
CMS Trigger Supervisor Software Documentation

Release 3.4.0

Glenn Dirkx

May 24, 2016

CONTENTS

1	Building your own Cell Panels	1
1.1	Welcome	1
1.2	Quickstart section	1
1.3	Advanced section	132
1.4	Available resources	140
2	Sphinx syntax examples	145
2.1	This is a Title	145

BUILDING YOUR OWN CELL PANELS

1.1 Welcome

This documentation will show you how to develop panels.

1.1.1 Scope of this document

This document contains both basic info and quickstarters, but also very detailed descriptions of the inner workings of the technologies used.

All examples are taken from the Subsystem Supervisor unless otherwise specified. (<https://svnweb.cern.ch/trac/cactus/browser/trunk/cactusprojects/subsystem/supervisor>)

1.2 Quickstart section

1.2.1 Setting up the Cell

Starting from this point we will assume you already have a working cell and you now have arrived at the point you wish to develop panels for it.

Front-end code (HTML, CSS, and JavaScript) have a separate build cycle, separate from the makefile. These are the steps necessary to setup this build system.

Making your life easier

We have copied the files that will be created in this page into a tarball. This will absolve you from having to create any files.

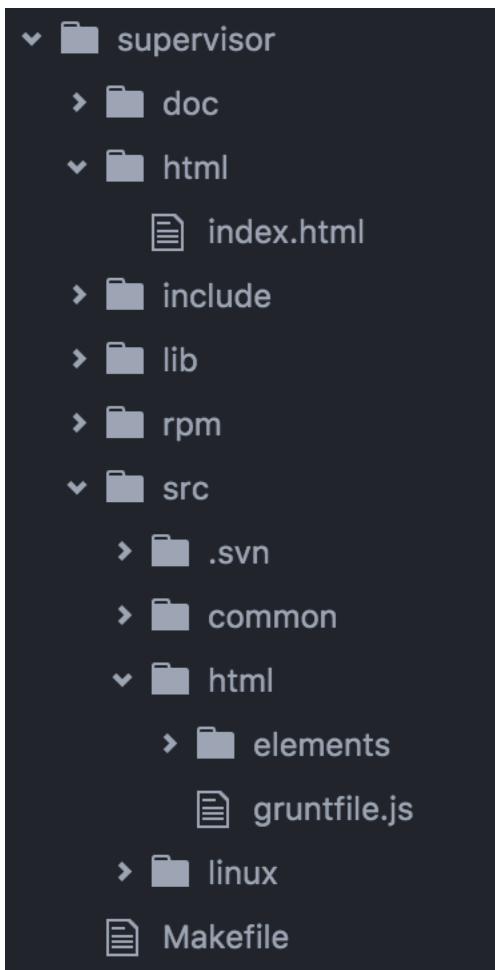
To use it, run:

```
1  svn export svn+ssh://svn.cern.ch/repos/cactus/trunk/cactuscore/ts/doc/cell-skeleton.tar
2  tar -xzvf cell-skeleton.tar
```

Now you can continue following this tutorial, but you don't have to create files anymore.

Making directories

Extend the file structure of your cell to include the following folders and files:



The html folder will contain the build output of the source files in the /src/html folder.

If you have static resources you wish to serve in your panel, put them in the /html folder. /src/html is only meant for source files that need to be processed in some way.

Setting up Grunt

A panel is composed of different code languages, namely HTML, CSS, and JavaScript. When you develop a panel, each of these languages are housed in their own files.

This makes things easier for you to read and allows your editor to use code highlighting and syntax checkers.

It also allows us to perform optimizations on your code. The JavaScript will be optimized and the SASS code will be compiled into CSS and enhanced for compatibility.

The build system will put all generated files into the /html folder.

Grunt (<http://gruntjs.com/>) is the tool we'll use to accomplish all this.

if you do not have npm installed, run

```
1 sudo yum install -y npm
```

now run

```
1 cd src/html
2 sudo npm install -g grunt-cli
```

This will install the grunt command line tools on your system, you only need to do this once since it's a global install (the -g option).

Put this in the /src/html/gruntfile.js file

/src/html/gruntfile.js

```
1 require('es6-shim');
2 module.exports = function(grunt) {
3     grunt.initConfig({
4         /*
5          *      Compile SASS code to CSS
6          */
7         sass: {
8             options: {
9                 sourcemap: 'none',
10                // style: 'expanded',
11                outputStyle: 'compressed',
12                noCache: true
13            },
14            dist: {
15                files: [{
16                    expand: true,
17                    cwd: '',
18                    src: [
19                        'elements/**/*.scss'
20                    ],
21                    dest: '',
22                    ext: '-min.css'
23                }]
24            }
25        },
26        /*
27         *      Add css prefixes for compatibility (mainly for Firefox ESL)
28         */
29        postcss: {
30            options: {
31                map: false,
32                processors: [
33                    require('autoprefixer')({
34                        browsers: ['firefox 24', 'IE 10', 'last 2 versions']
35                    })
36                ]
37            },
38            dist: {
39                src: ['elements/**/*-min.css']
40            }
41        },
42        /*
43         *      Process JavaScript
44         */
45        uglify: {
```

```
46     options: {
47         preserveComments: false,
48         srewIE8: true,
49         sourceMap: false
50     },
51     /*
52      Compile JavaScript on Polymer Elements
53     */
54     polymerjs: {
55         options: {
56             mangle: false,
57             sourceMap: false
58         },
59         files: [{
60             expand: true,
61             src: ['elements/**/*.js'],
62             ext: '-min.js'
63         }]
64     },
65 },
66
67     inline: {
68
69         dist: {
70             files: [{
71                 expand: true,
72                 cwd: '',
73                 src: [
74                     'elements/**/*.html'
75                 ],
76                 dest: '../../html/',
77                 ext: '.html'
78             }]
79         }
80     },
81
82     clean: {
83         cssfiles: {
84             options: {
85                 'no-write': false
86             },
87             src: ["elements/**/*-min.css"]
88         },
89         jsfiles: {
90             options: {
91                 'no-write': false
92             },
93             src: ["elements/**/*-min.js"]
94         }
95     },
96
97     /*
98      * Make package documentation
99     */
100    execute: {
101        target: {
102            src: ['elements/makeIndex.js'],
103            options: {
```

```

104         cwd: "./elements/"
105     }
106   }
107 });
108 );
109
110 grunt.loadNpmTasks('grunt-execute');
111 grunt.loadNpmTasks('grunt-contrib-uglify');
112 grunt.loadNpmTasks('grunt-postcss');
113 grunt.loadNpmTasks('grunt-contrib-clean');
114 grunt.loadNpmTasks('grunt-sass');
115 grunt.loadNpmTasks('grunt-inline');
116
117   grunt.registerTask('default', ['execute', 'sass', 'postcss', 'uglify', 'inline',
118   ↪'clean']);
119 };

```

This specifies what Grunt has to do, where to find source files, and where to put built files.

