Artesian

White Paper

Accelerating time to insight

> *"Data is the new oil. It's valuable, but if unrefined it cannot really be used. It has to be changed into gas, plastic, chemicals, etc. to create a valuable entity that drives profitable activity; so must data be broken down, analysed for it to have value."*
> Clive Humby, UK Mathematician and architect of Tesco's Club card (2006)

The metaphor above, coined back in 2006 by Clive Humby, architect of the Tesco clubcard program, outlines very well the issue faced by the increasing availability of data and the difficulties faced by practitioners when attempting to conform heterogeneous feeds of raw data in order to process it with algorithms in order to derive insights to support forecasting, strategic planning, to underpin operational choices and/or supporting decision making.

Data preparation, also known as *wrangling,* is a mandatory step to the extraction of value from raw data just like refining crude oil into gas, plastics, and chemicals is a mandatory step to its utilisation in industrial processes. To extend the metaphor above, systems tasked to perform data preparation can be likened to crude oil refineries in the context of an information economy, fulfilling the essential supply chain task of providing the required raw materials to insight factories (i.e. algorithms).

Furthermore, the pervasiveness of cloud computing and growing availability of internet connected devices is progressively increasing the types and volumes of raw data available, making it more difficult for businesses to wrangle it into forms suitable for the application of analytics in a consistent and timely fashion.

In Gartner's *Market Guide for Data Preparation* (revision of 10th January 2018) the authors stressed how *"[...] business is demanding faster time to insight than ever before to remain competitive, [...] insights are being derived from larger numbers of diverse data sources [...] with varying degrees of trustworthiness. Amid this increasing complexity and business urgency, business people are challenged to get the insights they need in time without IT assistance."*
Business users are being forced to "build their own refineries" in order to gather the required insights and remain competitive.

This is an undesirable situation from an enterprise perspective for a number of reasons, primarily:

- It reduces trust in the derived insights. This is due to the fact that a business executed, self-build approach to data wranglers tends to create multiple versions of the truth as different people process the same data in different fashions for their own purposes and frequently errors are introduced or simplifications are made that affect reuse or, when this is attempted, can introduce errors where incorrect assumptions are made.
- The resulting artefacts are rarely scalable and IT departments are often faced with the problem of having to optimise multiple versions of poorly performing data wrangling tools that have been crafted, with admirable Corinthian spirit, by less IT literate colleagues. IT departments are thus presented with the impossible task of assisting in the support of tools that have over time become institutionalised and business critical but that have not been designed with supportability in mind causing suboptimal utilisation of people's time and skills.
- The resulting fragmented data consumption pipeline stands as an obstacle to the democratisation of access to prepared data.

Building, operating and maintaining industry strength processes able to reliably wrangle large volumes of heterogeneous raw data feeds in a performant fashion is a resource intensive process and with this approach, it significantly affects the time that can be dedicated to analysis and insight building. In its survey Gartner outlines how *"organizations report that they spend more than 60% of their time in data preparation, leaving little time for actual analysis [...]"*.

What is Artesian?

Artesian aims to relieve energy markets practitioners and data scientists interested in the energy space from the complexity and burden of time series data preparation allowing them to focus on deriving insights from the data available to them through their own analytical models.

Our mission is to make Artesian a core building block of the data economy by hiding the complexity inherent in preparing, persisting and mediating access to varied forms of data across heterogeneous architectures rendering them immediately exploitable in analytical models.

To achieve this end, Artesian is not agnostic with respect to the data sources supported and correctly processes and stores the data supplied in a way that supports the creation of multi-source models quickly and straightforwardly, leaving the user time to focus on the analytics rather than data engineering and wrangling.

Artesian adopts a common and novel approach to data representation and transformation. In order to hide the complexity of utilizing data sourced in different forms, the system is designed to explicitly address the following areas:

- Conformance: map heterogeneous data sources into a set of data classes with known properties and based on a standardized set of descriptive metadata. Provide a framework for the application of standard and custom data transformation techniques;

- Persistence: provide an abstraction layer enabling conformed data in each different data class to be persisted and retrieved/distributed;

- Search and classification: standardize the approach to the application of metadata to the different data classes under management in order to facilitate search and retrieval of the appropriate data items;

- Security and privacy: provide a framework for the application of access management criteria and privacy filters to the data under management thus supporting new collaboration.

