Analytics Introduction

Contents

This will be a brief introduction to the analytics option of InferStack.

Overview

The analytics option is available for purchase for most InferStack products. It includes the Energy, KPI, Site Spark and Spark apps and the Energy Star, Green Button, Historian Analytics Kit, HVAC and Lighting extensions.

The primary goal of the analytics option is to automate the process of finding "issues of interest" in the data. We use rules to define the functions which scan the data looking for issues. When a rule finds a "hit", it generates a spark. Instead of paging through reports or charts, you can use rules and sparks to find exactly the issues you care about.

Setting Goals

The Analytics option can be used in many ways including:

- database for real-time and historized sensor data
- crunching sensor data and applying data transformations
- analytics of sensor data
- visualization of above

It is well suited to a range of analytic applications:

- portfolio level analysis: performance of a portfolio of buildings against energy and other KPI's
- systems level analytics: relationships between systems across different weather or load conditions
- equipment level analytics: identifying faults in the operation of specific pieces of equipment

Where it really shines is its ability to analyze time-series data from sensors and control systems to find what matters to you and your project. Once you have modeled your system and imported the data, it provides a suite of tools to write functions to query the data and apply data transformations such as rollups and normalizations. One of the most powerful features is the ability to crunch through your data looking for conditions which matter most, such as equipment failures or non-optimal operation.

So before you begin you should consider:

- what kinds of performance issues, patterns or equipment failure conditions are you interested in analytics finding for you?
- what data do you have available?
- what kinds of queries and data transformations are you interested in?
- what kinds of data visualizations are you interested in?

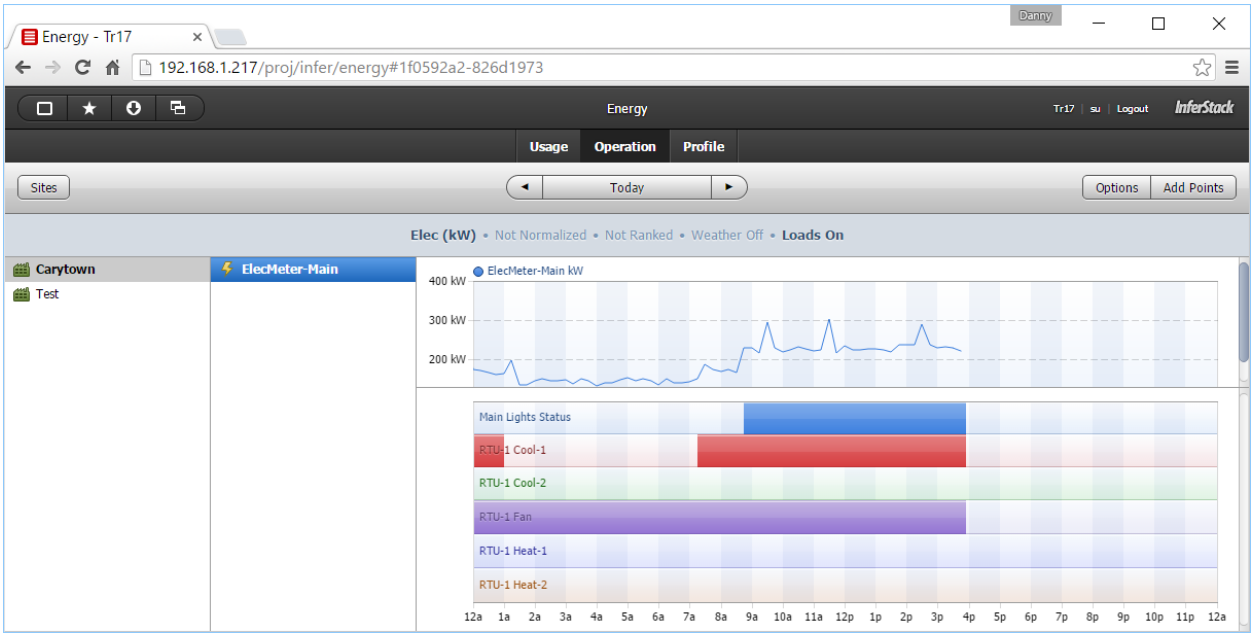
Remember that analytics is often an exploratory or cyclical process. Your initial data and analytics will likely lead to new questions and ideas which in turn might call for new sensor instrumentation and new analytic rules.