Setting up documentation

Your cell will automatically generate documentation. So anyone running your cell can browse to <hostname>:<port>/<package-path>/html/index.html and explore what elements your cell contains.

Make the /src/html/elements/makeIndex.js file and edit the first few lines.

/src/html/elements/makeIndex.js

```

1  #!/usr/bin/env node
2  var repositoryPath = "https://svnweb.cern.ch/trac/cactus/browser/trunk/cactusprojects/
3   ↪subsystem/supervisor/html-dev/elements/";
4  var projectName = "Subsystem Supervisor";
5  var projectPath = "subsystem/supervisor/html/elements/"
6
7  var fs = require('fs');
8  var path = require('path');
9
10 function getDirectories(srcpath) {
11   return fs.readdirSync(srcpath).filter(function(file) {
12     return fs.statSync(path.join(srcpath, file)).isDirectory() && file.indexOf('-') >
13   ↪-1 && file.indexOf('template') == -1;
14   });
15 }
16
17 var elements = getDirectories('.');
18 var result = [];
19 for (var i = 0; i < elements.length; i++) {
20   var element = elements[i];
21   var json = {name: element};
22   if (fs.existsSync(element + '/description.json')) {
23     var parsedJSON = require("./" + element + '/description.json');
24     if (!parsedJSON.description) {
25       console.error(element + '/description.json contains no description');
26     }
27     for (var property in parsedJSON) {
28       if (parsedJSON.hasOwnProperty(property)) {

```

```
28         json[property] = parsedJSON[property];
29     }
30 }
31 } else {
32     console.error(element + ' has no description.json file');
33     json.description = "no description...";
34 }
35 if ( !fs.existsSync(element + '/index.html') ) {
36     console.error(element + '/index.html does not exist');
37 }
38 result.push(json);
39 }
40
41 fs.readFile("./index_template.html", 'utf8', function(err, data) {
42     if (err) {
43         return console.error("index_template.html is missing or not readable");
44     }
45     data = data.replace(/<% data %>/g, JSON.stringify(result));
46     data = data.replace(/<% repositoryPath %>/g, repositoryPath);
47     data = data.replace(/<% projectName %>/g, projectName);
48     data = data.replace(/<% projectPath %>/g, projectPath);
49     fs.writeFile("index.html", data, function(err) {
50         if(err) {
51             return console.log(err);
52         }
53
54         console.log("index.html written");
55     });
56});
```

This will generate our package documentation using a template html file.

Create /src/html/elements/index_template.html

/src/html/elements/index_template.html

```
1 <!DOCTYPE html>
2 <html>
3     <head>
4         <meta charset="utf-8">
5         <title><% projectName %> documentation</title>
6
7         <meta name="viewport" content="width=device-width, minimum-scale=1.0, initial-
8             ↪scale=1, user-scalable=yes">
9             <meta name="mobile-web-app-capable" content="yes">
10            <meta name="apple-mobile-web-app-capable" content="yes">
11
12            <script>window.Polymer = window.Polymer || {} ; window.Polymer.dom = 'shadow';</
13             ↪script>
14             <script src="/extern/bower_components/webcomponentsjs/webcomponents-lite.js"></
15             ↪script>
16
17                 <link rel="import" href="/extern/bower_components/paper-material/paper-material.
18                   ↪html">
19                     <link rel="import" href="/extern/bower_components/iron-icon/iron-icon.html">
20                     <link rel="import" href="/extern/bower_components/iron-icons/iron-icons.html">
21                     <link rel="import" href="/extern/bower_components/prism-element/prism-highlighter.
22                         ↪html">
23                         <link rel="import" href="/extern/bower_components/marked-element/marked-element.
24                             ↪html">
```

```
19 <style>
20   body {
21     background-color: #FAFAFA;
22   }
23   paper-material {
24     max-width: 800px;
25     margin: auto;
26     background-color: white;
27   }
28   marked-element {
29     padding: 10px;
30     padding-top: 0;
31     display: block;
32     margin-bottom: 1em;
33   }
34   div[title] {
35     background-color: #81c784;
36     color: black;
37     font-family: roboto;
38     font-weight: normal;
39     margin: 0;
40     padding: 10px;
41     display: flex;
42     flex-direction: row;
43     padding: 15px 24px;
44     font-size: .8em;
45   }
46   div[title] [name] {
47     width: 250px;
48   }
49   h1 {
50     color: black;
51     font-family: roboto;
52     font-weight: normal;
53     margin: 0;
54     padding: 15px 24px;
55     font-size: 1.5em;
56   }
57   h2 {
58     color: black;
59     font-family: roboto;
60     font-weight: normal;
61     margin: 0;
62     text-align: center;
63     font-size: 1em;
64     margin: 0;
65     padding: 0;
66     padding-left: 10px;
67     padding-top: 10px;
68   }
69   a {
70     display: flex;
71     flex-direction: row;
72     padding: 15px 24px;
73     text-decoration: none;
74     color: black;
75     font-family: roboto;
76     border-bottom: 1px solid #EFEFEF;
```

```
77      }
78      a div[options] {
79          opacity: 0;
80          transition: opacity 200ms;
81      }
82      a:hover {
83          background-color: #fafafa;
84      }
85      a:hover div[options] {
86          opacity: 1;
87      }
88      a div[name] {
89          width: 250px;
90          font-weight: bold;
91      }
92      a div[flex] {
93          color: #757575;
94      }
95      [flex] {
96          flex: 1;
97      }
98      a div {
99          font-size: .8em;
100     }
101     iron-icon {
102         height: 16px;
103         z-index: 10;
104         color: #797979;
105     }
106     iron-icon:hover {
107         color: black;
108     }
109 
```

