Extending and Reusing the New ONOS GUI2

Sean Condon
Open Networking Foundation
A brief history - ONOS GUI, GUI2 and µONOS GUI

ONOS GUI (2015-2017)
AngularJS 1.x based with Java backend - Web Sockets

ONOS GUI2 (2018-)
Angular 7.x based with same Java backend and Web sockets
Divided in to Framework library, Topo Library and main app

µONOS GUI (2019-)
Angular 7.x based with Go backend and grpc-web
For onos-config and onos-topo
Reuses Framework library
What’s new? Not your old web app stack

• ECMAScript 6+ (ES6) - 2015
  • Not the Javascript you knew - OO and functional
  • Supported by all modern browsers now
• Typescript - strongly typed extension of ES6+
• Angular 2+ - Google’s open source web app platform
• Node JS 10/12, NPM, Bazel

Tooling

• Intellij Ultimate Edition or Web Storm
• Atom or MS VS Code
• Angular CLI
Web Development - 3 dimensions

Angular Component is an isolated section of all 3

- TS, CSS and HTML
- @Input() for parameters
- @Output() for events
- Inject services
- Referred to by selector
- CSS and dom is isolated
GUI2 on ONOS is about choosing

- best practice
- consistent approach

Libraries to make reusable

Use Angular structures at top level

- Module is a grouping unit
- Component is the main visual element

Lots and lots of choice - too much choice?
ONOS GUI2

- gui2
  - onos module
  - apps module
  - details module
  - flows module
  - ports module
  - intent module
  - ...

- gui2-topo-lib
  - gui2-topo-lib module

- gui2-fw-lib
  - gui2-fw-lib module

- fm-gui2-lib
  - fm-gui2-lib module

- roadm-gui-lib
  - roadm-gui-lib module

- Web sockets & REST

ONOS
- Flows
- Intents
- Applications
- ...

FM App

ROADM App

ONOS GUI
Java Backend
gui2-topo-lib in more detail

Panel Base

summary
topology

toolbar
instances
map-selector
details

background-svg
grid-svg
force-svg
map-svg
nodevice-svg
subregion-svg*
devicenode-svg*
hostnode-svg*
host-svg*

Zoomable
Draggable

Force Directed Graph

Traffic Service
Topology Service
Layout Service
Network Changes

- Reuse of mast & navigation
- Pending change selected
- Previous network changes
- Rollback (if existing change selected)
- Delete or commit Pending change
- Only one config part of this change
- Can add more

Network changes (4 total)

<table>
<thead>
<tr>
<th>NAME</th>
<th>USER</th>
<th>CREATED</th>
<th># CHANGES</th>
</tr>
</thead>
<tbody>
<tr>
<td>myPendingChanges</td>
<td>onos</td>
<td>Sep 2, 2019</td>
<td></td>
</tr>
<tr>
<td>added_a_stratum</td>
<td>User1</td>
<td>Sep 2, 2019</td>
<td></td>
</tr>
<tr>
<td>a_new_stratum</td>
<td>User1</td>
<td>Sep 2, 2019</td>
<td></td>
</tr>
<tr>
<td>keen_euler</td>
<td>User1</td>
<td>Sep 2, 2019</td>
<td></td>
</tr>
</tbody>
</table>

Remote Device

- Changes
  - Configuration Change
  - strat1-1.0.0 pending

- Other Configs to Add
  - strat2-1.0.0
  - device-741375084-1.0.0

- User: onos
  - Created: Sep 2, 2019, 2:18:20 PM
Configuration view

Layers panel
- Path bar
- Config layers
- Model plugin driven

Value editor
- Pending change
- Previous Config
- RW paths (for adding to config)
onos-config in more detail

- Configurations
- NW Changes
- Models
- Details
- Configuration
- Layers Panel
- Path Bar
- String Editor
- Boolean Editor
- Number Editor
- Zoomable
- Details
- Layer SVG*
- Container SVG*
- Change Value Pipe
- Link Filter Pipe
- Node Filter Pipe
- Model Service
- Model Temp
- Index Service
- Pending Net
- Change Service
- Hierarchy
- Layout Service
- Path Bar
- Value Utils
- OnosConfigGnmi Service
- OnosConfigAdmin Service
- OnosConfigDiags Service
- OnosTopoDevice Service
Code structure

- **Inputs and Outputs**
- **Injected services**
- **Default layout for component**
- **Default layout for service**
Angular CLI
Demo - get coding

1. Create new component to view changes in table
2. In the TS file add a new @Input() for the changeIDs
3. In its constructor add a reference to the OnosConfigDiagsService
4. Make call to service, implement callback - local cache. Proto file
5. Look at the @Component header - HTML, CSS, Selector
6. Edit the HTML to add a table and bind to data - *ngFor
7. Edit the CSS file for the table - border and background
8. Insert the selector for new component in to Layers Panel
9. Run with ‘ng serve’ to see it in action - Console, Inspect, Network
Demo - make a routed component - instead
10. Add an entry to the routing module
11. Make the entry in to Layers Panel in to a button instead
12. Inject Activated Route to constructor
13. Add ngOnInit to handle parameter
14. Can remove ngOnChanges and the @Input
15. Run and show the routing
Next steps

• Fix all remaining issues on onos-gui
• Update onos-config proto files for continuous updates
• As onos-control is developed migrate topology view to μONOS
• Add in full support for onos-topo
• Upgrade to Angular 8
• Move some onos-gui common pieces to gui2-fw-lib
How to Engage with Community

- Join #micro-onos channel on onosproject.slack.com
- Attend ONOS TST meetings
  - bi-weekly Wednesdays at 9:00 PST/PDT
- Fork and send pull-requests to https://github.com/onosproject/onos-gui repositories
- Participate in onos-dev@onosproject.org mailing list
- Questions?
  - email sean@opennetworking.org
Thank You

Follow Up Links:

https://github.com/opennetworkinglab/onos/blob/master/web/gui2/README.md
https://github.com/onosproject/onos-gui