This white paper aims to familiarise the reader with the functionality and architecture of **Artesian.**

Who is Artesian designed for?

Artesian has been designed with energy market practitioners in mind: traders, quants, risk analysts, marketers, consultants and in general anyone that requires timely access to data presented in a homogeneous fashion across a variety of data sources in order to power their own models, dashboards or algo-trading algorithms.

Artesian aims to deal with most commonly applied transformations in order to allow practitioners to focus on information and knowledge generation rather than data manipulation.

Granularity, time and relative to absolute period transformations (e.g. Jan 21 => M+1 - assuming of course we are in Dec 20) are handled out of the box. It also provides users with a flexible way to tag and retrieve data by applying a faceted search paradigm to its vast data storage. This makes it easy to create a taxonomy that reflects the organisation's own view of the data universe across data subscriptions and promotes internal semantic alignment.

What data sources does Artesian support?

Artesian allows each client to connect to a wide range of data sources relevant to energy market practitioners across Europe and supports publicly available data sources as well as those subject to license in addition to providing a write API to accommodate each client's own data. Subscription and compliance with license restrictions for each desired data source is the responsibility of the client.

Once the desired sources have been selected and credentials for access to the ones subject to authentication have been supplied by the client, it is possible to instantiate an Artesian instance. Upon instantiation, Artesian can, optionally, automatically populate itself with all historical data made available by each provider and will keep itself up to date thereafter. Artesian is not a multi-tenant system, thus each client is assigned their own private instance in order to physically ensure complete data segregation and privacy.

The following list of data sources is supported as of the time of the last update of this white paper (June 21, 2021) but it is constantly being extended.

Area	Data Provider	URL
Capacity Management	Damas	https://damas.terna.it/
Capacity Management	Joint Allocation Office (ex CASC)	http://www.jao.eu/main
Capacity Management	SEECAO	http://www.seecao.com
Capacity Management	Nordpool	http://www.nordpoolspot.com/
Commercial Data Providers	Alba	https://www.albasoluzioni.com/

Commercial Data Providers	AlbaFlows	https://www.albasoluzioni.com/
Commercial Data Providers	Genscape	http://www.genscape.com/
Commercial Data Providers	Meteogroup	http://www.meteogroup.com/
Commercial Data Providers	Meteologica	http://www.meteologica.com/
Commercial Data Providers	Volume (ex MK/Wattsight)	https://volueinsight.com/
Commercial Data Providers	PointConnect	https://www.refinitiv.com/en
Commercial Data Providers	Renen	http://renen.com
Market	BelPex	https://www.belpex.be/
Market	EirGrid	http://www.eirgridgroup.com/
Market	EirGridGroup	http://www.eirgridgroup.com/
Market	EPEX	https://www.epexspot.com/en/
Market	EXAA	https://www.epexspot.com/en/
Market	Gestore Mercati Energetici (GME)	http://www.mercatoelettrico.org/
Market	GMEWeb	http://www.mercatoelettrico.org/It/Tools/Accessodati.aspx
Market	Interconnector	http://www.interconnector.com/operational-data/daily-summary/
Market	LAGIE Operator of Electricity Market	https://www.dapeep.gr/
Market	OmieEs	http://www.omie.es/inicio
Market	RED.ELECTRA	http://www.ree.es/en
Market	MibGas	http://www.mibgas.es/mercados-de-gas/
Market	Single Electricity Market Operator	http://www.sem-o.com/Pages/default.aspx
Market	Southpool Regional Exchange (Slovenia)	http://www.bsp-southpool.com/
Market	Trayport Data Factory and stream persister (TDF)	http://www.trayport.com
Market	TrayportAssessmentEngine (requires TDF)	http://www.trayport.com
Market	TrayportBidAskEngine (requires TDF)	http://www.trayport.com