</style>

```
110 </head>
111 <body>
112     <h1><% projectName %> elements</h1>
113     <paper-material elevation="1">
114         <prism-highlighter></prism-highlighter>
115         <h2>To use any of these elements in your project:</h2>
116         <marked-element id="marked">
117             <div class="markdown-html code" id="code"></div>
118         </marked-element>
119     </paper-material>
120
121
122     <template is="dom-bind" id="app">
123         <paper-material elevation="1">
124             <div title>
125                 <div name>
126                     name
127                 </div>
128                 <div flex>
129                     description
130                 </div>
131             </div>
132             <template is="dom-repeat" items="{{data}}" as="element">
133                 <a href="#">\[\[element.name\]\]/index.html
```

```

135           [[element.name]]
136       </div>
137       <div flex>
138           [[element.description]]
139       </div>
140       <div options>
141           <iron-icon icon="description"></iron-icon>
142           <iron-icon icon="code" on-click="showCode" onclick="return false;"></
143           iron-icon>
144           <template is="dom-if" if="[[element.demo]]" restamp="true">
145               <iron-icon icon="visibility" on-click="showDemo" onclick="return_
146               false;"></iron-icon>
147               </template>
148           </div>
149       </a>
150   </template>
151 </paper-material>
152 </template>
153 <script>
154     var codeRepositoryRoot = "<% repositoryPath %>";
155     window.addEventListener('WebComponentsReady', function(e) {
156         var app = document.querySelector('#app');
157         app.data = <% data %>;
158
159         var el = document.getElementById('marked');
160         var snippet = el.unindent('<link rel="import" href="<% projectPath %>element-
161         name/element-name.html">');
162         // Boolean properties are displayed as checked="", so remove the ="" bit.
163         snippet = snippet.replace(/=""\//g, '');
164         el.set('markdown', '```\n' + snippet + '\n' + '```');
165     });
166     app.showCode = function(e) {
167         window.location = codeRepositoryRoot + e.model.element.name;
168     }
169     app.showDemo = function(e) {
170         window.location = e.model.dataHost.dataHost.element.name + "/" + e.model.
171         dataHost.dataHost.element.demo;
172     }
173 </script>
174 </body>
175 </html>

```

Now make a file html/index.html

/html/index.html

```

1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title></title>
6          <meta http-equiv="refresh" content="0; URL='elements/index.html'" />
7      </head>
8      <body>
9
10     </body>
11 </html>

```

This will redirect the user to the elements folder holding the documentation when they visit <hostname>:<port>/<package-name>/html/index.html

Using Grunt

Now you should be able to run

```
1 cd src/html  
2 grunt
```

And you should see whatever elements are present in /src/html/elements are built and put into /html/elements (you probably don't have any elements now).

Your makefile will copy the /html folder into the RPM's. The src/html folder will not be copied and will not be present in production systems. Anything that does not need building can be safely copied into the /html folder. No build system will delete that folder.

Now that you have an update /html folder you can run

```
1 make rpm
```

The makefile will make a new rpm containing the updated /html folder.

Setting up a template

You will probably want to make some elements of your own now, but where to start? We'll give you a script that, using some template element, can generate a general element definition for you.

Create the file /src/html/elements/new-element.js file

/src/html/elements/new-element.js

```
1 #!/usr/bin/env node  
2 process.stdin.resume();  
3 process.stdin.setEncoding('utf8');  
4 var util = require('util');  
5 var ncp = require('ncp').ncp;  
6 var replace = require("replace");  
7 var renamer = require("renamer");  
8 var path = require('path');  
9 ncp.limit = 16;  
10 var FindFiles = require("node-find-files");  
11 var fs = require('fs');  
12 var exec = require('child_process').exec;  
13  
14 process.stdout.write('name of the new element: ');\n15 process.stdin.on('data', function (text) {  
16  
17     var split = text.replace('\n', '').split('/');  
18     if (split.length == 1) {  
19         base = split[0];  
20         newname = base;  
21     } else if (split.length == 2) {  
22         base = split[0];  
23         newname = split[1];  
24     } else {  
25         console.error('\nname can only contain only one dash (/)');
```

```

26     process.exit();
27 }
28
29 if (newname == '') {
30     console.error('\nname cannot be empty');
31     process.stdout.write('name of the new element: ');
32
33 } else if (newname.indexOf('-') == -1) {
34     console.error('\nname must contain a dash (-)');
35     process.stdout.write('name of the new element: ');
36
37 } else {
38     if (split.length == 1 ) {
39         console.log('creating new element <' + base + '>...');

40     } else {
41         console.log('creating new element <' + newname + '> in <' + base + '>...');

42         return console.error("unfortunately we can't do this because we will mess up
43             ↪the .svn folders");
44     }
45
46     ncp('template-element', base, function (err) {
47         if (err) { return console.error(err); }

48         replace({
49             regex: "template-element",
50             replacement: newname,
51             paths: [base],
52             recursive: true,
53             silent: true,
54         });
55
56         var finder = new FindFiles({
57             rootFolder : base,
58             filterFunction : function (path, stat) {
59                 return (path.indexOf('template-element') > -1) ? true : false;
60             }
61         );
62         finder.on("match", function(strPath, stat) {
63             // console.log(strPath + " -> " + strPath.replace('template-element', ↪
64             ↪newname));
65             fs.rename(strPath, strPath.replace('template-element', newname), ↪
66             ↪function(err) {
67                 if ( err ) console.log('ERROR: ' + err);
68             });
69         }
70         finder.on("complete", function() {
71             console.log("removing any .svn folders in ", newname);
72             exec('rm -rf `find ' + newname + ' -type d -name .svn`', function (err, ↪
73             ↪stdout, stderr) {});
74             console.log("Finished");
75             process.exit();
76         })
77         finder.on("patherror", function(err, strPath) {
78             // Note that an error in accessing a particular file does not stop the whole
79             ↪show
80             console.log("Error for Path " + strPath + " " + err);
81         }
82         finder.on("error", function(err) {

```

```
79         console.log("Global Error " + err);
80         process.exit();
81     })
82     finder.startSearch();
83 });
84 }
85
86 }) ;
```

This will copy a folder *template-element* and rename the appropriate code to the element name you specified.

