SESSION
05

Edge Analytics Online Training

NON STREAMING DATA

Work with non streaming data.
Install a local node.

Lecturers

Göran Appelquist

CTO

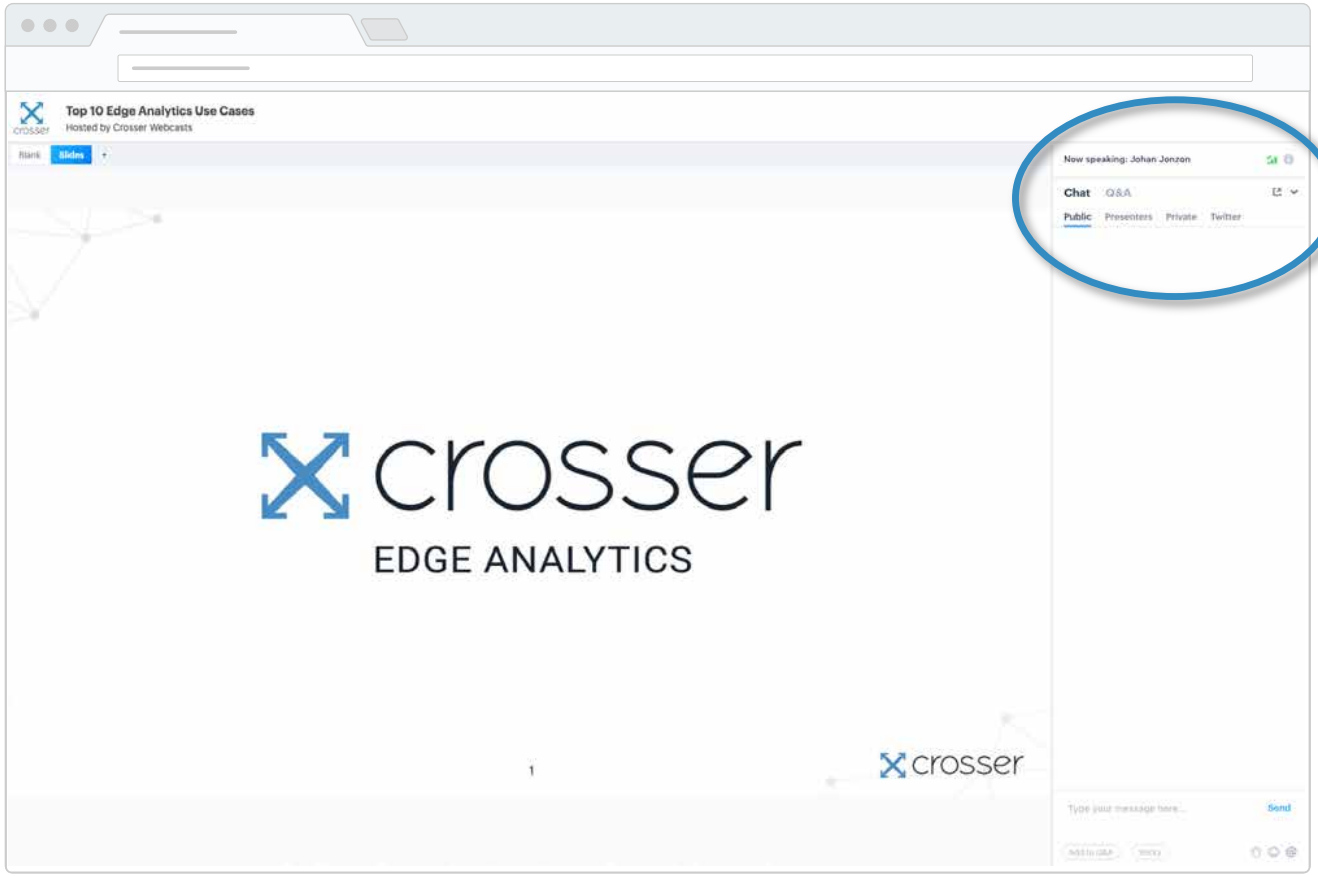
goran.appelquist@crosser.io

Mikael Isenberg

Solution Architect
Product Owner

mikael.isenberg@crosser.io

crosser


Top 10 Edge Analytics Use Cases
Hosted by Crosser Webcasts
Blank
Slides
crosser
EDGE ANALYTICS
1
crosser
Now speaking: Johan Jonzon
Chat
Q&A
Public
Presenters
Private
Twitter
Send