Market	CEGH Gas Exchange	http://www.cegh.at/
Market	EEX	https://www.eex.com/en/
REMIT Transparency	Gassco	http://flow.gassco.no/
REMIT Transparency	GMEPip	https://pip.ipex.it/PipWa/Front/#/PowerUmms
TSO/Infrastructure Operator	ENAGAS	http://www.enagas.es
TSO/Infrastructure Operator	ENTSO-E	https://www.entsoe.eu/
TSO/Infrastructure Operator	ENTSO-G	http://www.entsog.eu/
TSO/Infrastructure Operator	Societa' gasdotti Italia	http://www.gasdottitalia.it/it
TSO/Infrastructure Operator	GIETransparency	http://www.gie.eu/
TSO/Infrastructure Operator	LNG	https://alsi.gie.eu/#/
TSO/Infrastructure Operator	RTE	http://www.rte-france.com/
TSO/Infrastructure Operator	SNAM Rete Gas	http://www.snamretegas.it/
TSO/Infrastructure Operator	Swissgrid	https://www.swissgrid.ch
TSO/Infrastructure Operator	Terna	http://www.terna.it/
TSO/Infrastructure Operator	Tso.Germany	http://www.50hertz.com/de/ http://amprion.de/ http://www.tennet.eu/l https://www.transnetbw.com/
TSO/Infrastructure Operator	TernaSunSet	https://myterna.terna.it/SunSet/Public/
TSO/Infrastructure Operator	Admie	https://www.admie.gr/
Weather Data	AerisWeather	https://www.meteomatics.com/en/weather-api
Weather Data	DarkSky	https://darksky.net/dev/
Weather Data	IBMWeather	https://www.ibm.com/weather
Weather Data	MetDesk	http://www.metdesk.com/
Weather Data	METeorologistikInstitutt	https://api.met.no/weatherapi/
Weather Data	IlMeteo.it	https://www.ilmeteo.it/

What types of data can I manage?

Artesian has been designed to manage the following types of data:

Time Series: a time series in Artesian is a sequence of data points indexed by time coordinates where each data point is applicable to a specific time interval starting at its time-coordinate index. At present a time series does not need to be contiguous, meaning that not all time intervals within it need to have an assigned value, but it needs to be homogenous, i.e. each value needs to apply to a time interval of the same duration. In the rest of this document we refer to the time interval applicable to each value in a time series as *time series granularity*. Examples of this type of data are:

- The time series of hourly power prices generated by a specific spot market, say the GME in Italy;

- The time series illustrating how much gas has been consumed by a power plant on a daily basis for its production activities;

Versioned Time Series: a versioned time series is itself a time series, additionally indexed by the time coordinates of its version. Examples of this type of data are:

- *Meteorological forecasts* for a specific geographical area for a specific date. Multiple versions of the same forecasts can be received until the date of the forecast is effectively reached. Each version may differ from the previous one and typically will increase in accuracy as the date itself is approached.

- *Forward curves* derived from a market term structure and utilised for mark to market calculations for a set of energy commodity derivative transactions. Typically, multiple versions of a forward curve will be available, each one pertaining to a specific trade date and generated based on data known at a specific point in time.

Auctions: an auction in Artesian is a sequence of data points indexed by time and clustered by side and price. An auction does not need to be contiguous, meaning that not all time intervals within it need to have an assigned value and does not need to be homogenous, i.e: the price cluster can be increased by a non constant amount.