Now create /src/html/elements/template-element/template-element.html

/src/html/elements/template-element/template-element.html

```
1 <link rel="import" href="/extern/bower_components/polymer/polymer.html">
2 <link rel="import" href="/ts/common-elements/reset-css/reset-css.html">
3 <link rel="import" href="/ts/common-elements/iron-flex-layout-attributes/iron-flex-
4   ↴layout-attributes.html">
5
6 <!--
7 only the last comment before <dom-module ...> appears in the documentation
8 -->
9
10 <!--
11 Material design: [Click me] (https://www.google.com/design/spec/components/text-fields.
12   ↴html)
13
14 `template-element` is a template to be used when creating new elements.
15
16 Example:
17
18   <template-element></template-element>
19   <template-element test="test"></template-element>
20
21
22   ### Styling
23   The following custom properties and mixins are available for styling:
24
25   Custom property | Description | Default
26   -----|-----|-----
27   `--my-custom-color` | A custom css property | `black`
28   `--my-mixin-name` | A custom mixin | `{}`
29
30
31   @demo demo/index.html
32
33
34   -->
35   <dom-module id="template-element">
36     <template>
37       <!-- this makes your element follow the general theme (things like fonts) -->
38       <style include="reset-css"></style>
39
40       <!-- this will allow you to use flexbox easily -->
41       <!-- surf to /ts/common-elements/iron-flex-layout-attributes/index.html -->
42       <style include="iron-flex-layout-attributes"></style>
43
44       <link rel="stylesheet" type="text/css" href="css/template-element-min.css?__
45         ↴inline=true">
```

```

42   <h1>This is the template-element element!</h1>
43   <paper-button raised on-click="makeDinosaur">[someproperty]</paper-button>
44
45   </template>
46   <script src="javascript/template-element-min.js?__inline=true"></script>
47 </dom-module>
```

Notice the big comment just before the *dom-module* line. This will be used to generate documentation for your element. Be sure to update the description of the element in this comment.

Now create /src/html/elements/template-element/description.json

/src/html/elements/template-element/description.json

```

1  {
2    "description": "no description...",
3    "demo": "demo/index.html"
4 }
```

This file gives a description for the package documentation that will be generated by Grunt. Change the description to something sensible when you generate a new element with new-element.js. delete the *demo* line if you, at one point, decide to not provide a demo.

Now create /src/html/elements/template-element/index.html

/src/html/elements/template-element/index.html

```

1  <!--
2  This file renders documentation of the element
3  -->
4  <!doctype html>
5  <html>
6    <head>
7
8      <meta charset="utf-8">
9      <meta name="viewport" content="width=device-width, initial-scale=1.0">
10
11      <script src="/extern/bower_components/webcomponentsjs/webcomponents-lite.js"></
12      <script>
13        <link rel="import" href="/extern/bower_components/iron-component-page/iron-
14        component-page.html">
15
16    </head>
17    <body unresolved>
18      <!-- Note: if the main element for this repository doesn't
19          match the folder name, add a src=".html" attribute,
20          where <main-component>.html" is a file that imports all of the
21          components you want documented. -->
22      <iron-component-page></iron-component-page>
23
24    </body>
25 </html>
```

Now create /src/html/elements/template-element/javascript/template-element.js

/src/html/elements/template-element/javascript/template-element.js

```

1 Polymer({
2   is: 'template-element',
```

```
3      behaviors: [
4          // Polymer.PaperInputBehavior
5      ],
6
7      properties: {
8          /**
9             * Fired when you make a dinosaur
10            *
11            * @event made-a-dinosaur
12            */
13
14         /**
15             * The message the element will show
16             */
17
18         someproperty: {
19             type: String,
20             value: "Hello, World!",
21             //alternatively, this can be a computed property, based on other properties
22             // computed: 'computeFullName(first, last)'
23             //someproperty-changed event will be fired when property changes (required
24             ↪for data-binding to parent)
25             notify: true,
26             //element attribute will be updated when property changes
27             reflectToAttribute: true,
28             //function to execute if property changes
29             observer: '_disabledChanged',
30             //if true, cannot be updated except with _setSomeproperty(value)
31             readOnly: false
32         }
33     },
34     observers: [
35         // 'dosomething(someproperty, someotherproperty)'
36     ],
37
38     /**
39         * This will do something nice
40         *
41         * @param {Egg} egg The dinosaur egg.
42         * @return {Dinosaur}
43         */
44     makeDinosaur: function(egg) {
45         alert('you clicked the button!');
46         if (!egg) {egg = new Egg('velociraptor');}
47
48         // using this, developers can use your event to fire a function of their own
49         // <element-template on-made-a-dinosaur="customfunction"></element-template>
50         // the second argument is optional
51         this.fire('made-a-dinosaur', {fromEgg: egg});
52         return new Dinosaur(egg);
53     },
54
55     /**
56         * This is a private function, do not use
57         */
58     _destroyHumanity: function() {
59         // if you have a function you don't want others to use outside your element
60         // prefix the function with '_'
```

```

60     dinosaurs = new Array();
61     for (var i = 0; i < 100000000; i++) {
62       dinosaurs[i] = new Dinosaur();
63       dinosaurs[i]._killAllHumans();
64     }
65   },
66
67   // Fires when an instance of the element is created
68   // you have no data binding and the element does not contain html code yet
69   created: function() {},
70
71   // Fires when the local DOM has been fully prepared
72   // data binding works and the template html is ready
73   ready: function() {},
74
75   // Fires when the element was inserted into the document
76   attached: function() {},
77
78   // Fires when the element was removed from the document
79   detached: function() {},
80
81   // Fires when an attribute was added, removed, or updated
82   attributeChanged: function(name, type) {}
83 );

```

Note that this template serves as a boilerplate, and probably contains a lot of code you won't actually use. Delete lines you do not need in new elements you generate with this new-element.js script. Also notice the comments in the *properties* section and above every function definition. These are used to generate documentation for your element and follow the JSDoc syntax (<http://usejsdoc.org/about-getting-started.html>).

Now create /src/html/elements/template-element/demo/index.html

/src/html/elements/template-element/demo/index.html

```

1  <!doctype html>
2  <html>
3    <head>
4      <title>template-element demo</title>
5
6      <meta name="viewport" content="width=device-width, minimum-scale=1.0, initial-
7      ↪scale=1, user-scalable=yes">
8      <meta name="mobile-web-app-capable" content="yes">
9      <meta name="apple-mobile-web-app-capable" content="yes">
10
11      <script>window.Polymer = window.Polymer || {} ; window.Polymer.dom = 'shadow';</
12      ↪script>
13      <script src="/extern/bower_components/webcomponentsjs/webcomponents-lite.js"></
14      ↪script>
15
16      <link rel="import" href="../template-element.html">
17    </head>
18    <body unresolved>
19      <template-element></template-element>
20    </body>
21  </html>

```

This file is the demo. By default the demo only shows the element without any adjustments or data supplied to it. Adjust the demo if your element needs extra work or data before it becomes functional.