crosser

RECAP
- Session 4

- Modules
  - Modbus Reader
  - Array modules
- Other functions
  - Tag lists as resource
  - Manual tags

crosser

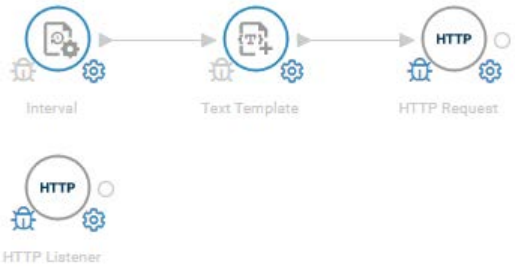
SESSION 5

Exercise 4

- Interval module
- Non streaming data
  - HTTP REQUEST
  - HTTP LISTENER
- Universal Connector

Exercise 5

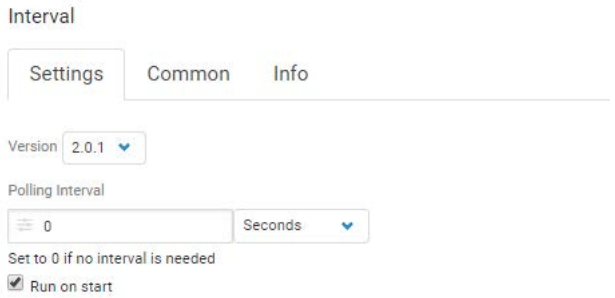
- Install a local Crosser Node
  - Docker
  - Windows
- Data directory
- Labels
- Dashboard



crosser

MODULE
- Interval

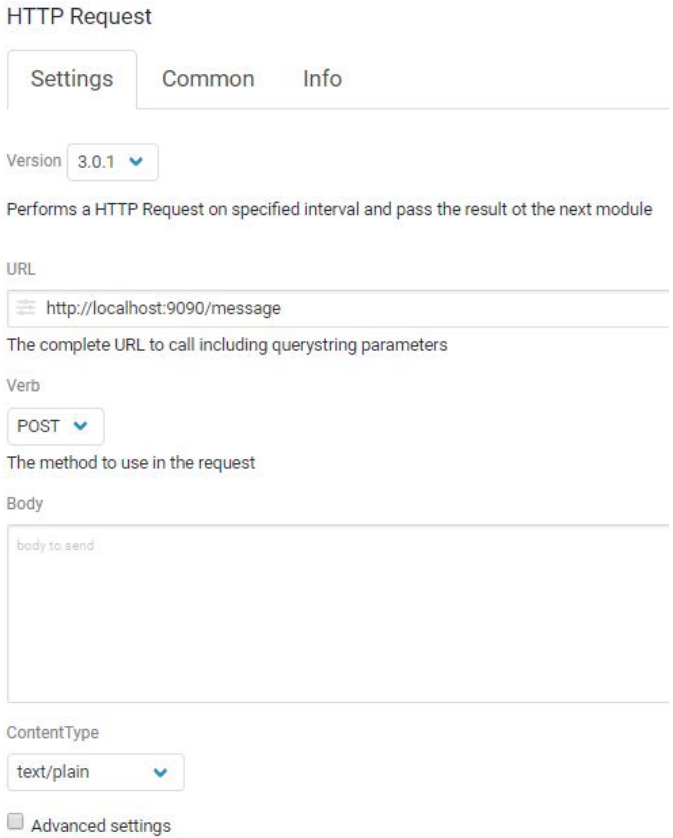
- Used when you want to input data into a flow without setting up external calls
- Set interval to call next module
  - Set 0 for only one call
- Run on start
  - True = The call to next module will be made instantly at start