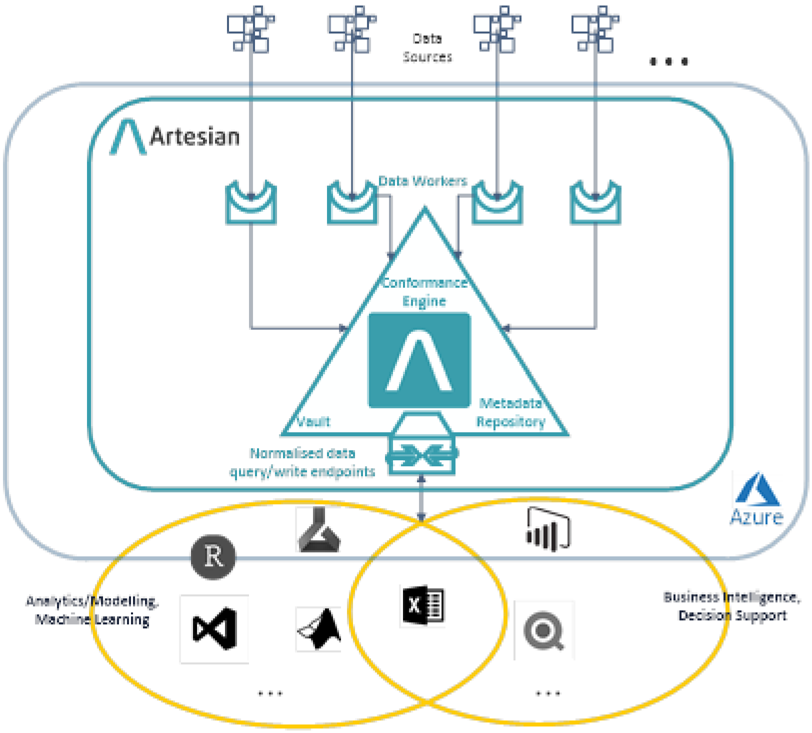
GME Public Offers (Offerte Pubbliche): the Italian power spot market (GME) can be considered a special case for Auction data, as Artesian will also store and make available, as a sequence of data points indexed by time, all individual offers (Offerte Pubbliche) underpinning the computation of the system marginal price for each market session.

Market Assessment: a market assessment is a time series of 8-tuples indexed by market product. The 8-tuple consists of the following data items: Settlement, High, Low, Open, Close, VolumePaid, VolumeGiven, VolumeTotal and will pertain to a specific product, e.g. The Aug-2021 contract for French base power. Market assessments can be received directly from data providers, or computed dynamically by the system at a specified frequency based on a stream of market offers.

BidAsk: a BidAsk is a time series of 6-tuples indexed by market product. The 6-tuple consists of the following data items: BestBidPrice, BestAskPrice, BestBidQuantity, BestAskQuantity, LastPrice, LastQuantity and will pertain to a specific product, e.g. The Aug-2017 contract for French base power.

Product Architecture

Artesian stands as an aggregator of information from multiple providers. It downloads data distributed by public and licensed providers through different channels and conforms it to common formats and structures. This in turn simplifies the retrieval process into analytical tools or even trading models.


Data Sources
• • •
Artesian
Data Workers
Conformance Engine
Vault
Metadata Repository
Normalised data query/write endpoints
Azure
R
Analytics/Modelling, Machine Learning
Business Intelligence, Decision Support
...
...

Key components of the architecture are:

- Data workers,
- Conformance engine, vault and metadata repository,
- Normalised data query endpoints.

Data Workers

Data workers are resilient components responsible for collecting raw data from the chosen data providers. They can typically handle the following sources:

- FTP providers (with and without authentication)
- Websites that provide data in their pages (with and without authentication)
- Websites that allow you to download files in multiple formats (xml, csv, xls, xlsx, zip etc…)
- Applications and websites that offer custom APIs to access their data.

Conformance engine, Vault and Metadata Repository

“*Conformance engine*” is the short name used internally in ARK to refer to the architecture component that is responsible for data conformance across heterogeneous data feeds and appropriate up and down sampling of time series granularity.

The conformance engine is the heart of Artesian. Its unique feature is the ability to

conform any parsed object received from the data workers, greedily generate all missing granularities and write information to the vault and the metadata repository.
The conformance engine is hosted in an Azure cloud environment and is able to dynamically scale its performance depending on the number of read and write requests received by the system.