Now create /src/html/elements/template-element/css/template-element.scss

/src/html/elements/template-element/css/template-element.scss

```
1 // for more info about styling an element:
2 // https://www.polymer-project.org/1.0/docs/devguide/styling.html
3
4 :host {
5     // always declare a display property for your element, otherwise it will appear
6     // to have height and width = 0 but yet it renders content...
7     display: block;
8 }
9
10 // :host can take an extra css selector as parameter
11 // this will apply when your element is used like this:
12 // <template-element disabled></template-element>
13 // or with data-binding
14 // <template-element disabled$="{{isDisabled}}></template-element>
15 :host([disabled]) {
16     color: gray;
17 }
18
19 .some-class, [some-attribute], some-element {
20     // use custom css properties like this
21     // color can be defined by another developer, it defaults to blue
22     color: var(--my-custom-color, blue);
23 }
24
25 // another developer can do this now:
26 // template-element {
27 //     --my-custom-color: green;
28 // }
29
30 .some-class, [some-attribute], some-element {
31     // use custom css mixins like this
32     // another developer can now inject extra css at this point
33     @apply(--my-mixin-name);
34 }
35
36 // another developer can do this now:
37 // template-element {
38 //     --my-mixin-name: #'{{
39 //         background-color: green;
40 //         border-radius: 4px;
41 //         border: 1px solid gray;
42 //     }}';
43 }
```

In a generated element, you most probably won't need any of this code except the very first block (:host {display: block}). The rest serves as code examples. Notice that --my-custom-color and --my-mixin-name also appeared in the comment in template-element.html.

Now you should be able to run

```
1 cd src/html/elements
2 chmod +x new-element.js
3 ./new-element.js
4 name of the new element: my-new-element
5 creating new element <my-new-element>...
6 removing any .svn folders in my-new-element
7 Finished
```

And you will see a new folder my-new-element in /src/html/elements, ready for you to be developed further into whatever you want to build today.

Registering your elements in C++

When you open a web browser and navigate to your cell, your browser needs to be instructed to load your elements. AjaXell can do this for you, but you need to provide a list of elements.

Create a file /src/html/elements/elements.html Now, you don't have any elements yet. So this file will be empty for now. But here is an example how it would look like if you would have two elements *my-first-element* and *my-second-element*:

```
1 <link rel="import" href="my-first-element/my-first-element.html">
2 <link rel="import" href="my-second-element/my-second-element.html">
```

Now open the Cell.cc file and append the following line in the start of the Cell::init() function

```
1 void subsystemsupervisor::Cell::init()
2 {
3     getContext ()->addImport ("/<package-path>/html/elements/elements.html");
4     ...
5 }
```

<package-path> will depend on the name of your cell. For the subsystem supervisor the addImport line would be this

```
1 getContext ()->addImport ("/subsystem/supervisor/html/elements/elements.html");
```

1.2.2 The structure of a panel

A panel consists of C++ code rendering the data and one or more Polymer elements rendering the GUI.

A Polymer element consists of HTML, CSS, and JavaScript code. Each of these you can develop in a separate file.

C++

The main task of the C++ code is to provide the front-end code with data. This is something very important to realize. It will keep your code clean and easier to understand and change later on.

A minimal C++ panel looks like this:

```
1 #include "MyPanel.h"
2 #include "ajax/toolbox.h"
3 #include "ajax/PolymerElement.h"
4
5 #include "log4cplus/loggingmacros.h"
6 #include <iostream>
7
8 MyPanel::MyPanel( tsframework::CellAbstractContext* context, log4cplus::Logger& logger ) : tsframework::CellPanel(context,logger) {
9     logger_ = log4cplus::Logger::getInstance(logger.getName() + ".MyPanel");
10 }
11 MyPanel::~MyPanel() {
12     remove();
13 }
14
15 void MyPanel::layout(cgicc::Cgicc& cgi)
```

```
16     {
17         remove();
18         setEvent("user-clicked-button", ajax::Eventable::OnClick, this, &MyPanel::clicky);
19
20         ajax::PolymerElement* mypanel = new ajax::PolymerElement("my-panel");
21         add(mypanel);
22     }
23
24 void MyPanel::clicky(CGICC::Cgicc& cgi, std::ostream& out) {
25     out << "This was executed because you clicked the button";
26 }
```

This code outputs ‘<my-panel></my-panel>’ on page load. This is the name of our polymer element that renders the GUI for this panel.

It also registers a callback ‘user-clicked-button’. When the server receives that callback it will execute clicky() and return whatever is piped into ‘out’.

HTML

The main file of our Polymer element is the HTML file. It defines the visual structure and inserts our CSS and JavaScript code. It looks something like this:

```
1 <link rel="import" href="/extern/bower_components/polymer/polymer.html">
2 <link rel="import" href="/ts/common-elements/reset-css/reset-css.html">
3 <link rel="import" href="/extern/bower_components/paper-button/paper-button.html">
4 <!--
5 `<my-element>` is the Polymer element of the MyPanel panel.
6
7 It features a button the user can click. And when the user clicks this button
8 the server will say it clicked the button.
9
10 @authors me
11 -->
12 <dom-module id="my-element">
13     <template>
14         <link rel="import" type="css" href="css/my-element-min.css?__inline=true">
15         <style include="reset-css"></style>
16
17         <paper-button on-click="_doClickButton">Click me please</paper-button>
18         <ts-ajax id="ajax"
19             data="{{ajax_result}}"
20             callback="user-clicked-button"
21             handle-as="text"></ts-ajax>
22     </template>
23     <script src="javascript/my-element-min.js?__inline=true"></script>
24 </dom-module>
```

JavaScript

The JavaScript of your element is what makes your element spring to life. It adds interactivity to your element. A basic JavaScript file looks like this:

```
1 Polymer({
2     is: "my-element",
```

```

3   properties: {
4     ajax_result: {
5       type: String,
6       value: "you haven't clicked the button yet..."
7     }
8   },
9   _doClickButton: function() {
10    this.$.ajax.generateRequest();
11  }
12 });

```

SASS

You may have heard about CSS, it allows you to style your HTML markup. It is very powerful. But it misses some features. One big missing features is the ability to nest your selectors. Or sometimes you want to create for-loops. Maybe you would like to set a variable for a color you use a lot...