Energy

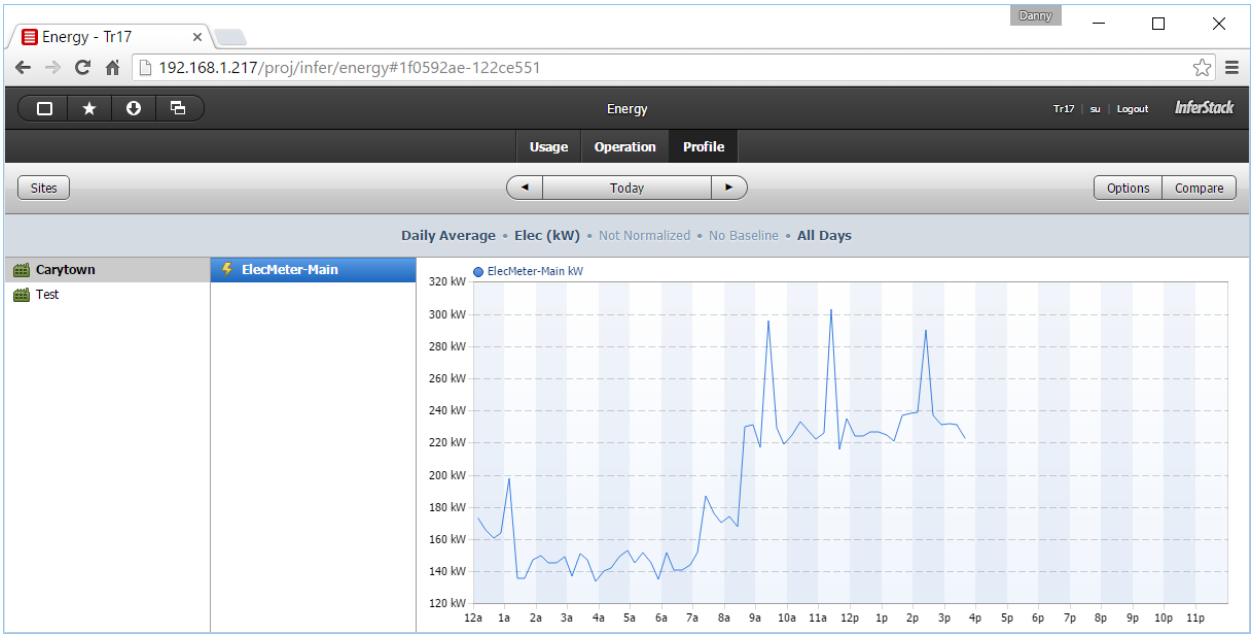
The Energy app provides usage, operation and profile information for your sites. You can select Sites, Dates and Options to analyze your energy usage, operation and profile.

