newPassword (string)	Boolean (true) if success / Exception Response if invalid password policy

### 3.5 INAVServices

- ▶ **updateINAV** – To update indicative NAV of corresponding ETF funds of each AMC with the following details:
  - The Security Symbol, which is submitted to the system by an AMC, must be the symbol of an ETF that is issued by that AMC itself.
  - Each AMC should submit INAV to the system starting from approx. 8.05 am - 5.00 pm, in every 15-30 seconds, of a trading day. Any INAV, submitted to the system before 8.05 am and after 5.00 pm, will not be broadcasted to the members.
  - Before the Market open,
    - If INAV of an ETF is unchanged, AMC should submit the latest INAV of the previous trading day to the system.
    - If INAV of an ETF (especially the ETF that uses the foreign stock as an underlying) is changed due to 1) the price of underlying stock at home market is changed or b) the exchange rate is changed, AMC will calculate and submit the updated INAV to the system.
  - Change and Percent Change are calculated from the difference between the latest INAV and NAV of the previous day.
  - INAV and Change are provided with 4 digit decimals.
  - Time is the calculation time from the AMC side.

The parameters and returned results of each operation can be described as below table.

Operation Name	Parameter	Returned Results
updateINAV	listOfINAV (string in JSON format) including the following fields for each fund: <ul style="list-style-type: none"> <li>- symbol (string)</li> <li>- iNAV (number)</li> <li>- chg (number)</li> <li>- percentChg (number)</li> <li>- time (string)</li> </ul>	Boolean (true) if success / Exception Response if any symbol can not be updated

Remark:

- The parameters are separated with comma ( , ). They are case-insensitive.
- The clients can update more than 1 ETF by separated each fund with brace ( { ... } ). For example,  

```
listOfINAV=[{symbol:"ETF1",iNAV:10.1234,chg:0.1234,percentChg:0.12,time:"095930"},{symbol:"ETF2",iNAV:9.1234,chg:-0.8765,percentChg:-0.87,time:"095930"}]
```

```
listOfNAV={{symbol:"ETF3",iNAV:10.4567,chg:0.4567,percentChg:0.45,time:"100100"}}
```

- If clients update iNAV of more than one ETF in one request and some error occurs, iNAV of all ETFs in that request will be ignored.
- For the symbol and time fields, which are in string format, the clients have to send the value of those fields under double quotation marks ( " " ).
- The format of parameter "time" is "hhmmss" i.e. "095915".

## 4 Data Format

### 4.1 StockQuotation

No.	Field Name	Description	Data Type	Size	Possible Value/Format
1	Symbol	Security symbol	String	20	
2	Name	Security long name	String	32	
3	isOddLot	Is odd lot flag	Boolean		
4	Prior	Prior closing price (If there is no trades yesterday, it will carry latest closing price)	Numeric	double	
5	Open	Opening price of the day	Numeric	double	
6	Project1	Projected open price (Morning session)	Numeric	double	
7	Project2	Projected open price (Afternoon session)	Numeric	double	
8	High	Highest price of the day	Numeric	double	
9	Low	Lowest price of the day	Numeric	double	
10	Last	Last executed price	Numeric	double	
11	Average	Average price	Numeric	double	
12	Volume	Last volume	Numeric	double	
13	Value	Last value	Numeric	double	
14	TotalVolume	Accumulated volume	Numeric	double	
15	TotalValue	Accumulated value	Numeric	double	
16	Bid	Top 3 bid (buy) orders			
17	- Rank	Rank number	Numeric	int	1-3
18	- Price	Bid price (if bid price is zero, it means ATO <sup>2</sup> or ATC <sup>3</sup> order which not specify price)	Numeric	double	
19	- Volume	Bid volume	Numeric	double	
20	Offer	Top 3 offer (sell) orders			

<sup>2</sup> ATO (At-The-Open) is the buy or sell at the opening price orders during Pre-open period of both morning and afternoon session.

<sup>3</sup> ATC (At-The-Close) is the buy or sell at the closing price orders during Call market period.

No.	Field Name	Description	Data Type	Size	Possible Value/Format
21	- Rank	Rank number	Numeric	int	1-3
22	- Price	Offer price (if offer price is zero, it means ATO or ATC order which not specify price)	Numeric	double	
23	- Volume	Offer volume	Numeric	double	
24	INAV	Indicative NAV of ETF			
25	- INAV	Indicative NAV	Numeric	double	
26	- Chg	The difference between latest INAV and NAV of the previous day	Numeric	double	
27	- PercentChg	The percentage of difference between latest INAV and NAV of the previous day	Numeric	double	
28	- Time	INAV calculation time which is stamped from the AMC side	Numeric	int	
29	Time	Updated time	Numeric	int	



#### 4.2 IndexQuotation

No.	Field Name	Description	Data Type	Size	Possible Value/Format
1	Symbol	Index symbol	String	8	
2	Name	Index full name	String	50	
3	Prior	Prior closing index	Numeric	double	
4	High	Highest index of the day	Numeric	double	
5	Low	Lowest index of the day	Numeric	double	
6	Last	Last index	Numeric	double	
7	Volume	Last volume	Numeric	double	
8	Value	Last value	Numeric	double	
9	TotalVolume	Accumulated volume	Numeric	double	
10	TotalValue	Accumulated value	Numeric	double	
11	Time	Index calculation time	Numeric	int	

Remark:

- Index includes index of market, industry, sector and subsector (if available for each exchange).
- Each index will be updated around every 15 seconds.

#### 4.3 *TFEXTradeSummary*

No.	Field Name	Description	Data Type	Size	Possible Value/Format
1	Symbol	Symbol	String	17	
2	MarketCode	Market Code	String	5	
3	List	List under Market	String	10	
4	Segment	Segment under Market List	String	10	
5	TotalTrade	Total Transactions	Numeric	double	
6	TotalVolume	Total Volume	Numeric	double	
7	OpenInterest	Open Interest of previous trading day	Numeric	double	
8	Time	Last update time	Numeric	int	

Remark: Trading in Night Session will be considered as trading of the next trading day.

#### 4.4 TFEXMostActive

No.	Field Name	Description	Data Type	Size	Possible Value/Format
1	Rank	Most active ranking number	Numeric	int	
2	Symbol	Security symbol	String	17	
3	PriorSettle	Settlement price of the previous trading date	Numeric	Double	
4	Settle	Settlement price of the current trading date	Numeric	Double	
5	Open	Opening AOM price of the day	Numeric	Double	
6	High	Highest AOM price of the day	Numeric	Double	
7	Low	Lowest AOM price of the day	Numeric	Double	
8	Last	Last AOM price	Numeric	Double	
9	Chg	The difference between latest AOM price (or settlement price if market closed) and prior settlement price	Numeric	Decimal	Nullable
10	PercentChg	The difference between latest AOM price (or settlement price if market closed) and prior settlement price	Numeric	Decimal	Nullable
11	Volume	Last AOM volume	Numeric	Double	
12	TotalVolume	Accumulated AOM and BT volume	Numeric	Double	
13	Bid	Top 3 bid (buy) orders			
14	- Rank	Bid ranking number	Numeric	Int	1-3
15	- Price	Bid price (if bid price is zero, it means MP order which not specify price)	Numeric	Double	
16	- Volume	Bid volume	Numeric	Double	
17	Offer	Top 3 offer (sell) orders			
18	- Rank	Offer ranking number	Numeric	Int	1-3
19	- Price	Offer price (if offer price is zero, it means	Numeric	Double	

No.	Field Name	Description	Data Type	Size	Possible Value/Format
		MP order which not specify price)			
20	- Volume	Offer volume	Numeric	Double	
21	Time	Last update time	Numeric	int	

## Remark:

- Trading in Night Session will be considered as trading of the next trading day.If Chg and PercentChg value are null, meaning that the system cannot calculate such value. For example,
  - On first trading date, there is no prior settlement price for comparison.
  - On last trading date of Options series, the type of final settlement price is different from the type of daily settlement price. The daily settlement price will be sent in premium value while the final settlement price will be sent in spot price. Therefore, the system cannot compare the final settlement price with the prior (daily) settlement price.