This is where SASS comes in (<http://sass-lang.com/>). SASS is CSS with superpowers. You write your styles using SASS, and the Grunt build tool will translate it to normal CSS for you.

Also note that we use another tool called autoprefixer (<https://css-tricks.com/autoprefixer/>). This will allow you to not worry about using vendor-prefixes (for example -webkit-transition vs transition) to keep your CSS compatible with older browsers.

A minimal CSS file looks like this:

```

1 :host {
2   display: block;
3 }

```

1.2.3 Demo 0: Hello World

Make the hello-world element

In your cell, run:

```

1 cd src/html/elements
2 ./new-element.js
3 name of the new element: hello-world
4 creating new element <hello-world>...
5 removing any .svn folders in hello-world
6 Finished

```

You now have a working *hello-world* element. We'll edit it soon.

Register the hello-world element

Edit src/html/elements/elements.html and add the following line

```

1 <link rel="import" href="hello-world/hello-world.html">

```

This will tell AjaXell to load our new element.

Edit the hello-world element

This is just a *hello, world!* example. We don't need much of the stuff our template generated.

Edit src/html/elements/hello-world/hello-world.html

```
1 <link rel="import" href="/extern/bower_components/polymer/polymer.html">
2 <link rel="import" href="/ts/common-elements/reset-css/reset-css.html">
3
4 <!--
5 `hello-world` is the simplest panel imaginable. It displays 'Hello, World!', 
6 nothing more.
7
8 Example:
9
10    <hello-world></hello-world>
11
12 @demo demo/index.html
13 -->
14 <dom-module id="hello-world">
15   <template>
16     <style include="reset-css"></style>
17     <link rel="stylesheet" type="text/css" href="css/hello-world-min.css?__inline=true
18   >
19
20     <h1>Hello, World!</h1>
21
22   </template>
23   <script src="javascript/hello-world-min.js?__inline=true"></script>
</dom-module>
```

Note that we didn't delete the reset-css include. This is recommended to do, *reset-css* provides us with some general css (fonts, theme colors, ...).

Now edit src/html/elements/hello-world/css/hello-world.scss

```
1 :host {
2   display: block;
3 }
```

It is recommended to always have a display directive in the :host{} section. This tells the browser how the element will behave inside a page. The most used are 'block', 'inline-block', and 'inline'. 'block' elements try to take as much width as possible, while 'inline' elements only take the width and height they need. 'inline-block' is an inline element that can still have a manually set width or height

Now edit src/html/elements/hello-world/javascript/hello-world.js

```
1 Polymer({
2   is: 'hello-world'
3 });
```

This is the minimal required javascript for a Polymer element. It only declares the existence of the *hello-world* element.

Now execute Grunt to build our new Polymer element.

```
1 cd src/html
2 grunt
```

Make the hello-world panel

Make a new c++ file /src/common/panels/HelloWorld.cc

```

1 #include "subsystem/supervisor/panels/HelloWorld.h"
2 #include "ajax/PolymerElement.h"
3
4 using namespace subsystempanels;
5 HelloWorld::HelloWorld( tsframework::CellAbstractContext* context, log4cplus::Logger& logger)
6 :tsframework::CellPanel(context, logger) {
7     logger_ = log4cplus::Logger::getInstance(logger.getName() + ".HelloWorld");
8 }
9
10 void HelloWorld::layout(cgicc::Cgicc& cgi) {
11     remove();
12     add(new ajax::PolymerElement("hello-world"));
13 }
```

The remove(); function clears any previously existing output buffer from the HelloWorld panel. If you remove that line and you request the panel twice, you get 2 *hello-world* elements.

Make the include/subsystem/supervisor/panels/HelloWorld.h file.

```

1 ifndef _subsystem_supervisor_panels_HelloWorld_h_
2 define _subsystem_supervisor_panels_HelloWorld_h_
3
4 include "ts/framework/CellPanel.h"
5 include "log4cplus/logger.h"
6 include "cgicc/Cgicc.h"
7
8 namespace subsystempanels {
9     class HelloWorld: public tsframework::CellPanel {
10         public:
11             HelloWorld(tsframework::CellAbstractContext* context, log4cplus::Logger& logger);
12             void layout(cgicc::Cgicc& cgi);
13     };
14 }
15 endif
```

Register the new class in the Makefile.

```

1 Sources= \
2     version.cc \
3     Cell.cc \
4     CellContext.cc \
5     Configuration.cc \
6     ...
7     panels/HelloWorld.cc \
8     ...
```

Now register your new panel in the menu so users can access it.

Edit src/common/Cell.cc

```

1 #include "subsystem/supervisor/panels/HelloWorld.h"
2 ...
3 void subsystemsupervisor::Cell::init() {
```

```
4 ...
5 tsframework::CellPanelFactory* panelF = getContext()->getPanelFactory();
6 ...
7 panelF->add<subsystempanels::HelloWorld>("HelloWorld");
```

Now you can compile your cell and you should see the HelloWorld panel in the menu under the ‘control-panels’ section.

The screenshot shows the Multicell Supervisor Control Panels interface. At the top, there is a navigation bar with the text "MULTICELL SUPERVISOR > Control Panels > HelloWorld". To the right of the navigation bar is a logo for "XDAQ" with a question mark icon. The main content area has a title "Hello, World!" in bold. On the left side, there is a sidebar with the following sections and items:

- Commands**
 - > Candela
 - > Default
 - > Monitor
 - > Supervisor
 - > TStore
 - > XDAQ
- Operations**
- Control Panels**
 - About
 - ButtonExample
 - Chart examples
 - Code editor example
 - Collapsible Demo
 - DialogExample
 - EmbeddingExample
 - Flexbox layout example
 - Form example
 - GtDbPanel
 - HelloWorld
 - InputFile
 - MeasurementExample

Also your element has created some documentation. Surf to <hostname>:<port>/<package-name>/html/index.html and you will see the package documentation for your cell. *hello-world* will be in there, and clicking it brings up the documentation for your *hello-world* element.