An Azure bus service is utilised as a channel to send write requests to the conformance engine. Every message has a retry policy and is put in a 'wait' state in case the conformance engine goes offline. This ensures that every write request is executed and never lost. Key components of the conformance engine are:

- *Azure service bus*: handles all write requests. It retries failed requests and waits if the system is unavailable. This guarantees that every request is processed and no data is lost.
- *Conformance engine*: conforms received data into one of three forms:
- Time Series (TimeSeriesId, Date/Time, Value)
- Versioned Time Series (TimeSeriesId, TimeSeriesVersion, DateTime, Value)
- Market Assessment (MASId, ReportTime, Open, Close, High, Low, Settlement, VolumePaid, VolumeGiven, VolumeTotal)
- *Granularity extender*: starting from data in the form of a Time Series or Versioned Time Series, the granularity extender is able to generate all missing granularities. Raw data is always provided in a single granularity (e.g. a daily Time Series) and this module can pre-compute data in other granularities by up-sampling and down-sampling the Time Series. Supported granularities are: 10min, 15min, 30min, Hour, Day, Week, Month, Quarter, Season and Year. Out of the box the system supports simple up-sampling (via averages or sums) and down sampling (through value replication or division) algorithms but more complex algorithms can be developed on request.
- *Metadata Repository*: a SQL database where all metadata is stored.
- *Vault*: a custom storage component that contains *txt* files for Time Series, Versioned Time Series and Market Assessments.

The V*ault* is a cloud based system that stores *binary blobs*. It leverages Azure Blob Storage technology allowing storage up to 500TB. Each data item from each data provider is saved in a different binary blob file and the system is read optimised by being able to read the correct chunk of data within each file. Internally, interaction with the vault is achieved through the following components:

- Vault Writer: this component is able to write metadata associated with a specific data provider to the metadata repository separating it from the data itself, which is instead written to the vault. In order to do this the Vault Writer implements a custom storage driver able to save all data as *doubles* in *txt* files.

- Vault Reader: this component is able to read data from the metadata repository and from the vault. It implements a custom storage driver that is able to optimise reading from the saved *txt* files in the vault and extract multiple time series at different granularities. This approach allows the system to guarantee fast read performance even when many data items are being requested.

To give an example of the amount of storage utilised by the system, an hourly times series over a year will contain 8760 individual values that, when saved as 8-byte double precision numbers, will occupy roughly 70KB. Taking this as a benchmark the product is able to store almost 7 billion hourly year long time series while retaining the ability to write and read data efficiently. Market assessments are saved in the vault using Azure Table Storage Technology which allows handling more complicated data structures.

Both technologies support the redundancy mechanisms provided by the Azure platform.

The metadata repository is an Azure SQL database that stores all metadata related to the items saved in the vault.

Normalised REST data query endpoints

Data can be accessed through a simple web interface and viewed from office automation platforms such as Microsoft Excel or accessed programmatically via popular analytical or development tools (e.g. Matlab, Visual Studio). The interface is composed of:

- Endpoints to navigate the metadata
- WebApi controllers to retrieve the data.

To download the actual data the product provides the following endpoints:

- *Time Series Controller*: to submit queries on Time Series and Versioned Time Series.
- *Market Assessment Controller*: to submit queries on Market Assessments.
- *BidAsk Controller:* to submit queries on BidAsk.
- *Auction Controller*: to submit queries on Auction.
- *GME Public Offers Controller*: to submit queries on GME Public Offers.