## 5 WSDL (Web Services Description Language)

The member can get WSDL of each service by specifying URL as follows:

- ▶ WSDL of StockQuotationServices

Real-time: <http://www.prs.set.or.th/webservice/StockQuotationServices?wsdl>

Delay: <http://www.prs.set.or.th/delayservice/StockQuotationServices?wsdl>

- ▶ WSDL of IndexQuotationServices

Real-time: <http://www.prs.set.or.th/webservice/IndexQuotationServices?wsdl>

Delay: <http://www.prs.set.or.th/delayservice/IndexQuotationServices?wsdl>

- ▶ WSDL of TradeSummaryServices

Real-time: <http://www.prs.set.or.th/webservice/TradeSummaryServices?wsdl>

Delay: <http://www.prs.set.or.th/delayservice/TradeSummaryServices?wsdl>

- ▶ WSDL of PasswordServices

Real-time: <http://www.prs.set.or.th/webservice/PasswordServices?wsdl>

Delay: <http://www.prs.set.or.th/delaybservice/PasswordServices?wsdl>

- ▶ WSDL of INAVServices

Real-time: <http://www.prs.set.or.th/webservice/INAVServices?wsdl>

## 6 System Flow Diagram

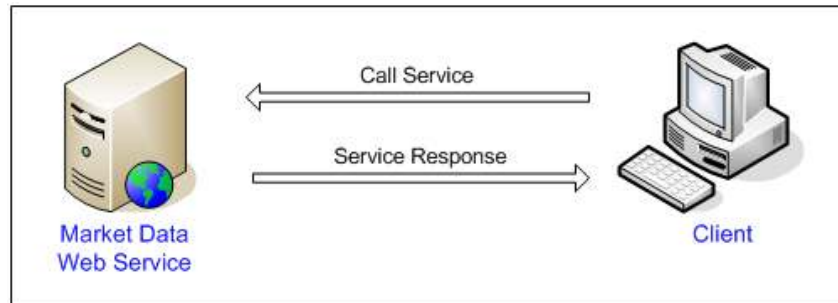


Figure 2 – System Flow Diagram

When client calls any services from Market Data Web Services server, user name and password must be submitted to the server for authentication. After calling service, the client has to detect returned message from the server whether there is any error occurred. If the error has not been found, the client can retrieve data from the returned result.

Normally the client is able to automatically call services from program at client site. The member can follow “*Development Guidelines*” in next section for program development.

However the member can test calling services by specifying URL directly on browser. For example, if you would like to get the latest prices of BBL and ADVANC, you can specify URL as follow:

<http://www.prs.set.or.th/webservice/StockQuotationServices?method=getStockQuotation&listOfStockSymbol=bbl;advanc>

Your browser will show popup windows and wait for you to enter user name and password for authentication. After authenticated, the server will return the results which will be shown on client’s browser.

## 7 Development Guidelines

### 7.1 Authentication

The Market Data Web Services authenticates the clients from the following values:

- User name
- Password
- Client IP

The authentication will be processed every time when the client calls service. Only permitted clients can call services from Market Data Web Services.

Whenever client calls services from Market Data Web Services server, **User name** and **Password** must be submitted to the server for authentication. For the **Client IP**, the server can automatically detect from the requested string. For clients who call services over internet channel, Client IP will be detected by using global IP.

For secure authentication through internet, the server uses **Digest access authentication** method. The password will be encrypted by using MD5 algorithm, therefore, it is not sent in plaintext over the network.

If the clients authenticate successfully, the server will return results of requested service to the client. But if the clients authenticate unsuccessfully, the server will return error message to the client. You can see the example of exception response in Appendix C – Example Responses.

Remark: User name and password will be assigned by SET. Moreover, the member is required to report global IP to SET.

#### SET Security Policy

According to SET's security policy, the members should prepare your programs to follow the following policies:

- ▶ The clients will be forced to change their passwords before they are permitted to call any services from Market Data Web Services due to these reasons: 1) it is the first time login; 2) passwords are reset by SET administrator in case those users forget their passwords; and 3) password is expired.

The clients can call *PasswordServices* in order to change your passwords.

- ▶ Password will be expired every 3 months period. SET will send alert e-mail to the members in advance.

If the members have not yet changed their passwords until they are expired, the server will force clients to change their password when clients login to the system next time.

- ▶ Login will be disabled in case the clients try to login with incorrect password up to 5 times.
- ▶ Clients' password can be changed in accordance with SET's password policy as follows:
  - The password length must be 8 to16 characters.
  - The password must be different from login name and reversible of login name.  
For example, if your login name is "12345", your new password must not be "12345" and "54321".
  - The password must be different from 5 previous ones and must not be the one used within the past 6 months.
  - The password must contains characters, numbers and special character (e.g.: \$, #, ? or !).

## **7.2 Error Detection**

After calling service, the client has to detect returned message whether there is any error occurred. The errors may be caused by authentication process, parameters validation or there are some problems while processing data. If there is any error, the server will return error exception to the client and then the client can get error codes and error strings from such exception. The client can identify the cause of error from the returned error codes or error strings. Please see the possible error codes in Appendix D – Application Error Codes.

## **7.3 Returned Result**

The results of StockQuotationServices and IndexQuotationServices are returned in **Vector of object** (StockQuotation and IndexQuotation respectively). The number of items in returned vector depends on the specified parameters. It may contain only 1 item if the client request data for only 1 stock symbol. While it may contain many items if the client request the data of many stock symbols or markets.