1.2.4 Demo 1: Ajax and data binding

Probably you would like your C++ code to supply some data to your panel. We will use the *ts-ajax* element in the *common-elements* package to retrieve our data, then we will use data-binding to display the data in our panel.

Make the data-binding element

In your cell, run:

```
1 cd src/html/elements
2 ./new-element.js
3 name of the new element: data-binding
4 creating new element <data-binding>...
5 removing any .svn folders in data-binding
6 Finished
```

Register the data-binding element

Edit src/html/elements/elements.html and add the following line

```
1 <link rel="import" href="data-binding/data-binding.html">
```

This will tell AjaXell to load our new element.

Edit the data-binding element

Edit src/html/elements/data-binding/data-binding.html

```
1 <link rel="import" href="/extern/bower_components/polymer/polymer.html">
2 <link rel="import" href="/ts/common-elements/reset-css/reset-css.html">
3 <link rel="import" href="/ts/common-elements/ts-ajax/ts-ajax.html">
4 <link rel="import" href="/extern/bower_components/paper-button/paper-button.html">
5
6 <!--
7 `data-binding` retrieves some data from C++ and displays it to the user
8
9 Example:
10
11 <data-binding></data-binding>
12
13 -->
14 <dom-module id="data-binding">
15   <template>
16     <style include="reset-css"></style>
17     <link rel="stylesheet" type="text/css" href="css/data-binding-min.css?__
18       inline=true">
19
20     <p>This is the data-binding example. It will fetch some data using the
21       `ts-ajax` element and display it here.</p>
22
23     <h1>example 1</h1>
24     <ts-ajax data="{{example1}}" callback="example1function" handle-as="text" auto></
25       ts-ajax>
26     <span>[[example1]]</span>
27
28     <h1>example 2</h1>
29     <ts-ajax id="example2" data="{{example2}}" callback="example2function" handle-as=
30       "json"></ts-ajax>
31     <paper-button raised on-click="doCallback">Do example2function callback</paper-
32       button><br>
33
34     <template is="dom-repeat" items="[[example2]]" as="item">
35       <span>[[item]]</span><br>
36     </template>
37
38   </template>
39   <script src="javascript/data-binding-min.js?__inline=true"></script>
40 </dom-module>
```

Note the {{...}} and [[...]] code. This is our data binding code. Consider the highlighted lines. Line 23 tells Polymer to link the *data* variable from ts-ajax with our own *example1* variable. This way, if ts-ajax changes its *data* variable our own *example1* variable will change too.

When we use the `{...}` syntax this change goes both ways. `[...]` goes one way only. Use the latter to display some final result as we did in line 24, where we don't anticipate a source of change.

Now edit `src/html/elements/data-binding/css/data-binding.scss`

```
1 :host {  
2   display: block;  
3 }
```

Now edit `src/html/elements/data-binding/javascript/data-binding.js`

```
1 Polymer({  
2   is: 'data-binding',  
3  
4   properties: {  
5     example1: {  
6       type: String,  
7       value: "no data from C++ yet..."  
8     },  
9     example2: {  
10       type: Array,  
11       value: function() {  
12         return ["no data from C++ yet..."];  
13       }  
14     }  
15   },  
16  
17   doCallback: function() {  
18     // this.$ is a shorthand selector,  
19     // it allows us to select an element in our template by id  
20     this.$.example2.generateRequest();  
21   }  
22 });
```

Now execute Grunt to build our new Polymer element.

```
1 cd src/html  
2 grunt
```

Make the data-binding panel

Make a new c++ file `/src/common/panels/DataBinding.cc`

```
1 #include "subsystem/supervisor/panels/DataBinding.h"  
2 #include "ajax/PolymerElement.h"  
3 #include "json/json.h"  
4 #include <sstream>  
5  
6 using namespace subsystempanels;  
7 DataBinding::DataBinding( tsframework::CellAbstractContext* context, log4cplus::  
8   ~Logger& logger)  
9 :tsframework::CellPanel(context, logger) {  
10   logger_ = log4cplus::Logger::getInstance(logger.getName() + ".DataBinding");  
11 }  
12  
13 void DataBinding::layout(cgicc::Cgicc& cgi) {  
14   remove();
```

```

14     setEvent("example1function", ajax::Eventable::OnClick, this, &DataBinding::
15     ↪example1);
16     setEvent("example2function", ajax::Eventable::OnClick, this, &DataBinding::
17     ↪example2);
18     add(new ajax::PolymerElement("data-binding"));
19 }
20
21 void DataBinding::example1(CGICC::Cgicc& cgi, std::ostream& out) {
22     out << "This text is generated using C++!";
23 }
24
25 void DataBinding::example2(CGICC::Cgicc& cgi, std::ostream& out) {
26     Json::Value root(Json::arrayValue);
27     for (size_t i = 0; i < 10; i++) {
28         std::stringstream ss;
29         ss << "This is text " << i << " generated by C++";
30         root.append(ss.str());
31     }
32     out << root;
33 }
```

Notice that in the HTML code earlier we specified callback="example1function" in one of the ts-ajax elements. Notice the #include "json/json.h" line. We import the jsoncpp library this way in order to create an array of strings in example2().

Edit your Makefile to add jsoncpp as a dependency.

```
1 DependentLibraries = tsframework tsajaxell ... jsoncpp
```

Make the include/subsystem/supervisor/panels/DataBinding.h file.

```

1 #ifndef _subsystem_supervisor_panels_DataBinding_h_
2 #define _subsystem_supervisor_panels_DataBinding_h_
3
4 #include "ts/framework/CellPanel.h"
5 #include "log4cplus/logger.h"
6 #include "cgicc/Cgicc.h"
7
8 namespace subsystempanels {
9     class DataBinding: public tsframework::CellPanel {
10     public:
11         DataBinding(tsframework::CellAbstractContext* context, log4cplus::Logger& ↪
12         logger);
13         void layout(CGICC::Cgicc& cgi);
14     private:
15         void example1(CGICC::Cgicc& cgi, std::ostream& out);
16         void example2(CGICC::Cgicc& cgi, std::ostream& out);
17     };
18 }
#endiff
```

Register the new class in the Makefile.

```
1 Sources=\
2   version.cc\
3   Cell.cc\
4   CellContext.cc\
5   Configuration.cc\
```

```
6     ...
7     panels/DataBinding.cc \
8     ...
```

Now register your new panel in the menu so users can access it.