The following queries are possible using the *Time Series Controller*:

```
ts/{granularity}/{start}/{end}?{tsids}
ts/{granularity}/{interval}?{tsids}
ts/{granularity}/{period}?{tsids}
ts/{granularity}/{periodFrom}/{periodTo}?{tsids}

vts/LastOfDays/{vstart}/{vend}/{granularity}/{start}/{end}?{tsids}
vts/LastOfDays/{vPeriod}/{granularity}/{start}/{end}?{tsids}
vts/LastOfDays/{vPeriodFrom}/{vPeriodTo}/{granularity}/{start}/{end}?{tsids}
vts/LastOfDays/{vstart}/{vend}/{granularity}/{interval}?{tsids}
vts/LastOfDays/{vPeriod}/{granularity}/{interval}?{tsids}
vts/LastOfDays/{vPeriodFrom}/{vPeriodTo}/{granularity}/{interval}?{tsids}
vts/LastOfDays/{vstart}/{vend}/{granularity}/{period}?{tsids}
vts/LastOfDays/{vPeriod}/{granularity}/{period}?{tsids}
vts/LastOfDays/{vPeriodFrom}/{vPeriodTo}/{granularity}/{period}?{tsids}
vts/LastOfDays/{vstart}/{vend}/{granularity}/{periodFrom}/{periodTo}?{tsids}
vts/LastOfDays/{vPeriod}/{granularity}/{periodFrom}/{periodTo}?{tsids}
vts/LastOfDays/{vPeriodFrom}/{vPeriodTo}/{granularity}/{periodFrom}/{periodTo}?{tsids}

vts/LastOfMonths/{vstart}/{vend}/{granularity}/{start}/{end}?{tsids}
vts/LastOfMonths/{vPeriod}/{granularity}/{start}/{end}?{tsids}
vts/LastOfMonths/{vPeriodFrom}/{vPeriodTo}/{granularity}/{start}/{end}?{tsids}
vts/LastOfMonths/{vstart}/{vend}/{granularity}/{interval}?{tsids}
vts/LastOfMonths/{vPeriod}/{granularity}/{interval}?{tsids}
vts/LastOfMonths/{vPeriodFrom}/{vPeriodTo}/{granularity}/{interval}?{tsids}
vts/LastOfMonths/{vstart}/{vend}/{granularity}/{period}?{tsids}
vts/LastOfMonths/{vPeriodFrom}/{vPeriodTo}/{granularity}/{periodFrom}/{periodTo}?{tsids}
vts/LastOfMonths/{vPeriodFrom}/{vPeriodTo}/{granularity}/{period}?{tsids}
vts/LastOfMonths/{vstart}/{vend}/{granularity}/{periodFrom}/{periodTo}?{tsids}
vts/LastOfMonths/{vPeriod}/{granularity}/{periodFrom}/{periodTo}?{tsids}
vts/LastOfMonths/{vPeriodFrom}/{vPeriodTo}/{granularity}/{periodFrom}/{periodTo}?{tsids}

vts/Last{n}/{granularity}/{start}/{end}?{tsids}
```

```
vts/Last{n}/{granularity}/{interval}?{tsids}
vts/Last{n}/{granularity}/{period}?{tsids}
vts/Last{n}/{granularity}/{periodFrom}/{periodTo}?{tsids}

vts/MostRecent/{granularity}/{start}/{end}?{tsids}
vts/MostRecent/{vstart}/{vend}/{granularity}/{start}/{end}?{tsids}
vts/MostRecent/{vPeriod}/{granularity}/{start}/{end}?{tsids}
vts/MostRecent/{vPeriodFrom}/{vPeriodTo}/{granularity}/{start}/{end}?{tsids}
vts/MostRecent/{granularity}/{interval}?{tsids}
vts/MostRecent/{vstart}/{vend}/{granularity}/{interval}?{tsids}
vts/MostRecent/{vPeriod}/{granularity}/{interval}?{tsids}
vts/MostRecent/{vPeriodFrom}/{vPeriodTo}/{granularity}/{interval}?{tsids}
vts/MostRecent/{granularity}/{period}?{tsids}
vts/MostRecent/{vstart}/{vend}/{granularity}/{period}?{tsids}
vts/MostRecent/{vPeriodFrom}/{vPeriodTo}/{granularity}/{periodFrom}/{periodTo}?{tsids}
vts/MostRecent/{vPeriodFrom}/{vPeriodTo}/{granularity}/{period}?{tsids}
vts/MostRecent/{granularity}/{periodFrom}/{periodTo}?{tsids}
vts/MostRecent/{vstart}/{vend}/{granularity}/{periodFrom}/{periodTo}?{tsids}
vts/MostRecent/{vPeriod}/{granularity}/{periodFrom}/{periodTo}?{tsids}
vts/MostRecent/{vPeriodFrom}/{vPeriodTo}/{granularity}/{periodFrom}/{periodTo}?{tsids}

vts/MUV/{granularity}/{start}/{end}?{tsids}
vts/MUV/{granularity}/{interval}?{tsids}
vts/MUV/{granularity}/{period}?{tsids}
vts/MUV/{granularity}/{periodFrom}/{periodTo}?{tsids}

vts/Version/{version}/{granularity}/{start}/{end}?{tsids}
vts/Version/{version}/{granularity}/{interval}?{tsids}
vts/Version/{version}/{granularity}/{period}?{tsids}
vts/Version/{version}/{granularity}/{periodFrom}/{periodTo}?{tsids}
```

where

{granularity}: the granularity of extraction
{start}: start date for the data range