Usage

Operation



Profile



Energy Star

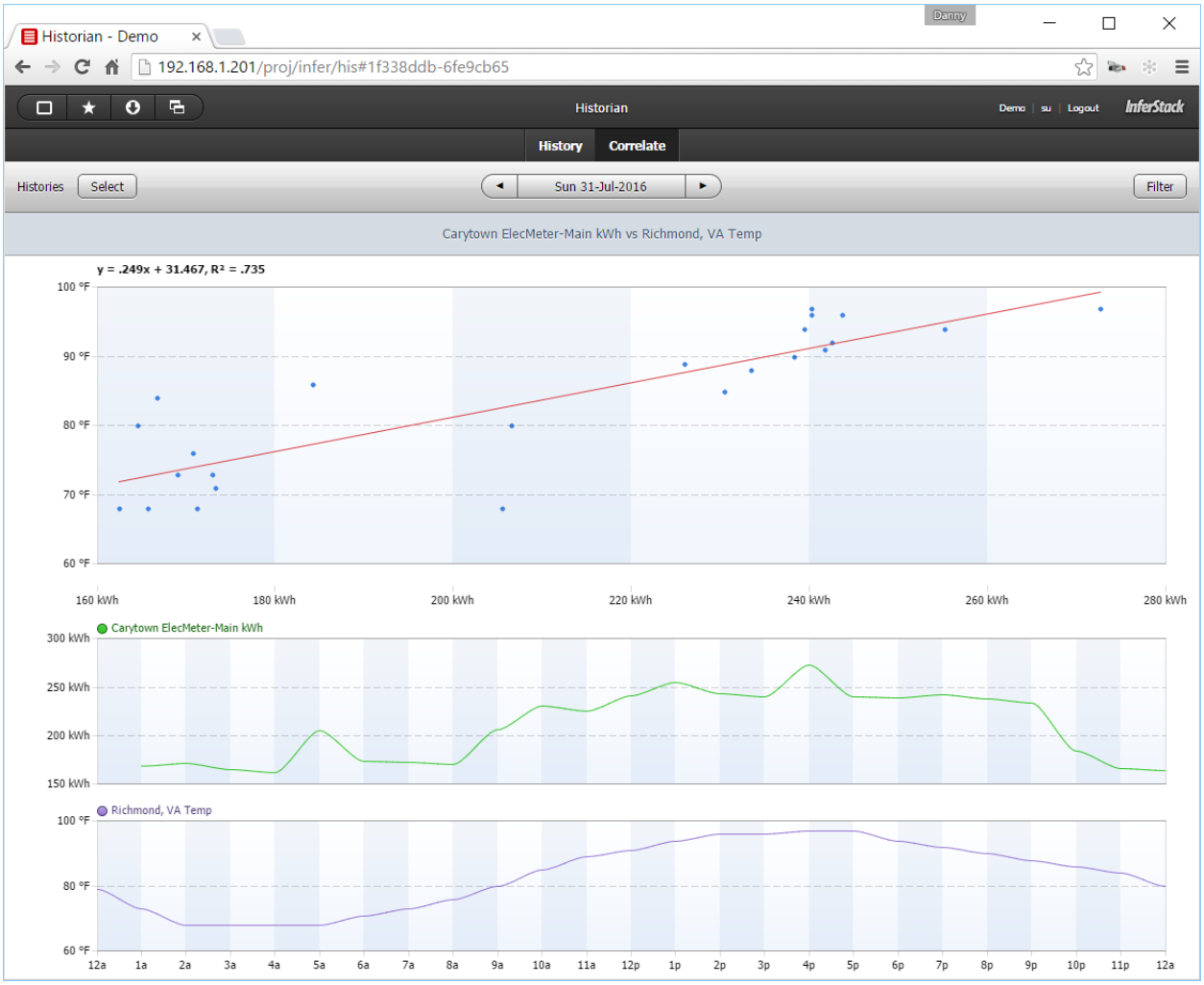
The Energy Star extension implements a client interface to the Porfolio Manager web service. Features include:

- An Energy Star connector maps to an account and is used to manage your sites, meters, usage
- Ability to manage your Energy Star properties and map them to SkySpark sites
- Ability to manage your Energy Star meters and map them to SkySpark meter points
- Ability to manage your Energy Star meter usage data
- Push history data from SkySpark to Energy Star

See http://licensing.intellastar.com/doc/ext-energyStar/doc#overview for more information.

Historian Correlate

The Historian app provides a Correlate tab when analytics has been purchased to look at correlation between two points.



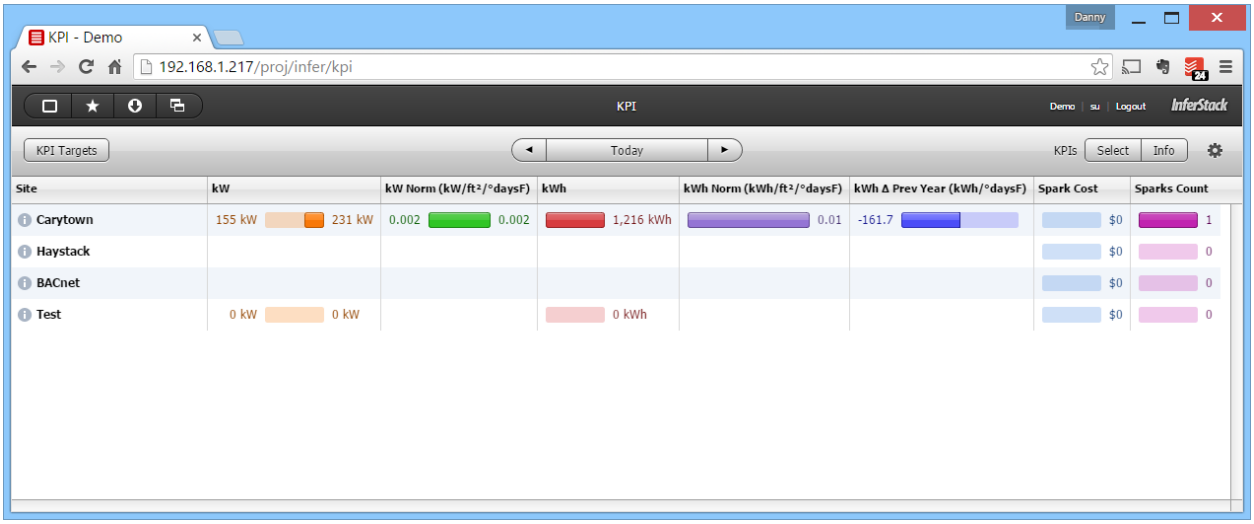
KPI

The KPI app is used to compute key performance indicators using Axon functions.