crosser

MODULE
- HTTP Request

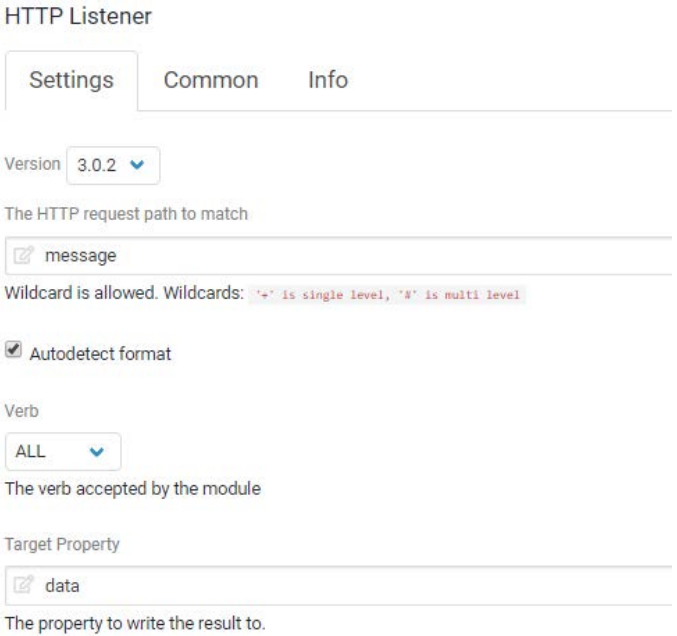
- HTTP client that can generate external HTTP requests with fixed or dynamic data (from flow messages)
- The URL is fixed, for dynamic URLs use a Universal Connector
- Can be an input (GET) or an output (POST) module
- Can use Basic or Bearer authentication (advanced setting)



crosser

MODULE
- HTTP Listener

- Internal HTTP server
- Listens on port 9090 by default
  - Can be changed in docker-compose.yml with Docker
  - Can be changed in data/httpconfiguration.json when running as a Windows service
- Incoming data can be filtered on the URL path (routing) and the verbs to accept. Wildcards can be used to match URLs.
- Data can be converted from JSON if Content-Type headers are provided



crosser

UNIVERSAL CONNECTOR

- Build your own REST API connectors

- Generic Modules
- Used to connect to REST API:s
- Wizard to create your own Universal Connector modules
- All Universal Connector modules are available from the FlowStudio as any another module

crosser

UNIVERSAL CONNECTOR

crosser

EXERCISE 4

Use the HTTP Request module as an Input

Use the HTTP Request as an Output

EXERCISE 4A: NON-STREAMING DATA

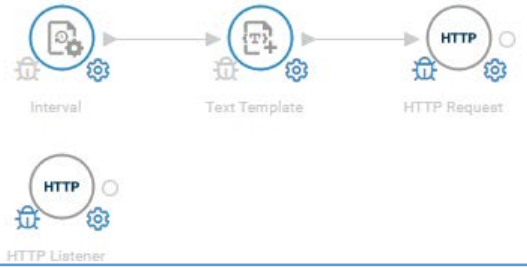
- Use the HTTP Request module as an Input