{end}: end date for the data range

{interval}: the relative interval of the data range out of:

- RollingWeek
- RollingMonth
- RollingQuarter
- RollingYear
- WeekToDate
- MonthToDate
- QuarterToDate
- YearToDate

{period}/{periodFrom}/{periodTo}: the relative Duration of the data range to be extracted defined accordingly to the ISO 8601 standard (ref: https://en.wikipedia.org/wiki/ISO_8601#Durations)

{tsids}: list of TimeSeries Ids separated by commas

{vstart}: start of the version range

{vend}: end of the version range

{version}: specific timestamp of the version of interest

{vperiod}/{vperiodFrom}/{vperiodTo}: the relative Duration of the version date range to be extracted defined accordingly to the ISO 8601 standard (ref: https://en.wikipedia.org/wiki/ISO_8601#Durations)

LastOfDays: returns the newest available version of the day

LastOfMonths: returns the newest available version of the month

MUV: returns the Most Up to date Version

Last{n}: returns the last n versions

MostRecent: returns the most recent version available in the version period defined

SingleVersion: returns the specified version

The following queries are possible through the *Market Assessment Controller*:

mas/{start}/{end}?{masids},{tenors}

mas/{interval}?{masids},{tenors}

mas/{period}?{masids},{tenors}

mas/{periodFrom}/{periodTo}?{masids},{tenors} where:

{start}: start date for the Report Date range
{end}: end date for the Report Date range
{masids}: list of Market Assessment Ids separated by commas
{tenors}: list of tenors (relative or absolute) separated by commas that are to be extracted
{interval}: the relative interval of the Report Date range out:

- RollingWeek
- RollingMonth
- RollingQuarter
- RollingYear
- WeekToDate
- MonthToDate
- QuarterToDate
- YearToDate

{period}/{periodFrom}/{periodTo}: the relative Duration of the data range to be extracted defined accordingly to the ISO 8601 standard (ref: https://en.wikipedia.org/wiki/ISO_8601#Durations)

The following queries are possible through the *BidAsk Controller*:

ba/{start}/{end}?{baids},{tenors}
ba/{interval}?{baids},{tenors}
ba/{period}?{baids},{tenors}
ba/{periodFrom}/{periodTo}?{baids},{tenors}

where:

{start}: start date for the Report Date range
{end}: end date for the Report Date range
{baids}: list of BidAsk Ids separated by commas
{tenors}: list of tenors (relative or absolute) separated by commas that are to be extracted
{interval}: the relative interval of the Report Date range out:

- RollingWeek
- RollingMonth

- RollingQuarter
- RollingYear
- WeekToDate
- MonthToDate
- QuarterToDate
- YearToDate

{period}/{periodFrom}/{periodTo}: the relative Duration of the data range to be extracted defined accordingly to the ISO 8601 standard (ref: https://en.wikipedia.org/wiki/ISO_8601#Durations)

The following queries are possible through the *Auctions Controller*:

auction/{start}/{end}?{auctionids}
auction/{period}?{auctionids}
auction/{periodFrom}/{periodTo}?{auctionids}

where:

{start}: start date for the Report Date range
{end}: end date for the Report Date range
{auctionids}: list of Auctions Ids separated by commas
{period}/{periodFrom}/{periodTo}: the relative Duration of the data range to be extracted defined accordingly to the ISO 8601 standard (ref: https://en.wikipedia.org/wiki/ISO_8601#Durations)

The following queries are possible through the *GME Public Offers Controller*:

gmepublicoffer/v1.0/extract/{date}/{purpose}/{status}

where:

{date}: date extracted
{purpose}: the data extracted should be relative to the Bid or Offer
{status}: the extracted data should be in one of the following status: ACC, REJ, INC, REP, REV, SUB

Data write endpoints

A single endpoint supports the writing of data to Artesian. Values are passed according to a common payload for all the data types described above. The common payload implies that only the fields appropriate to the data type being written will need to be populated.