We use the following terms to discuss the KPI design:

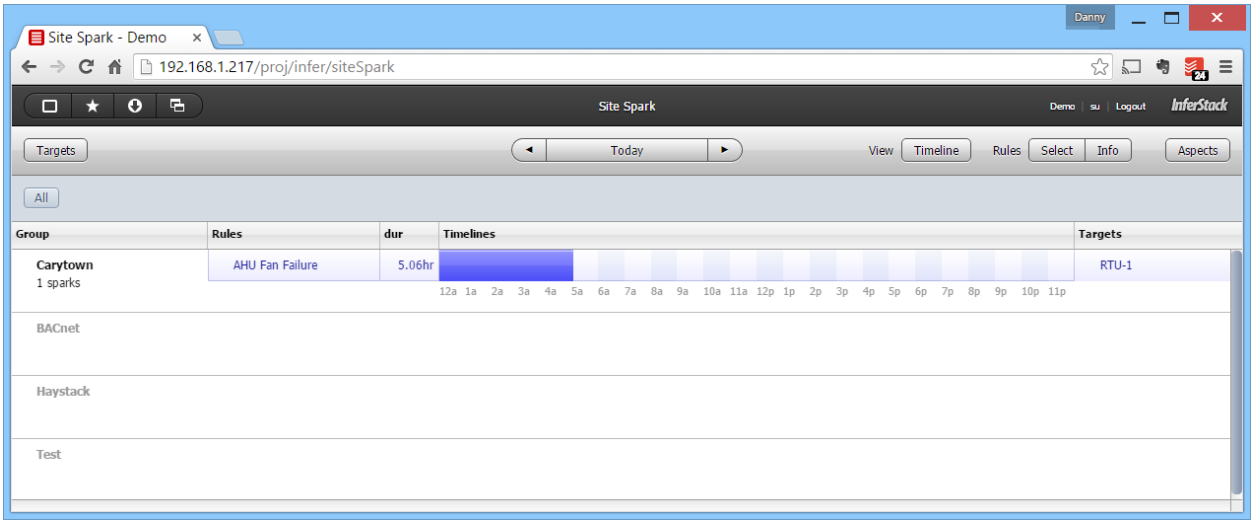
- **KPI Rec**: this is the record in the database that describes the KPI: dis, func, target filter, and help
- **KPI Func**: this is the function that takes a target and a date and range and computes a KPI Result
- **KPI Result**: a Dict that provides the summary information computed for a given target and date range

KPIs are measured on a target over a variable range of time. For example total kWh consumption of a site over a range of time such as yesterday, last month, this year, etc.



Site Spark

The Site Spark app is used to navigate and visualize the sparks found in your sites and equipment. You can filter by date or date range, rules and targets.



Spark

The Spark App is used to create and manage rules including:

- viewing sparks in tabular format
- creating new rules
- enabling and disabling rules
- debugging rules