#### 7.4 Client Flow Chart

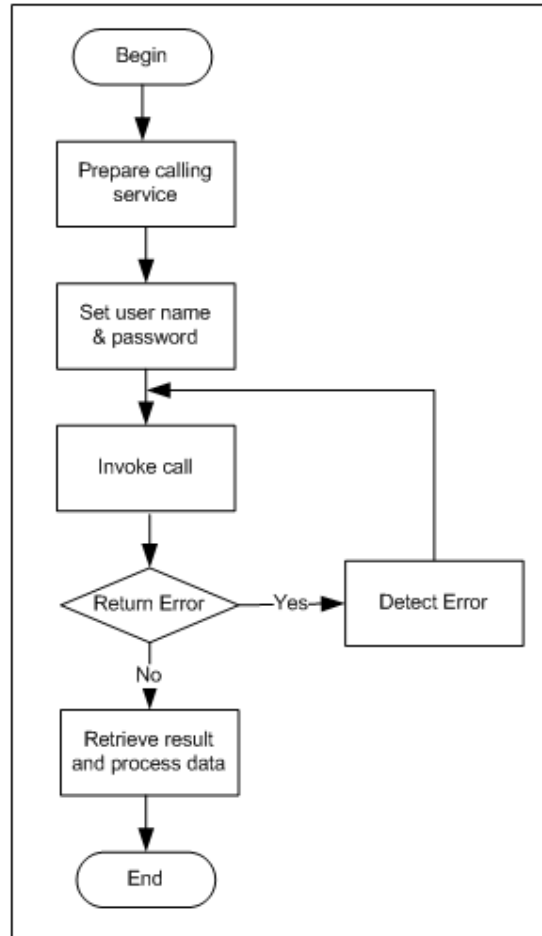


Figure 3 – Client Flow Chart

The programming steps of client are shown in figure 3 – Client Flow Chart. It includes the following steps:

- Step 1) Prepare calling service
- Step 2) Set user name and password
- Step 3) Invoke call
- Step 4) Detect Error
- Step 5) Retrieve result and process data

The example of JAVA source code for each step is shown below.

```
import org.apache.axis.*;
import org.apache.axis.configuration.SimpleProvider;
import org.apache.axis.transport.http.CommonsHTTPSender;
import org.apache.axis.transport.http.HTTPConstants;
import org.apache.axis.client.AxisClient;
import org.apache.axis.client.Call;
import org.apache.axis.client.Service;
import org.apache.axis.encoding.XMLType;
import org.apache.axis.encoding.ser.BeanSerializerFactory;
import org.apache.axis.encoding.ser.BeanDeserializerFactory;
import javax.xml.namespace.QName;
import javax.xml.rpc.ParameterMode;
import java.net.URL;
import java.util.Vector;
import java.util.Hashtable;

/-- Step 1. Prepare calling service
// Set XML Tag with namespace
QName stockQuotationQName = new QName("http://www.prs.set.or.th/webservice", "StockQuotation");
QName topPriceQName = new QName("http:// www.prs.set.or.th/webservice", "TopPrice");

// Set Service Endpoint
String endpoint = "http://www.prs.set.or.th/webservice/StockQuotationServices";
SimpleProvider engineConfiguration = new SimpleProvider();
engineConfiguration.deployTransport("http", new SimpleTargetedChain(new CommonsHTTPSender()));
Service service = new Service(engineConfiguration);

// Don't use multirefs option
service.getEngine().setOption(AxisClient.PROP_DOMULTIREFS, Boolean.FALSE);
Call call = (Call)service.createCall();

// Set Mapping between XML Tag and Java Class
call.registerTypeMapping(StockQuotation.class, stockQuotationQName, new BeanSerializerFactory
    (StockQuotation.class, stockQuotationQName), new BeanDeserializerFactory (StockQuotation.class,
    stockQuotationQName));
call.registerTypeMapping(TopPrice.class, topPriceQName, new BeanSerializerFactory (TopPrice.class,
    topPriceQName), new BeanDeserializerFactory (TopPrice.class, topPriceQName));

// Set Call Target Endpoint
call.setTargetEndpointAddress(new URL(endpoint));

/-- Set Property don't use chunk property
Hashtable myheader = new Hashtable();
myheader.put(HTTPConstants.HEADER_TRANSFER_ENCODING_CHUNKED, false);
call.setProperty(HTTPConstants.REQUEST_HEADERS, myheader);

// Set Operation Name
call.setOperationName(new QName("StockQuotationServices", "getStockQuotation"));

// Set Parameters and Return of Operation
call.addParameter("listOfStockSymbol", XMLType.SOAP_STRING, ParameterMode.IN);
call.setReturnType(new QName("http://xml.apache.org/xml-soap", "Vector"), Vector.class);

Object[] listOfSymbol = {"BBL;ADVANC"}; // BBL,ADVANC are stock symbol
```

```
//-- Step 2. Set user name and password
```

```
call.setUsername("username");  
call.setPassword("password");
```

```
try {
```

```
//-- Step 3. Invoke call
```

```
Vector v = (Vector)call.invoke(listOfSymbol);
```

```
//-- Step 5. Retrieve result and process data
```

```
System.out.println(v);
```

```
...
```

```
} catch(AxisFault e) {
```

```
//-- Step 4. Detect error
```

```
System.out.println("FaultCode[" + e.getFaultCode() + "]);
```

```
e.printStackTrace();
```

```
}
```

---

## Appendix A: XML Schema for the returned objects

### XML Schema for StockQuotation

```

<schema targetNamespace="http://www.prs.set.or.th/webservice"
  xmlns="http://www.w3.org/2001/XMLSchema">
  <import namespace="http://xml.apache.org/xml-soap" />
  <import namespace="http://enumeration.marketdata.datastore.dds.set.or.th" />
  <import namespace="http://schemas.xmlsoap.org/soap/encoding/" />
  <complexType abstract="true" name="INAV">
    <sequence>
      <element name="INAV" type="xsd:double" />
      <element name="Chg" type="xsd:double" />
      <element name="PercentChg" type="xsd:double" />
      <element name="Time" type="xsd:int" />
    </sequence>
  </complexType>
  <complexType abstract="true" name="StockQuotation">
    <sequence>
      <element name="Symbol" type="soapenc:string" />
      <element name="Name" type="soapenc:string" />
      <element name="IsOddLot" type="xsd:boolean" />
      <element name="Prior" type="xsd:double" />
      <element name="Open" type="xsd:double" />
      <element name="Project1" type="xsd:double" />
      <element name="Project2" type="xsd:double" />
      <element name="High" type="xsd:double" />
      <element name="Low" type="xsd:double" />
      <element name="Last" type="xsd:double" />
      <element name="Average" type="xsd:double" />
      <element name="Volume" type="xsd:double" />
      <element name="Value" type="xsd:double" />
      <element name="TotalVolume" type="xsd:double" />
      <element name="TotalValue" type="xsd:double" />
      <element name="Bid" type="apachesoap:Vector" /> <!-- Vector of
        TopPrice -->
      <element name="Offer" type="apachesoap:Vector" /> <!-- Vector of
        TopPrice -->
      <element name="INAV" nillable="true" type="impl:INAV" />
      <element name="Time" type="xsd:int" />
    </sequence>
  </complexType>
  <complexType abstract="true" name="TopPrice">
    <sequence>
      <element name="Price" type="xsd:double" />
      <element name="Volume" type="xsd:double" />
    </sequence>
    <attribute name="Rank" type="xsd:int" />
  </complexType>
</schema>