The payload has the following fields:

- ID: is the unique identifier of the data that we're going to write, and is composed of two fields:
    - Provider: name of the data provider
    - Name: name of the item that is going to be written
- Version: timestamp of the version, to be populated only for a VTS
- Timezone: Identifier of the Time Zone. Only UTC supported if the data is written is at hourly or lower granularity
- DownloadedAt: the UTC timestamp of the acquisition/generation of the data being written
- DeferCommendExecution: specifies whether execution will be asynchronous or not
- DeferDataGeneration: specifies if the MUV computation will be asynchronous or not
- KeepNulls: specifies if NULL values should be saved or skipped
- Rows: to be used when writing TS/VTS data. Supports the following fields:
    - key: time coordinate of the value
    - value: value associated with the time coordinate
- MarketAssessment: to be used when writing MAS data. Fields to populate are:
    - key: time coordinate of the value
    - value: value associated with the time coordinate. Fields to populate are:
        - Settlement
        - Open
        - Close
        - High
        - Low
        - VolumPaid
        - VolumGiven
        - Volume

- BidAsk: to be used when writing BidAsk data. Fields to populate are:
    - key: time coordinate of the value
    - value: value associated with the time coordinate. Fields to populate are:

        - BestBidPrice
        - BestAskPrice
        - BestBidQuantity
        - BestAskQuantity
        - LastPrice
        - LastQuantity
- AuctionRows: to be used in case of Auctions writing. Fields to populate are:
    - BidTimestamp: time coordinate of the value
    - Bid: value associated with the time coordinate. The available fields are:
        - Price
        - Quantity

Instantiating an Artesian warehouse

Artesian can be connected to a range of data sources, both public, and licensed. Clients are responsible for licensing the data sources they require and complying with any licensing restrictions. Once the set of required data sources have been selected an Artesian instance can be prepared for the client. The process can be broken down into three easy steps:

1. Discuss requirements with an ARK consultant who will help to identify the required data workers and establish if new ones should be developed.
2. Provide credentials for connectivity to each data source (where applicable) – the client is responsible for holding appropriate licenses to the data sources for the intended utilisation.
3. ARK enables the data workers and monitors them for you. Your personal Artesian instance optionally populates itself with any historical data available at the point of instantiation and keeps itself up to date thereafter.

Is Artesian secure?

The only public point of access to the system is the web interface. All access to the REST endpoints and the WebApi controllers is secured through state of the art security technologies offered by Microsoft: OAuth2 with bearer token authentication integrated with the Azure Active Directory (Azure AD).

Azure AD uses public key cryptography to sign tokens and verify their validity. Bearer tokens do not have a built-in mechanism to prevent unauthorised parties from using them and are therefore always transmitted over a secure transport layer. Azure AD is the identity provider adopted, responsible for issuing security tokens upon successful authentication of users and applications.

The product's security layer exploits Azure Active Directory Authentication Libraries, the same used by Microsoft for access to their online web resources (e.g. Office 365, Azure Portal).

Closing Remarks

Artesian is a rapidly evolving platform and chances are that this white paper, albeit still correct, is probably already out of date and incomplete. In order to discover any new and upcoming features of the platform please contact us by email on *enquiries@ark-energy.eu*.

About Us

Artesian is an Ark product. Artesian was developed in-house with the input of the best energy traders and data scientists.
We are a technology company working within the energy sector: we are specialists in a highly competitive industry. Our aim is simple: to create accessible, reliable software for our clients. Our technology is based in the cloud, and our offices are in Ireland and Italy. We're a young organisation priding ourselves on our combination of business and IT knowledge, coupled with experience and drive, giving us a competitive edge and the ability to deliver Software as a Service (SaaS) solutions capable of substantially reducing the total cost of ownership for our clients.

For more information see: www.ark-energy.eu