1. Create a new flow and add an ‘Interval’ module. Set the *Polling Interval* to 0 and make sure *Run on start* is enabled.
2. Add a ‘HTTP Request’ module:
    1. Set the *URL* to: https://api.chucknorris.io/jokes/random
    2. Make sure GET is set as *Verb*, ie what HTTP request to make
3. Run the flow and check the output of the ‘HTTP Request’ module (*data* property)
4. Add a ‘JSON’ module and check the result

crosser

EXERCISE 4B: NON-STREAMING DATA

- Use the HTTP Request module as an Output



1. Create a new flow and add an ‘Interval’ module. Set the *Polling Interval* to 0 and make sure *Run on start* is enabled.
2. Add a ‘Text Template’ module and enter some message. Set the *Target Property* to body
3. Add a ‘HTTP Request’ module:
    1. Set the *URL* to: http://localhost:9090/message
    2. Make sure POST is set as *Verb*, and *Content Type* is set to plain/text
4. Add a ‘HTTP Listener’ module and set the *Path* to message
5. Run the flow and check the output of the ‘HTTP Listener’ module
6. Change the *Path* setting in the ‘HTTP Listener’ module to see how the path can be used to select data

crosser

EXERCISE 5

Install a local Crosser node

Install local Crosser Node

- Two options for installation of a Crosser Node
  - As a Docker container
  - As a Windows service
- How to install you Crosser Node you will find in the documentation

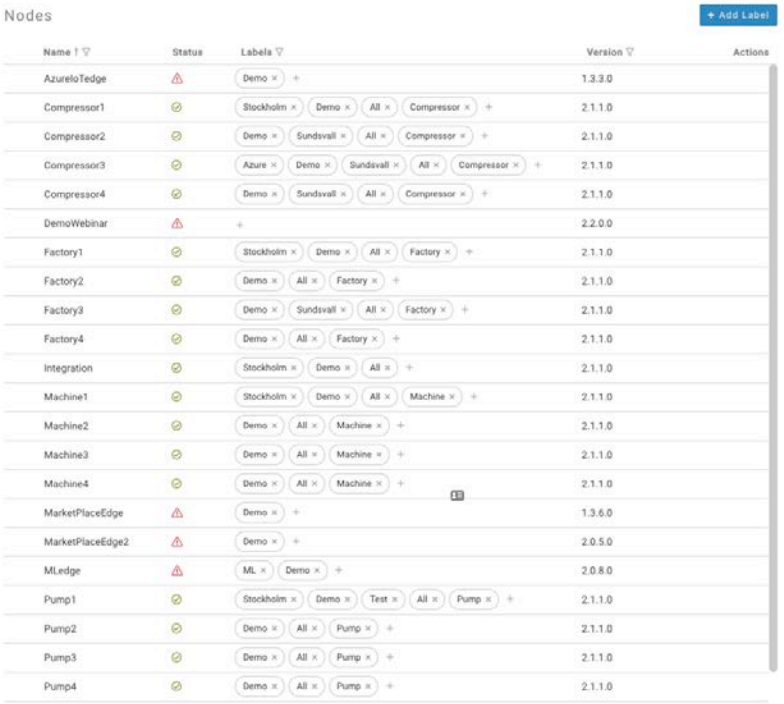
THE ‘DATA’ DIRECTORY IN CROSSER NODES

- Maps local file system to container file system
- Used for:
  - Node configuration files - *(data/*.json)*
  - Log files - *(data/logs)*
  - Flows - *(data/flows)*
  - Resources - *(data/flowresources)*
- Can be accessed from flow modules as “data/…”, eg in file readers and code modules

These files are critical for the operation of the node – Be careful!

LABELS
– Group Nodes

- Labels are strings added to Nodes
- Labels are used for:
  - Group Nodes together.
  - Deploy the same flow to several Nodes at once.
  - User rights



crosser

DASHBOARD

- The Dashboard is based on status information from the Nodes delivered at regular intervals (default every 10 seconds)
- You can see:
  - Connectivity summary
  - Traffic summary over all nodes (current values and graphs)
  - Per node data:
    - Connectivity status
    - Software version
    - CPU/Memory usage (hw platform, not container)
    - Traffic in/out
    - Flows running/stopped