```

### XML Schema for IndexQuotation

```
<schema targetNamespace="http://www.prs.set.or.th/webService"
  xmlns="http://www.w3.org/2001/XMLSchema">
  <import namespace="http://xml.apache.org/xml-soap" />
  <import namespace="http://enumeration.marketdata.datastore.dds.set.or.th" />
  <import namespace="http://schemas.xmlsoap.org/soap/encoding/" />
  <complexType abstract="true" name="IndexQuotation">
    <sequence>
      <element name="Symbol" type="soapenc:string" />
      <element name="Name" type="soapenc:string" />
      <element name="Prior" type="xsd:double" />
      <element name="High" type="xsd:double" />
      <element name="Low" type="xsd:double" />
      <element name="Last" type="xsd:double" />
      <element name="Volume" type="xsd:double" />
      <element name="Value" type="xsd:double" />
      <element name="TotalVolume" type="xsd:double" />
      <element name="TotalValue" type="xsd:double" />
      <element name="Time" type="xsd:int" />
    </sequence>
  </complexType>
</schema>
```

### XML Schema for TFEXTradeSummary

```
<schema targetNamespace="http://www.prs.set.or.th/webService"
  xmlns="http://www.w3.org/2001/XMLSchema">
  <import namespace="http://xml.apache.org/xml-soap" />
  <import namespace="http://enumeration.marketdata.datastore.dds.set.or.th" />
  <import namespace="http://schemas.xmlsoap.org/soap/encoding/" />
  <complexType abstract="true" name="TFEXTradeSummary">
    <sequence>
      <element name="Symbol" type="soapenc:string" />
      <element name="MarketCode" type="soapenc:string" />
      <element name="List" type="soapenc:string" />
      <element name="Segment" type="soapenc:string" />
      <element name="TotalTrade" type="xsd:double" /> <!-- AOM + PT -->
      <element name="TotalVolume" type="xsd:double" /> <!-- AOM + PT -->
      <element name="OpenInterest" type="xsd:double" />
      <element name="Time" type="xsd:int" />
    </sequence>
  </complexType>
</schema>
```

**XML Schema for MostActive**

```

<schema targetNamespace="http://www.prs.set.or.th/webservice"
  xmlns="http://www.w3.org/2001/XMLSchema">
  <import namespace="http://xml.apache.org/xml-soap" />
  <import namespace="http://enumeration.marketdata.datastore.dds.set.or.th" />
  <import namespace="http://schemas.xmlsoap.org/soap/encoding/" />
  <complexType abstract="true" name="MostActive">
    <sequence>
      <element name="ID" type="soapenc:string" />
      <element name="MostActiveList" type="apachesoap:Vector" /> <!--
        Vector of TFEXMostActive -->
    </sequence>
  </complexType>
  <complexType abstract="true" name="TFEXMostActive">
    <sequence>
      <element name="Symbol" type="soapenc:string" />
      <element name="PriorSettle" type="xsd:double" />
      <element name="Settle" type="xsd:double" />
      <element name="Open" type="xsd:double" />
      <element name="High" type="xsd:double" />
      <element name="Low" type="xsd:double" />
      <element name="Last" type="xsd:double" /> <!-- Last AOM price -->
      <element name="Chg" nillable="true" type="soapenc:decimal" />
      <element name="PercentChg" nillable="true" type="soapenc:decimal" />
      <element name="Volume" type="xsd:double" /> <!-- Last AOM volume -->
      <element name="TotalVolume" type="xsd:double" /> <!-- AOM + BT -->
      <element name="OpenInterest" type="xsd:double" /> <!-- OI of previous
        day -->
      <element name="Bid" type="apachesoap:Vector" /> <!-- Vector of
        TopPrice -->
      <element name="Offer" type="apachesoap:Vector" /> <!-- Vector of
        TopPrice -->
      <element name="Time" type="xsd:int" />
    </sequence>
    <attribute name="Rank" type="xsd:int" />
  </complexType>
  <complexType abstract="true" name="TopPrice">
    <sequence>
      <element name="Price" type="xsd:double" />
      <element name="Volume" type="xsd:double" />
    </sequence>
    <attribute name="Rank" type="xsd:int" />
  </complexType>
</schema>

```

## Appendix B: Example Requests

### Example Requests for StockQuotationServices

- ▶ getStockQuotation

<http://www.prs.set.or.th/webservice/StockQuotationServices?method=getStockQuotation&listOfStockSymbol=bbl;advanc&isOddLot=N>

```
<?xml version="1.0" encoding="utf-8" ?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <getStockQuotationRequest xmlns="http://www.prs.set.or.th/webservice/">
      <listOfStockName>BBL;ADVANC</listOfStockName>
      <isOddLot>N</isOddLot>
    </getStockQuotationRequest>
  </soap:Body>
</soap:Envelope>
```

- ▶ getStockQuotationByTime

<http://www.prs.set.or.th/webservice/StockQuotationServices?method=getStockQuotationByTime&listOfStockSymbol=bbl;advanc&isOddLot=N&time=103000>

```
<?xml version="1.0" encoding="utf-8" ?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <getStockQuotationByTimeRequest
      xmlns="http://www.prs.set.or.th/webservice/">
      <listOfStockName>BBL;ADVANC</listOfStockName>
      <isOddLot>N</isOddLot>
      <time>103000</time>
    </getStockQuotationByTimeRequest>
  </soap:Body>
</soap:Envelope>
```

▶ `getStockQuotationByMarket`

[http://www.prs.set.or.th/webservice/StockQuotationServices?method=getStockQuotationByMarket  
&listOfMarket=set;mai&isOddLot=N](http://www.prs.set.or.th/webservice/StockQuotationServices?method=getStockQuotationByMarket&listOfMarket=set;mai&isOddLot=N)

```
<?xml version="1.0" encoding="utf-8" ?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <getStockQuotationByMarketRequest
      xmlns="http://www.prs.set.or.th/webservice/">
      <listOfMarket>SET;MAI</listOfMarket>
      <isOddLot>N</isOddLot>
    </getStockQuotationByMarketRequest>
  </soap:Body>
</soap:Envelope>
```

▶ `getStockQuotationByMarketTime`

[http://www.prs.set.or.th/webservice/StockQuotationServices?method=getStockQuotationByMarket  
Time&listOfMarket=set;mai&isOddLot=N&time=103000](http://www.prs.set.or.th/webservice/StockQuotationServices?method=getStockQuotationByMarketTime&listOfMarket=set;mai&isOddLot=N&time=103000)

```
<?xml version="1.0" encoding="utf-8" ?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <getStockQuotationByMarketTimeRequest
      xmlns="http://www.prs.set.or.th/webservice/">
      <listOfMarket>SET;MAI</listOfMarket>
      <isOddLot>N</isOddLot>
      <time>103000</time>
    </getStockQuotationByMarketTimeRequest>
  </soap:Body>
</soap:Envelope>
```



**Example Requests for IndexQuotationServices**

## ▶ getIndexQuotation

<http://www.prs.set.or.th/webservice/IndexQuotationServices?method=getIndexQuotation&listOfIndex=bank;energ>

```
<?xml version="1.0" encoding="utf-8" ?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <getIndexQuotationRequest xmlns="http://www.prs.set.or.th/webservice/">
      <listOfIndex>BANK;ENERG</listOfIndex>
    </getIndexQuotationRequest>
  </soap:Body>
</soap:Envelope>
```

