Webapps

Make use of web applications at DESY cryogenic controls dept.

Joerg Penning
EPICS-Conference at ITER June 2019
Overview

1 Motivation

2 Functional requirements
   • Asset management
   • Displaying archived data
   • Displaying live data

3 Realization
   • Implementation
   • Tools
   • Deployment
Motivation

Exchange of ideas

• We will be happy to learn what others do and tell what we do …

• Are we able to participate in an open source community?

Background

• Cryogenic controls
  i.e. IT infrastructure for all cryogenics

• Support for facilities group

Open Source

• Of course
Motivation

A lot of different assets

> 650 Valves
> 2500 Temp. Sensors
> 800 Press. Sensors
> 100 Level Sensors
> 200 Flow Sensors
Functional requirements

Should be accessible without further preparation, i.e. on the smart phone

Asset management
- Repair
- Stock
- Cost units

Archive data retrieval
- Watch trends when you are out-of-office on-site or on-call

Probe-style live data
- Find out about some record when you are out-of-office

OPI-style data display
- Watching the energy consumption
- Gained attention quickly at DESY
Asset management
A defective part must be exchanged

- Exchange part
  - Where are the spare parts?
  - Which procedure must be followed?
  - Which control system processes will be affected?
- Handle the defective part
  - Do we still have warranty?
  - Where to send to for repair?
  - Do we have to fill up stock?
- Take precautionary measure
  - Do other components of this type have problems too?
  - Where are they built in?
Asset management
Make use of heterogeneous databases

Asset-Database
- Manages orders and stock
- Enhanced to trace status (location)
- Enhanced to contain documentation
- IDs: Asset-ID, Inventory-ID

DCT-Database (Database Creation Tool)
- IDs: IOName, Channelname
- Side effect of creating EPICS-Databases

IO-Configuration-Database (‘Device Database’)
- IDs: IOName, Asset-ID
- Contains Profibus-Addresses, Documentation

DESY Inventory
- ID: Inventory-ID
Generic handling of xml data in the browser

Calling the (OSGi-) services that provides the data.

Calling the web application that provides the data (different services)

CSS related data

XML-Data

Tomcat with embedded OSGi-Framework

Service Dispatcher

Service Inventory

Service Asset

Service Device

Service EPICS

Service LDAP

Service Cable

Scanning some device code

CSS control system studio
Asset management

From scanned label to inventory
Asset management

Showing details and history
Asset management

Perform status updates
Asset management

Make use of the ‘is-contained-in’ relationship
Data display

Archived data via Archive Request Server
Data display

Live data in Probe-style
Data display
Live data in OPI-style

Lastmanagement DESY
Sat 01 Jun 2019 19:14:47

Leistungsumaximum
Gesamt DESY
15 Min-Wert SOLL
15 Min-Wert IST

MW
19,897

21.4 MW
19.9 MW
98.33 %
91.71 %

Generator in Betrieb
Generator gestört
Overview

1 Motivation

2 Functional requirements
   • Asset management
   • Displaying archived data
   • Displaying live data

3 Realization
   • Implementation
   • Tools
   • Deployment
Implementation

Web-Server backend uses the CSS code base and the service dispatcher
Implementation

Web-Server frontend

LiveScript
Angular
Stomp
ngx (Widgets)

Frontend

Backend

Web-Server

Frontend

client side in browser - frontend

Websocket - STOMP

HTTP REST

webserver - display-backend

Websocket - CAJ - Dal2

external server providing PV data via Dal2
Implementation

Web-Server frontend communication
Tools

Eclipse
- Web-Server-Backend
- CSS code base

Git-Repository
- stash.desy.de
- (not yet)

Visual Studio Code
- Web-Server-Frontend
- Live Script
Deployment
What do we need to make it run?

Web-Server

- Spring Boot application is deployed as a .war file to some DESY-Server
- IT opens the gate for us

Scanner-App

- Is required for scanning the data matrix labels
- Available for iOS and Android
Thank you
Contact

DESY. Deutsches Elektronen-Synchrotron
Jörg Penning
MKS-2 Cryogenic Controls
joerg.penning@desy.de

www.desy.de