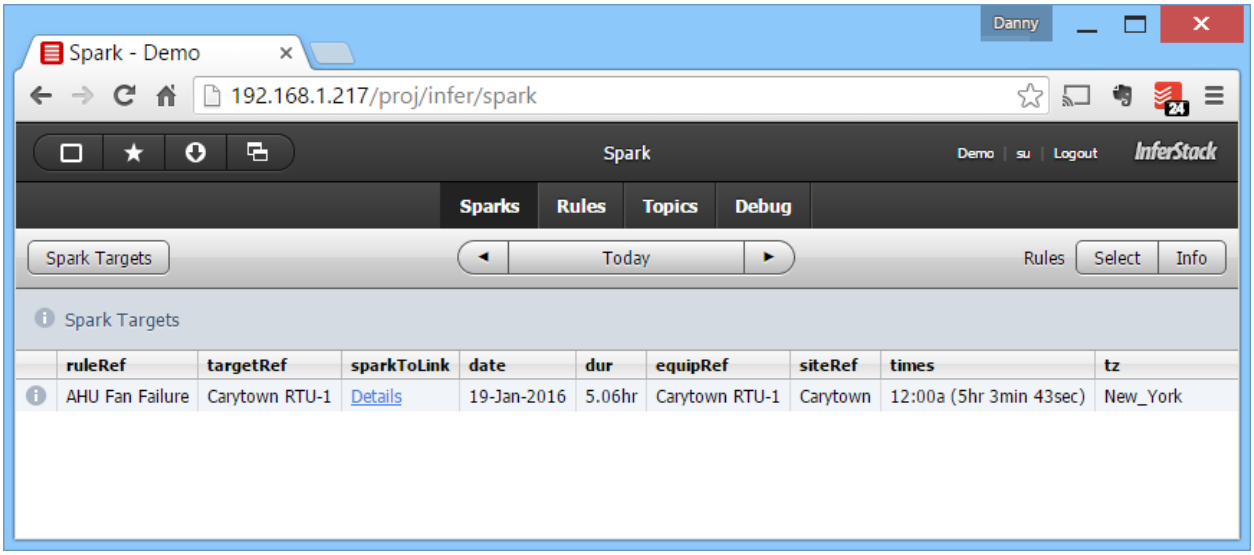
Sparks

Sparks are represented as a dict with an arbitrary set of tags (name/value pairs). All sparks have the following tags:

- spark: marker tag

- date: Date for the rule hit
- ruleRef: Ref for the generating rule
- targetRef: Ref for the target the rule ran against
- periods: encoded timeline of the sparks during the date
- tz: timezone to use to convert date into timestamps
- any other tags generated by the rule function

Note that sparks are **not** records in the folio database. They are computed and cached as needed by the spark rule engine.



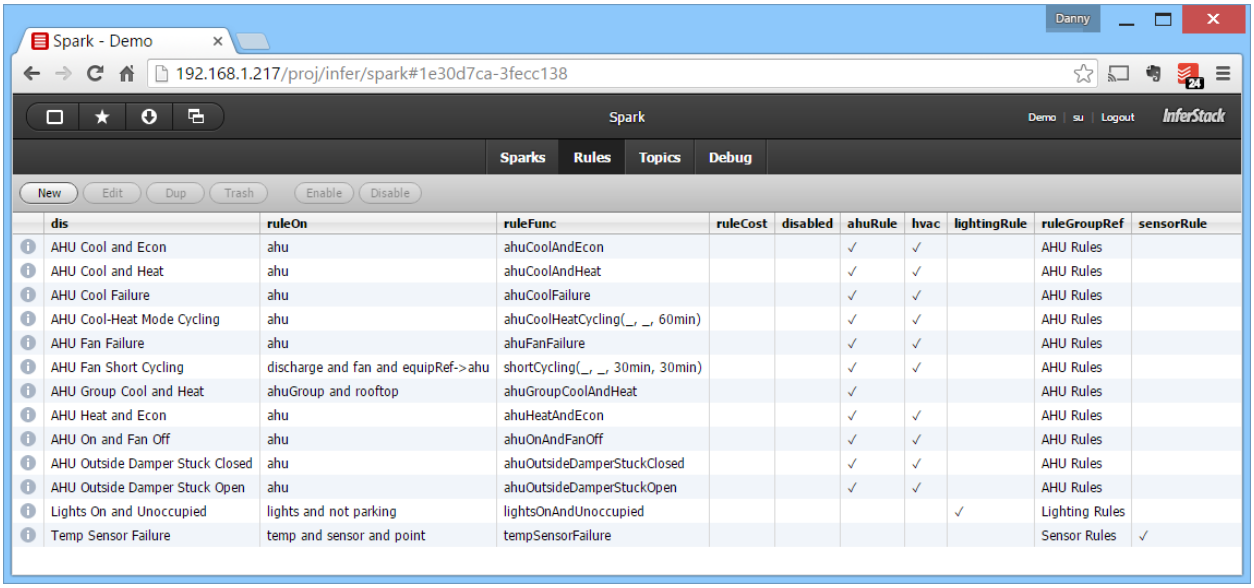
Rules

A rule is a record in folio with the following tags:

- rule: marker tag
- dis: all rules should define a nice display name
- help: rules should define documentation for end users
- ruleOn: filter which specifies targets of rule
- ruleFunc: Axon expression which evaluates to function which computes sparks
- disabled: marker tag to temporarily disable the rule from running
- color: optional tag to assign a color to visualizations; this should be one of the predefined chart colors
- ruleCost: cost formula

All rules are configured *on* some target filter. For example a rule might be configured to run only on AHUs via the "ahu" filter. Or maybe just on RTUs via the filter "ahu and rooftop". You can organize your rules against targets however it best fits your project.