## ▶ getIndexQuotationByTime

<http://www.prs.set.or.th/webservice/IndexQuotationServices?method=getIndexQuotationByTime&listOfIndex=bank;energ&time=103000>

```
<?xml version="1.0" encoding="utf-8" ?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <getIndexQuotationByTimeRequest
      xmlns="http://www.prs.set.or.th/webservice/">
      <listOfIndex>BANK;ENERG</listOfIndex>
      <time>103000</time>
    </getIndexQuotationByTimeRequest>
  </soap:Body>
</soap:Envelope>
```

**Example Requests for TradeSummaryServices**

## ▶ getMarketSummary

<http://www.prs.set.or.th/webservice/TradeSummaryServices?method=getMarketSummary&listOfID=s50if;sf&session=>

```
<?xml version="1.0" encoding="utf-8" ?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
<soap:Body>
  <getMarketSummaryRequest xmlns="http://www.prs.set.or.th/webservice/">
    <listOfID>S50IF;SF</listOfID>
    <session>D</session>
  </getMarketSummaryRequest>
</soap:Body>
</soap:Envelope>
```

## ▶ getMostActiveVolume

<http://www.prs.set.or.th/webservice/TradeSummaryServices?method=getMostActiveVolume&listOfRanking=%5B%7BID%3A%22S50IF%22%2Crank%3A1%7D%2C%7BID%3A%22SF%22%2Crank%3A2%7D%5D&session=>

```
<?xml version="1.0" encoding="utf-8" ?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
<soap:Body>
  <getMostActiveVolumeRequest xmlns="http://www.prs.set.or.th/webservice/">
    <listOfRanking>[{"ID":"S50IF",rank:1},{ID:"SF",rank:2}]</listOfRanking>
    <session>D</session>
  </getMostActiveVolumeRequest>
</soap:Body>
</soap:Envelope>
```

**Example Requests for PasswordServices**

- ▶ changePassword

<http://www.prs.set.or.th/webservice/PasswordServices?method=changePassword&newPassword=password>

```
<?xml version="1.0" encoding="utf-8" ?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
<soap:Body>
  <changePasswordRequest xmlns="http://www.prs.set.or.th/webservice/">
    <newPassword>password</newPassword>
  </changePasswordRequest>
</soap:Body>
</soap:Envelope>
```

**Example Requests for INAVServices**

- ▶ updateINAV

<http://www.prs.set.or.th/webservice/INAVServices?method=updateINAV&listOfINAV=%5B%7Bsymbol%3A%22ETF1%22%2CiNAV%3A10.1234%2Cchg%3A0.1234%2CpercentChg%3A0.12%2Ctime%3A%22095930%22%7D%2C%7Bsymbol%3A%22ETF2%22%2CiNAV%3A9.1234%2Cchg%3A-0.8765%2CpercentChg%3A-0.87%2Ctime%3A%22095930%22%7D%5D>

```
<?xml version="1.0" encoding="utf-8" ?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
<soap:Body>
  <updateNAVRequest xmlns="http://www.prs.set.or.th/webservice/">
    <listOfINAV>[{symbol:"ETF1",iNAV:10.1234,chg:0.1234,percentChg:0.12,time:"095930"},{symbol:"ETF2",iNAV:9.1234,chg:-0.8765,percentChg:-0.87,time:"095930"}] </listOfINAV>
  </updateNAVRequest>
</soap:Body>
</soap:Envelope>
```

## Appendix C: Example Responses

### Example Responses for StockQuotationServices

- ▶ `getStockQuotation / getStockQuotationByMarket`

```

<?xml version="1.0" encoding="UTF-8" ?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Header>
  <getStockQuotationResult xsi:type="ns1:Vector"
    xmlns:ns1="http://xml.apache.org/xml-soap">
  <item xsi:type="ns1:Vector">
    <item xsi:type="ns2:StockQuotation"
      xmlns:ns2="http://www.prs.set.or.th/webservice">
      <Symbol xsi:type="soapenc:string"
        xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">TFTSE</Sym
        bol>
      <Name xsi:type="soapenc:string"
        xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">THAIDEX
        FTSE SET LARGE</Name>
      <IsOddLot xsi:type="xsd:boolean">false</IsOddLot>
      <Prior xsi:type="xsd:double">11.89</Prior>
      <Open xsi:type="xsd:double">12.0</Open>
      <Project1 xsi:type="xsd:double">0.0</Project1>
      <Project2 xsi:type="xsd:double">0.0</Project2>
      <High xsi:type="xsd:double">12.0</High>
      <Low xsi:type="xsd:double">11.89</Low>
      <Last xsi:type="xsd:double">11.9</Last>
      <Average xsi:type="xsd:double">11.91</Average>
      <Volume xsi:type="xsd:double">30000.0</Volume>
      <Value xsi:type="xsd:double">357000.0</Value>
      <TotalVolume xsi:type="xsd:double">87600.0</TotalVolume>
      <TotalValue xsi:type="xsd:double">1043000.0</TotalValue>
      <Bid xsi:type="ns1:Vector">
        <item Rank="1" xsi:type="ns2:TopPrice">
          <Price xsi:type="xsd:double">11.89</Price>
          <Volume xsi:type="xsd:double">35000.0</Volume>
        </item>
        <item Rank="2" xsi:type="ns2:TopPrice">
          <Price xsi:type="xsd:double">11.87</Price>
          <Volume xsi:type="xsd:double">22400.0</Volume>
        </item>
        <item Rank="3" xsi:type="ns2:TopPrice">
          <Price xsi:type="xsd:double">11.45</Price>
          <Volume xsi:type="xsd:double">600.0</Volume>
        </item>
      </Bid>
      <Offer xsi:type="ns1:Vector">
        <item Rank="1" xsi:type="ns2:TopPrice">
          <Price xsi:type="xsd:double">11.91</Price>
          <Volume xsi:type="xsd:double">13000.0</Volume>
        </item>
        <item Rank="2" xsi:type="ns2:TopPrice">
          <Price xsi:type="xsd:double">11.93</Price>
          <Volume xsi:type="xsd:double">10400.0</Volume>

```

```

    </item>
    <item Rank="3" xsi:type="ns2:TopPrice">
      <Price xsi:type="xsd:double">11.97</Price>
      <Volume xsi:type="xsd:double">9500.0</Volume>
    </item>
  </Offer>
  <INAV xsi:type="ns2:INAV">
    <INAV xsi:type="xsd:double">11.9463</INAV>
    <Chg xsi:type="xsd:double">0.0377</Chg>
    <PercentChg xsi:type="xsd:double">0.31</PercentChg>
    <Time xsi:type="xsd:int">163910</Time>
  </INAV>
  <Time xsi:type="xsd:int">163905</Time>
</item>
</item>
</getStockQuotationResult>
</soapenv:Header>
<soapenv:Body>
<getStockQuotationResponse
  soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
</soapenv:Body>
</soapenv:Envelope>