EXERCISE 5a: Install a local Crosser node
- Option 1: Docker

- Prerequisites
  - Docker is installed
  - Docker Compose is installed
- Installation steps (see docs)
  1. Register a node on the *Nodes->Register Nodes* page (copy the *NodeId* and *AccessKey*)
  2. Login to docker.crosser.io (credentials in Crosser Cloud) from your local machine
  3. Get docker-compose.yml file (from documentation)
  4. Create a file named edgenode.env for node credentials and add the credentials you generated
  5. Start the Node with ‘docker-compose up –d’
  6. Check on the *Nodes* or *Dashboard* pages that the node is active
  7. Test one of your flows on this node

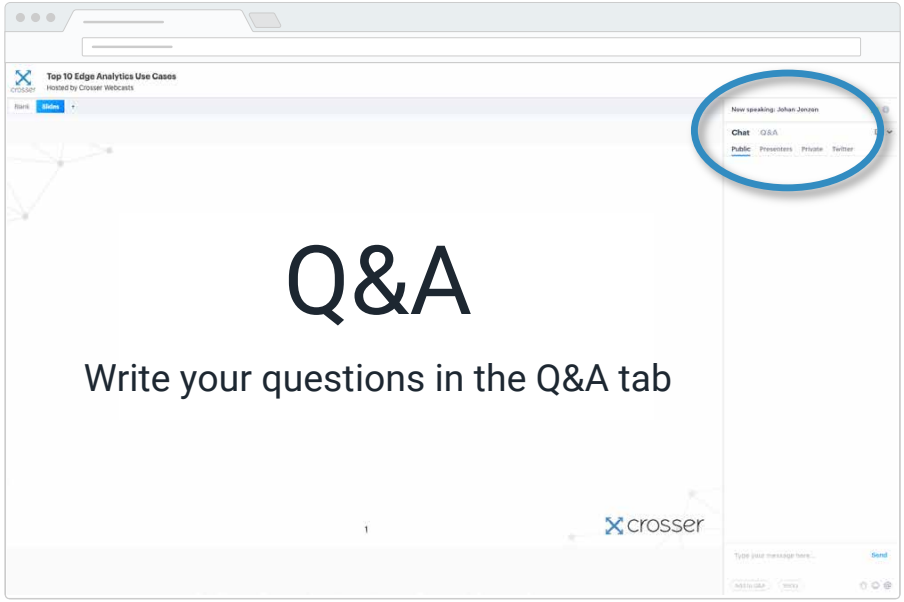
EXERCISE 5a: Install a local Crosser node
- Option 2: Windows Service

- Prerequisites
  - Windows 10
- Installation steps (see docs)
  1. Register a node on the *Nodes->Register Nodes* page (copy the *NodeId* and *AccessKey*)
  2. Download the Windows installer from the *Nodes* page, on the *Register Nodes* tab
  3. Start the Node with 'Crosser.EdgeNode.Service.Windows start'
  4. Check on the *Nodes* or *Dashboard* pages that the node is active
  5. Once verified that the Node is running and connected to cloud you can stop the Node with 'Crosser.EdgeNode.Service.Windows stop'
  6. Install the Windows service with 'Crosser.EdgeNode.Service.Windows install'
  7. Check on the *Nodes* or *Dashboard* pages that the node is active
  8. Test one of your flows on this node

EXERCISE 1G: NODE GROUPS

1. On the *Nodes* page, add a label with your name to at least 2
2. On the *Flows* page, use the *Deploy* function on one of your flows without the 'Nexmo' module
3. In the popup select *Labels* and in the list select the label with your name and do *Deploy*
4. Click on the number in the *Nodes* column on your flow and verify the flow is now running on the nodes where you added the label
5. Remove the flow from all nodes using the *Delete* function in the *Nodes* window

SESSION – 05 END



Mikael Isenberg
Solution Architect
Product Owner
mikael.isenberg@crosser.io