You can create a `ruleCost` tag on a rule to have the spark engine automatically compute a `cost` tag on each spark for that rule

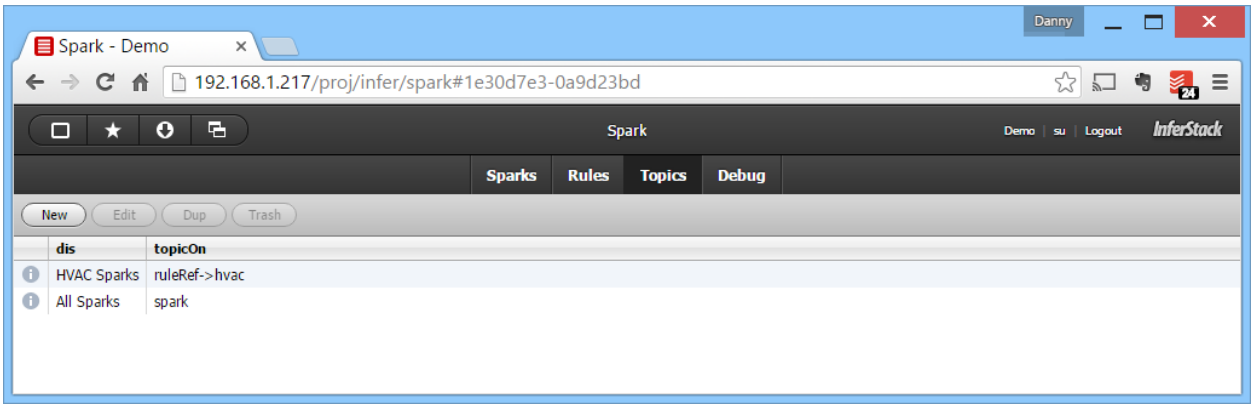


dis	ruleOn	ruleFunc	ruleCost	disabled	ahuRule	hvac	lightingRule	ruleGroupRef	sensorRule
AHU Cool and Econ	ahu	ahuCoolAndEcon			✓	✓		AHU Rules	
AHU Cool and Heat	ahu	ahuCoolAndHeat			✓	✓		AHU Rules	
AHU Cool Failure	ahu	ahuCoolFailure			✓	✓		AHU Rules	
AHU Cool-Heat Mode Cycling	ahu	ahuCoolHeatCycling(_, _, 60min)			✓	✓		AHU Rules	
AHU Fan Failure	ahu	ahuFanFailure			✓	✓		AHU Rules	
AHU Fan Short Cycling	discharge and fan and equipRef->ahu	shortCycling(_, _, 30min, 30min)			✓	✓		AHU Rules	
AHU Group Cool and Heat	ahuGroup and rooftop	ahuGroupCoolAndHeat			✓			AHU Rules	
AHU Heat and Econ	ahu	ahuHeatAndEcon			✓	✓		AHU Rules	
AHU On and Fan Off	ahu	ahuOnAndFanOff			✓	✓		AHU Rules	
AHU Outside Damper Stuck Closed	ahu	ahuOutsideDamperStuckClosed			✓	✓		AHU Rules	
AHU Outside Damper Stuck Open	ahu	ahuOutsideDamperStuckOpen			✓	✓		AHU Rules	
Lights On and Unoccupied	lights and not parking	lightsOnAndUnoccupied					✓	Lighting Rules	
Temp Sensor Failure	temp and sensor and point	tempSensorFailure						Sensor Rules	✓

Topics

Topics are a way to organize your sparks into categories just like alarms and notes. Topics are the primary way to manage user subscriptions. A topic is a simple record with the following tags:

- dis: always give your topics a nice display name
- topic: marker tag indicating a general topic
- sparkTopic: marker tag indicating specifically a spark topic
- topicOn: a filter string for matching sparks