```

▶ getStockQuotationByTime / getStockQuotationByMarketTime

```

<?xml version="1.0" encoding="UTF-8" ?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Header>
  <getStockQuotationResult xsi:type="ns1:Vector"
    xmlns:ns1="http://xml.apache.org/xml-soap">
    <item xsi:type="ns1:Vector">
      <item xsi:type="ns2:StockQuotation"
        xmlns:ns2="http://www.prs.set.or.th/webservice">
        <Symbol xsi:type="soapenc:string"
          xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">BBL</Symbol>
        <Name xsi:type="soapenc:string"
          xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">Bangkok Bank
          Limited</Name>
        <IsOddLot xsi:type="xsd:boolean">>false</IsOddLot>
        <Prior xsi:type="xsd:double">109.0</Prior>
        <Open xsi:type="xsd:double">110.0</Open>
        <Project1 xsi:type="xsd:double">111.0</Project1>
        <Project2 xsi:type="xsd:double">115.0</Project2>
        <High xsi:type="xsd:double">115.0</High>
        <Low xsi:type="xsd:double">110.0</Low>
        <Last xsi:type="xsd:double">112.0</Last>
        <Average xsi:type="xsd:double">113.0</Average>
        <Volume xsi:type="xsd:double">101.0</Volume>
        <Value xsi:type="xsd:double">10100.0</Value>
        <TotalVolume xsi:type="xsd:double">10000.0</TotalVolume>
        <TotalValue xsi:type="xsd:double">1.0E7</TotalValue>
        <Bid xsi:type="ns1:Vector"/>
        <Offer xsi:type="ns1:Vector"/>
        <INAV xsi:type="ns2:INAV" xsi:nil="true" />
        <Time xsi:type="xsd:int">103000</Time>

```

```
</item>

<item xsi:type="ns3:StockQuotation"
  xmlns:ns3="http://www.prs.set.or.th/webservice">
  <Symbol xsi:type="soapenc:string"
    xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">BBL</Symbol>
  <Name> ... </Name>
  <IsOddLot> ... </IsOddLot>
  <Prior> ... </Prior>
  <Open> ... </Open>
  <Project1> ... </Project1>
  <Project2> ... </Project2>
  <High> ... </High>
  <Low> ... </Low>
  <Last> ... </Last>
  <Average> ... </Average>
  <Volume > ... </Volume >
  <Value > ... </Value >
  <TotalVolume > ... </TotalVolume >
  <TotalValue > ... </TotalValue >
  <Bid/>
  <Offer/>
  <INAV xsi:type="ns2:INAV" xsi:nil="true" />
  <Time xsi:type="xsd:int">103130</Time>
</item>
</item>

<item xsi:type="ns1:Vector">
  <item xsi:type="ns4:StockQuotation"
    xmlns:ns4="http://www.prs.set.or.th/webservice">
    <Symbol xsi:type="soapenc:string"
      xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">ADVANC</Symbol>
    <Name> ... </Name>
    <IsOddLot> ... </IsOddLot>
    <Prior> ... </Prior>
    <Open> ... </Open>
    <Project1> ... </Project1>
    <Project2> ... </Project2>
    <High> ... </High>
    <Low> ... </Low>
    <Last> ... </Last>
    <Average> ... </Average>
    <Volume > ... </Volume >
    <Value > ... </Value >
    <TotalVolume > ... </TotalVolume >
    <TotalValue > ... </TotalValue >
    <Bid/>
    <Offer/>
    <INAV xsi:type="ns2:INAV" xsi:nil="true" />
    <Time xsi:type="xsd:int">103010</Time>
  </item>
  <item xsi:type="ns5:StockQuotation"
    xmlns:ns4="http://www.prs.set.or.th/webservice">
    <Symbol xsi:type="soapenc:string"
      xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">ADVANC</Symbol>
    <Name> ... </Name>
    <IsOddLot> ... </IsOddLot>
    <Prior> ... </Prior>
    <Open> ... </Open>
```

```

    <Project1> ... </Project1>
    <Project2> ...</Project2>
    <High> ... </High>
    <Low> ... </Low>
    <Last> ... </Last>
    <Average> ... </Average>
    <Volume > ... </Volume >
    <Value > ... </Value >
    <TotalVolume > ... </TotalVolume >
    <TotalValue > ... </TotalValue >
    <Bid/>
    <Offer/>
    <INAV xsi:type="ns2:INAV" xsi:nil="true" />
    <Time xsi:type="xsd:int">103320</Time>
  </item>
</item>

...

</getStockQuotationResult>
</soapenv:Header>
<soapenv:Body>
<getStockQuotationResponse
  soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
</soapenv:Body>
</soapenv:Envelope>

```

### Example Responses for IndexQuotationServices

▶ getIndexQuotation

```

<?xml version="1.0" encoding="UTF-8" ?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Header>
  <getIndexQuotationResult xsi:type="ns1:Vector"
    xmlns:ns1="http://xml.apache.org/xml-soap">
    <item xsi:type="ns1:Vector">
      <item xsi:type="ns2:IndexQuotation"
        xmlns:ns3="http://www.prs.set.or.th/webservice">
        <Symbol xsi:type="soapenc:string"
          xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">BANK</Sym
          bol>
        <Name xsi:type="soapenc:string"
          xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">Banking</N
          ame>
        <Prior xsi:type="xsd:double">900.0</Prior>
        <High xsi:type="xsd:double">1100.0</High>
        <Low xsi:type="xsd:double">999.0</Low>
        <Last xsi:type="xsd:double">1000.0</Last>
        <Volume xsi:type="xsd:double">1001.0</Volume>
        <Value xsi:type="xsd:double">1001000.0</Value>
        <TotalVolume xsi:type="xsd:double">10000.0</TotalVolume>
        <TotalValue xsi:type="xsd:double">1.1111111E7</TotalValue>
        <Time xsi:type="xsd:int">103100</Time>
      </item>
    </item>
  </item>

```

```

<item xsi:type="ns1:Vector">
  <item xsi:type="ns3:IndexQuotation"
    xmlns:ns5="http://www.prs.set.or.th/webservice">
    <Symbol xsi:type="soapenc:string"
      xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">ENERG</Sym
      mbol>
    <Name xsi:type="soapenc:string"
      xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">Energy &
      Utilities</Name>
    <Prior xsi:type="xsd:double">900.0</Prior>
    <High xsi:type="xsd:double">1100.0</High>
    <Low xsi:type="xsd:double">999.0</Low>
    <Last xsi:type="xsd:double">1000.0</Last>
    <Volume xsi:type="xsd:double">1001.0</Volume>
    <Value xsi:type="xsd:double">1001000.0</Value>
    <TotalVolume xsi:type="xsd:double">10000.0</TotalVolume>
    <TotalValue xsi:type="xsd:double">1.0E7</TotalValue>
    <Time xsi:type="xsd:int">103100</Time>
  </item>
</item>
...
</getIndexQuotationResult>
</soapenv:Header>
<soapenv:Body>
  <getIndexQuotationResponse
    soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
</soapenv:Body>
</soapenv:Envelope>

```

► getIndexQuotationByTime

```

<?xml version="1.0" encoding="UTF-8" ?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<soapenv:Header>
<getIndexQuotationResult xsi:type="ns1:Vector"
  xmlns:ns1="http://xml.apache.org/xml-soap">
<item xsi:type="ns1:Vector">
  <item xsi:type="ns2:IndexQuotation"
    xmlns:ns2="http://www.prs.set.or.th/webservice">
    <Symbol xsi:type="soapenc:string"
      xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">BANK</Sym
      bol>
    <Name xsi:type="soapenc:string"
      xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">Banking</N
      ame>
    <Prior xsi:type="xsd:double">900.0</Prior>
    <High xsi:type="xsd:double">1100.0</High>
    <Low xsi:type="xsd:double">999.0</Low>
    <Last xsi:type="xsd:double">1000.0</Last>
    <Volume xsi:type="xsd:double">1001.0</Volume>
    <Value xsi:type="xsd:double">1001000.0</Value>
    <TotalVolume xsi:type="xsd:double">10000.0</TotalVolume>