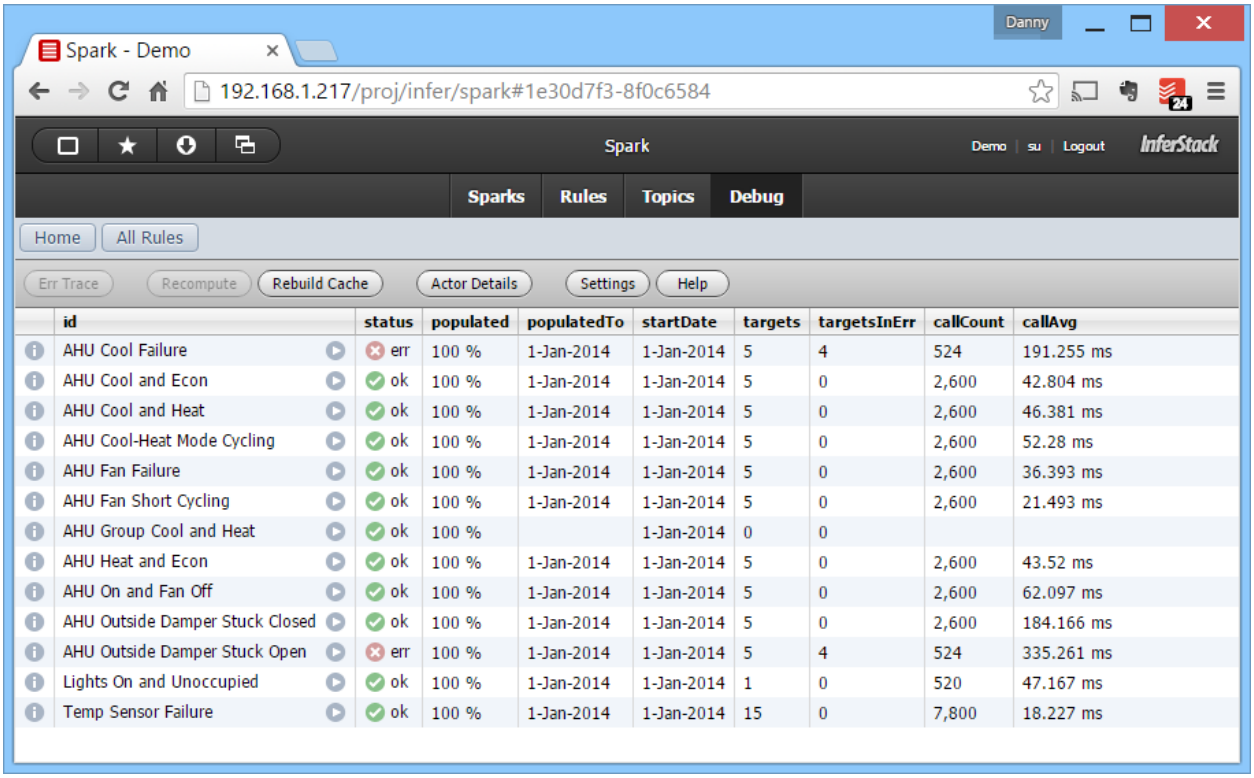
dis	topicOn
HVAC Sparks	ruleRef->hvac
All Sparks	spark

Debug

The Debug tab in the Spark App is used to manage the cache engine. The navigation bar provides access to various functionality:

- **Settings**: command on toolbar which allows tuning of ext level settings
- **Recompute**: command on toolbar to recompute portion of cache
- **Home**: root of debug navigation tree
- **Summary**: key performance variables within engine
- **All Rules**: summary of cache for each rule

- **Rule**: summary of cache for each target within a rule
- **Rule Target**: details of cache for a specific rule/target combination
- **Engine Log**: log of spark engine activity
- **DictBase**: details of dictbase files and which are loaded into main memory