```



```
<TotalValue xsi:type="xsd:double">1.23456789012E11</TotalValue>
<Time xsi:type="xsd:int">103000</Time>
</item>
<item xsi:type="ns3:IndexQuotation"
  xmlns:ns3="http://www.prs.set.or.th/webservice">
  <Symbol xsi:type="soapenc:string"
    xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">BANK</Sym
    bol>
  <Name xsi:type="soapenc:string"
    xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">Banking</N
    ame>
  <Prior xsi:type="xsd:double">900.0</Prior>
  <High xsi:type="xsd:double">1100.0</High>
  <Low xsi:type="xsd:double">999.0</Low>
  <Last xsi:type="xsd:double">1000.0</Last>
  <Volume xsi:type="xsd:double">1001.0</Volume>
  <Value xsi:type="xsd:double">1001000.0</Value>
  <TotalVolume xsi:type="xsd:double">10000.0</TotalVolume>
  <TotalValue xsi:type="xsd:double">1.111111E7</TotalValue>
  <Time xsi:type="xsd:int">103100</Time>
</item>
</item>
<item xsi:type="ns1:Vector">
  <item xsi:type="ns4:IndexQuotation"
    xmlns:ns4="http://www.prs.set.or.th/webservice">
    <Symbol xsi:type="soapenc:string"
      xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">ENERG</Sy
      mbol>
    <Name xsi:type="soapenc:string"
      xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">Energy &
      Utilities</Name>
    <Prior xsi:type="xsd:double">900.0</Prior>
    <High xsi:type="xsd:double">1100.0</High>
    <Low xsi:type="xsd:double">999.0</Low>
    <Last xsi:type="xsd:double">1000.0</Last>
    <Volume xsi:type="xsd:double">1001.0</Volume>
    <Value xsi:type="xsd:double">1001000.0</Value>
    <TotalVolume xsi:type="xsd:double">10000.0</TotalVolume>
    <TotalValue xsi:type="xsd:double">1.0E7</TotalValue>
    <Time xsi:type="xsd:int">103000</Time>
  </item>
  <item xsi:type="ns5:IndexQuotation"
    xmlns:ns5="http://www.prs.set.or.th/webservice">
    <Symbol xsi:type="soapenc:string"
      xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">ENERG</Sy
      mbol>
    <Name xsi:type="soapenc:string"
      xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">Energy &
      Utilities</Name>
    <Prior xsi:type="xsd:double">900.0</Prior>
    <High xsi:type="xsd:double">1100.0</High>
    <Low xsi:type="xsd:double">999.0</Low>
    <Last xsi:type="xsd:double">1000.0</Last>
    <Volume xsi:type="xsd:double">1001.0</Volume>
    <Value xsi:type="xsd:double">1001000.0</Value>
    <TotalVolume xsi:type="xsd:double">10000.0</TotalVolume>
    <TotalValue xsi:type="xsd:double">1.0E7</TotalValue>
    <Time xsi:type="xsd:int">103100</Time>
  </item>
</item>
```

```

</item>
</item>
...
</getIndexQuotationResult>
</soapenv:Header>
<soapenv:Body>
  <getIndexQuotationResponse
    soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
</soapenv:Body>
</soapenv:Envelope>

```

### Example Responses for TradeSummaryServices

▶ getMarketSummary

```

<?xml version="1.0" encoding="UTF-8" ?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Header>
  <getMarketSummaryResult xsi:type="ns1:Vector"
    xmlns:ns1="http://xml.apache.org/xml-soap">
    <item xsi:type="ns1:Vector">
      <item xsi:type="ns2:TFEXTradeSummary "
        xmlns:ns2="http://www.prs.set.or.th/webservice">
        <Symbol xsi:type="soapenc:string"
          xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"> S50IF
        </Symbol>
        <MarketCode xsi:type="soapenc:string"
          xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"> TXI</Market
          Code>
        <List xsi:type="soapenc:string"
          xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"> IF</List>
        <Segment xsi:type="soapenc:string"
          xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"> S50IF</S
          egment>
        <TotalTrade xsi:type="xsd:double"> 238.0</TotalTrade>
        <TotalVolume xsi:type="xsd:double"> 3338.0</TotalVolume>
        <OpenInterest xsi:type="xsd:double"> 25750.0</OpenInterest>
        <Time xsi:type="xsd:int"> 143905</Time>
      </item>
      <item xsi:type="ns3:TFEXTradeSummary "
        xmlns:ns2="http://www.prs.set.or.th/webservice">
        <Symbol xsi:type="soapenc:string"
          xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"> SF</Symbol
          >
        <MarketCode xsi:type="soapenc:string"
          xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"> TXS</Market
          Code>
        <List xsi:type="soapenc:string"
          xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"> SF</List>
        <Segment xsi:type="soapenc:string"
          xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"></Segmen
          t>
        <TotalTrade > ... </TotalTrade >

```

```

    <TotalVolume > ... </TotalVolume >
    <OpenInterest> ... </OpenInterest>
    <Time> ... </Time>
  </item>
  ...
</item>
</getMarketSummaryResult>
</soapenv:Header>
<soapenv:Body>
<getMarketSummaryResponse
  soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
</soapenv:Body>
</soapenv:Envelope>

```

► getMostActiveVolume

```

<?xml version="1.0" encoding="UTF-8" ?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Header>
  <getMostActiveVolumeResult xsi:type="ns1:Vector"
    xmlns:ns1="http://xml.apache.org/xml-soap">
    <item xsi:type="ns1:Vector">
      <item xsi:type="ns2:MostActive"
        xmlns:ns2="http://www.prs.set.or.th/webservice">
        <ID xsi:type="soapenc:string"
          xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">S50IF</Sym
          bol>
        <MostActiveList xsi:type="ns1:Vector">
          <item Rank="1" xsi:type="ns2:TFEXMostActive"
            xmlns:ns2="http://www.prs.set.or.th/webservice">
            <Symbol xsi:type="soapenc:string"
              xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">S50Z10
            </Symbol>
            <PriorSettle xsi:type="xsd:double">710.7</Prior>
            <Settle xsi:type="xsd:double">0.0</Prior>
            <Open xsi:type="xsd:double">712.0</Open>
            <High xsi:type="xsd:double">712.0</High>
            <Low xsi:type="xsd:double">711.8</Low>
            <Last xsi:type="xsd:double">711.9</Last>
            <Chg xsi:type="soapenc:decimal"
              xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">1.2<
              /Chg>
            <PercentChg xsi:type="soapenc:decimal"
              xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">0.17
            </PercentChg >
            <Volume xsi:type="xsd:double">3.0</Volume>
            <TotalVolume xsi:type="xsd:double">876.0</TotalVolume>
            <OpenInterest xsi:type="xsd:double">176.0</OpenInterest>
            <Bid xsi:type="ns1:Vector">
              <item Rank="1" xsi:type="ns2:TopPrice">
                <Price xsi:type="xsd:double">711.8</Price>
                <Volume xsi:type="xsd:double">5.0</Volume>
              </item>
              <item Rank="2" xsi:type="ns2:TopPrice">
                <Price xsi:type="xsd:double">711.7</Price>