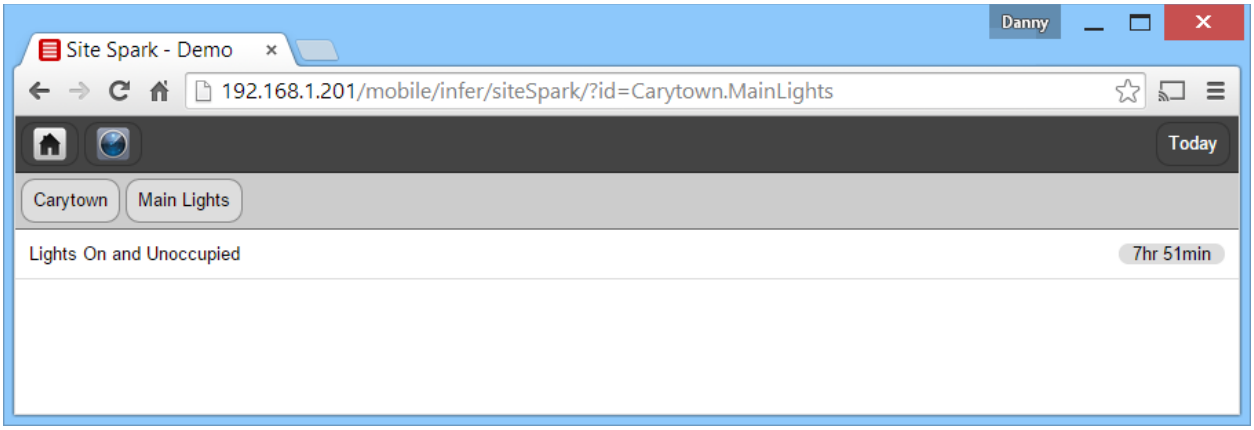
id	status	populated	populatedTo	startDate	targets	targetsInErr	callCount	callAvg
AHU Cool Failure	err	100 %	1-Jan-2014	1-Jan-2014	5	4	524	191.255 ms
AHU Cool and Econ	ok	100 %	1-Jan-2014	1-Jan-2014	5	0	2,600	42.804 ms
AHU Cool and Heat	ok	100 %	1-Jan-2014	1-Jan-2014	5	0	2,600	46.381 ms
AHU Cool-Heat Mode Cycling	ok	100 %	1-Jan-2014	1-Jan-2014	5	0	2,600	52.28 ms
AHU Fan Failure	ok	100 %	1-Jan-2014	1-Jan-2014	5	0	2,600	36.393 ms
AHU Fan Short Cycling	ok	100 %	1-Jan-2014	1-Jan-2014	5	0	2,600	21.493 ms
AHU Group Cool and Heat	ok	100 %		1-Jan-2014	0	0		
AHU Heat and Econ	ok	100 %	1-Jan-2014	1-Jan-2014	5	0	2,600	43.52 ms
AHU On and Fan Off	ok	100 %	1-Jan-2014	1-Jan-2014	5	0	2,600	62.097 ms
AHU Outside Damper Stuck Closed	ok	100 %	1-Jan-2014	1-Jan-2014	5	0	2,600	184.166 ms
AHU Outside Damper Stuck Open	err	100 %	1-Jan-2014	1-Jan-2014	5	4	524	335.261 ms
Lights On and Unoccupied	ok	100 %	1-Jan-2014	1-Jan-2014	1	0	520	47.167 ms
Temp Sensor Failure	ok	100 %	1-Jan-2014	1-Jan-2014	15	0	7,800	18.227 ms

Mobile

If you log into InferStack from a smart phone or check the Mobile option, you get Mobile access to Site Spark if analytics is enabled.

Site Spark (if enabled)

Site Spark allows you to view sparks.



Extensions Without Apps

Green Button

Parse a Green Button XML file into a grid of usage data from any IO handle The results are returned as grid with following columns:

- ts: DateTime of interval starting timestamp in UTC
- val: usage value in native Green Button uom (unit of measurement)
- cost: cost in USD

Options may be used to implictly map the data:

- tz: Str timezone name to convert from UTC
- unit: Str unit name to convert from

Currently this function requires there to be exactly one entry with an `<IntervalBlock>` and exactly one entry with a `<ReadingType>`.

Historian Analytics Kit

The hisKit extension provides a library of functions for analyzing time series data stored in the historian.

Interpolation

When doing time series analytics we often want to correlate different histories. For example we might want to compute the differential between two temperature sensors. Depending on the system which originally sampled the data, the time series is unlikely to be aligned by timestamp precisely. For example a typical query might result in:

Windows and Slices

Often when analyzing time series data, we wish to examine "windows" of time. We formally define a *window* as a timestamp and duration pair. We use the term *slice* to indicate a sub-set of historical data which is inclusive of a window.

HVAC

The hvacExt is used to model equipment associated with heating, cooling, and air conditioning. The HVAC extension enhances the site/equip/point model as defined by the equipExt.

The following equipment types are defined by the hvac extension:

- ahu: air handler unit (custom AHU or packaged RTU)
- vav: variable air volume unit
- chillerPlant: group equip used to generate chilled water
- chiller: remove heat from a liquid
- coolingTower: transfer waste heat into atmosphere
- heatExchanger: transfer heat from one medium to another
- boilerPlant: group of one or more boilers
- boiler: generates hot water or steam for heating

Lighting

The lightingExt is used to model the lighting system of a facility. It enhances the site/equip/point model as defined by the equipExt.

Tariff

The Tariff Extension provides a common model for describing energy tariffs in SkySpark.

To setup a tariff and apply it to a meter, the following high-level steps need to be completed.

1. Create a tariff record
2. Create records for all the charges in your tariff
3. Create the tariff history point that defines the bill periods for your meter

Conclusion

That was a brief overview of the analytics option of InferStack.

See http://www.intellastar.com for more information.