```

```

        <Volume xsi:type="xsd:double">2.0</Volume>
    </item>
    <item Rank="3" xsi:type="ns2:TopPrice">
        <Price xsi:type="xsd:double">711.4</Price>
        <Volume xsi:type="xsd:double">6.0</Volume>
    </item>
</Bid>
<Offer xsi:type="ns1:Vector">
    <item Rank="1" xsi:type="ns2:TopPrice">
        <Price xsi:type="xsd:double">711.9</Price>
        <Volume xsi:type="xsd:double">3.0</Volume>
    </item>
    <item Rank="2" xsi:type="ns2:TopPrice">
        <Price xsi:type="xsd:double">712.0</Price>
        <Volume xsi:type="xsd:double">4.0</Volume>
    </item>
    <item Rank="3" xsi:type="ns2:TopPrice">
        <Price xsi:type="xsd:double">712.3</Price>
        <Volume xsi:type="xsd:double">5.0</Volume>
    </item>
</Offer>
    <Time xsi:type="xsd:int">143905</Time>
</item>
</MostActiveList>
</item>
</item>
<item xsi:type="ns1:Vector">
    <item xsi:type="ns2:MostActive"
        xmlns:ns2="http://www.prs.set.or.th/webservice">
        <ID xsi:type="soapenc:string"
            xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">SF</Symbol>
    <MostActiveList xsi:type="ns1:Vector">
        <item Rank="1" xsi:type="ns2:TFEXMostActive"
            xmlns:ns2="http://www.prs.set.or.th/webservice">
            <Symbol xsi:type="soapenc:string"
                xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">KTBZ10</Symbol>
            <PriorSettle> ... </PriorSettle>
            <Settle> ... </Settle>
            <Open> ... </Open>
            <High> ... </High>
            <Low> ... </Low>
            <Last> ... </Last>
            <Chg> ... </Chg>
            <PercentChg> ... </PercentChg>
            <Volume> ... </Volume>
            <TotalVolume> ... </TotalVolume>
            <OpenInterest> ... </OpenInterest>
        </Bid/>
        </Offer/>
        <Time> ... </Time>
    </item>
    <item Rank="2" xsi:type="ns2:TFEXMostActive"
        xmlns:ns2="http://www.prs.set.or.th/webservice">
        <Symbol xsi:type="soapenc:string"
            xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">ITDZ10</Symbol>
        <PriorSettle> ... </PriorSettle>
        <Settle> ... </Settle>
        <Open> ... </Open>
    </item>

```

```
<High> ... </High>
<Low> ... </Low>
<Last> ... </Last>
<Chg> ... </Chg>
<PercentChg> ... </PercentChg>
<Volume> ... </Volume>
<TotalVolume> ... </TotalVolume>
<OpenInterest> ... </OpenInterest>
<Bid/>
<Offer/>
<Time> ... </Time>
</item>
</MostActiveList>
</item>
</item>
...
</item>
</getMostActiveVolumeResult>
</soapenv:Header>
<soapenv:Body>
<getMostActiveVolumeResponse
  soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
</soapenv:Body>
</soapenv:Envelope>
```

Example Responses for PasswordServices and INAVServices

- ▶ In case of success

```
<?xml version="1.0" encoding="UTF-8" ?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <changePasswordResponse
      soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
      <changePasswordResult xsi:type="xsd:boolean">true</changePasswordResult>
    </changePasswordResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

Example Empty Object Response (in case of no data found)

```
<?xml version="1.0" encoding="UTF-8" ?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <getStockQuotationResponse
      soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
      <getStockQuotationResult xsi:type="ns1:Vector"
        xmlns:ns1="http://xml.apache.org/xml-soap" />
    </getStockQuotationResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

Example Exception Response

```
<?xml version="1.0" encoding="UTF-8" ?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <soapenv:Fault>
      <faultcode>1003</faultcode>
      <faultstring>Invalid Client IP</faultstring>
      <detail>
        <ns1:exceptionName
          xmlns:ns1="http://xml.apache.org/axis/">th.or.set.dds.webservices.exception.AuthenticationException
        </ns1:exceptionName>
      </detail>
    </soapenv:Fault>
  </soapenv:Body>
</soapenv:Envelope>
```

## Appendix D: Application Error Codes

The possible returned error codes from PRS Web Services are shown in the below table. The clients can detect Error Code from returned tag <faultcode>.

Error Code	Description
<b>Authentication and Authorization Error Codes</b>	
1001	Invalid User
1003	Invalid Client IP
1004	Force change password
1005	User is disabled
1010	Do not allow to call the requested service [ <i>Service name</i> ]
1011	Do not allow to call the requested symbol [ <i>Symbol</i> ]
<b>Change Password Services Error Codes</b>	
1101	Invalid password policy, password length must be 8 to16 characters.
1102	Invalid password policy, password must be different from login name and reversible of login name.
1103	Invalid password policy, password must be different from 5 previous ones and must not be the one used within the past 6 months.
1104	Invalid password policy, password must contain characters, numbers and special character
<b>Services Error Codes</b>	
2001	Invalid Argument[listOfStockSymbol] : listOfStockSymbol is empty
2002	Invalid Argument[listOfStockSymbol] : Invalid Stock Symbols
2101	Invalid Argument[listOfMarket] : listOfMarket is empty
2102	Invalid Argument[listOfMarket] : Invalid Market ID
2201	Invalid Argument[listOfIndex] : listOfIndex is empty
2202	Invalid Argument[listOfIndex] : Invalid Index
2301	Invalid Argument[Time] : Invalid Time Format
2302	Invalid Argument[Time] : Time is empty
2401	Invalid Argument[listOfINAV] : listOfINAV is empty
2402	Invalid Argument[listOfINAV] : Invalid listOfINAV [ <i>Invalid detail</i> ]
2501	Invalid Argument[listOfID] : listOfID is empty
2502	Invalid Argument[listOfID] : Invalid ID
2601	Invalid Argument[listOfIRanking] : listOfIRanking is empty
2602	Invalid Argument[listOfIRanking] : Invalid listOfIRanking [ <i>Invalid detail</i> ]
2701	Invalid Argument[session] : session is empty
2702	Invalid Argument[session] : Invalid session

Error Code	Description
<b>Other Error Codes</b>	
9999	System Error

Moreover there are the other error messages returned from Web Server which shown in the below table.

Error String	Description
<b>Authentication Error</b>	
HTTP Status 401	Specify Invalid User or Invalid Password
<b>Services Error</b>	
No such operation '<Operation Name>'	Specify invalid operation name or invalid parameters
The AXIS engine could not find a target service to invoke! targetService is <Service Name>	Specify invalid service name

Note – For the error 'Invalid User', client may receive returned error either 'HTTP Status 401' from Web Server or error code '1001' from PRS Web Services.