Edit src/common/Cell.cc

```
1 #include "subsystem/supervisor/panels/DataBinding.h"
2 ...
3 void subsystemsupervisor::Cell::init() {
4     ...
5     tsframework::CellPanelFactory* panelF = getContext()->getPanelFactory();
6     ...
7     panelF->add<subsystempanels::DataBinding>("ts-ajax and data-binding");
```

That's right, you can have spaces in your menu names.

Now you can compile your cell and you should see the “ts-ajax and data-binding” panel in the menu under the ‘control-panels’ section.

This is the data-binding example. It will fetch some data using the `ts-ajax` element and display it here.

example 1

This text is generated using C++!

example 2

DO EXAMPLE2FUNCTION CALLBACK

This is text 0 generated by C++
This is text 1 generated by C++
This is text 2 generated by C++
This is text 3 generated by C++
This is text 4 generated by C++
This is text 5 generated by C++
This is text 6 generated by C++
This is text 7 generated by C++
This is text 8 generated by C++
This is text 9 generated by C++

Also your element has created some documentation. Surf to <hostname>:<port>/<package-name>/html/index.html and you will see the package documentation for your cell. *data-binding* will be in there, and clicking it brings up the documentation for your *data-binding* element.

Be sure to check out the documentation for the *ts-ajax* element at <hostname>:<port>/ts/common-elements/html/index.html

1.2.5 Demo 2: The flexbox layout

To setup a layout of your panels you can use the flexbox layout system. This is a set of new CSS directives that seeks to make designing layouts much simpler. It would deprecate the use of *float: left* and other nonsense.

Make the flexbox-layout element

In your cell, run:

```
1 cd src/html/elements
2 ./new-element.js
3 name of the new element: flexbox-layout
4 creating new element <flexbox-layout>...
5 removing any .svn folders in flexbox-layout
6 Finished
```

Register the flexbox-layout element

Edit src/html/elements/elements.html and add the following line

```
1 <link rel="import" href="flexbox-layout/flexbox-layout.html">
```

This will tell AjaXell to load our new element.

Edit the flexbox-layout element

Edit src/html/elements/flexbox-layout/flexbox-layout.html

```
1 <link rel="import" href="/extern/bower_components/polymer/polymer.html">
2 <link rel="import" href="/ts/common-elements/reset-css/reset-css.html">
3 <link rel="import" href="/ts/common-elements/ts-colors/ts-colors.html">
4 <link rel="import" href="/ts/common-elements/iron-flex-layout-attributes/iron-flex-
5   ↴layout-attributes.html">
6
7 <!--
8 `flexbox-layout` is a demo showcasing the capabilities of iron-flex-layout-attributes.
9 @demo demo/index.html
10 -->
11 <dom-module id="flexbox-layout">
12   <template>
13     <style include="reset-css"></style>
14     <style include="ts-colors"></style>
15
16     <!-- this will allow you to use flexbox easily -->
17     <!-- surf to /ts/common-elements/iron-flex-layout-attributes/index.html -->
18     <style include="iron-flex-layout-attributes"></style>
19
20     <link rel="stylesheet" type="text/css" href="css/flexbox-layout-min.css?__
21       ↴inline=true">
22   </template>
23   <script src="javascript/flexbox-layout-min.js?__inline=true"></script>
24 </dom-module>
```

We'll add more stuff as we go along. Note the import for *ts-colors*. It will allow us to do easily add colors to our stuff. It implements the material design color palette (<https://www.google.com/design/spec/style/color.html#color-color-palette>).

The following code will contain a blue box:

```
1 <div blue-400>this will have a blue background</div>
2 <div blue-100>this will have a light blue background</div>
```

Now edit src/html/elements/flexbox-layout/css/flexbox-layout.scss

```
1 :host {
2   display: block;
3 }
4
5 [square] {
6   width: 100px;
7   height: 100px;
8   margin: 1em;
9   img {
10     width: 100px;
11     height: 100px;
12   }
13 }
```

Now edit src/html/elements/flexbox-layout/javascript/flexbox-layout.js

```
1 Polymer({
2   is: "flexbox-layoutexample"
3 });
```

Flexbox layouts

Horizontal layout Add the following code to the HTML template

```
1 <h2>Horizontal layout</h2>
2 <div horizontal layout blue-200>
3   <div square blue-100></div>
4   <div square blue-100></div>
5   <div square blue-100></div>
6 </div>
```

It will render this:



Horizontal layout with flex The flex attribute instructs an element to take as much space as possible in the direction of the layout (horizontal or vertical).

If there are multiple elements with the flex attribute the available space will be divided equally between them.

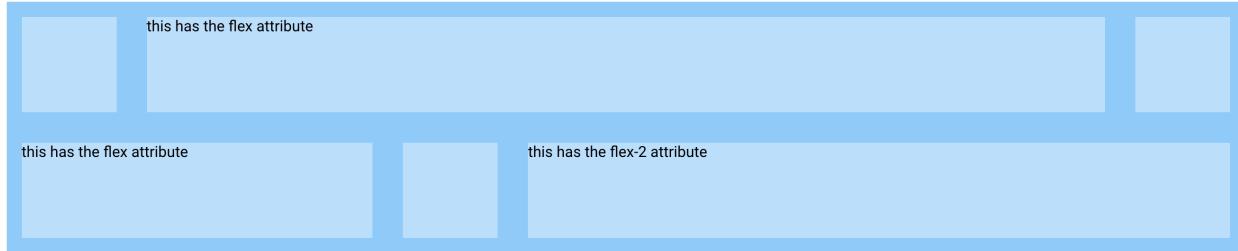
There are also the flex-2, flex-3, ..., flex-12 attributes. When multiple elements in the same layout have flex attributes, this will assign a greater weight to the elements. An element with the flex-2 attribute will be twice as big as the element with the flex attribute in the same layout.

Add the following code to the HTML template

```

1 <div horizontal layout blue-200>
2   <div square blue-100></div>
3   <div square blue-100 flex>this has the flex attribute</div>
4   <div square blue-100></div>
5 </div>
6 <div horizontal layout blue-200>
7   <div square blue-100 flex>this has the flex attribute</div>
8   <div square blue-100></div>
9   <div square blue-100 flex-2>this has the flex-2 attribute</div>
10 </div>
```

It will render this:



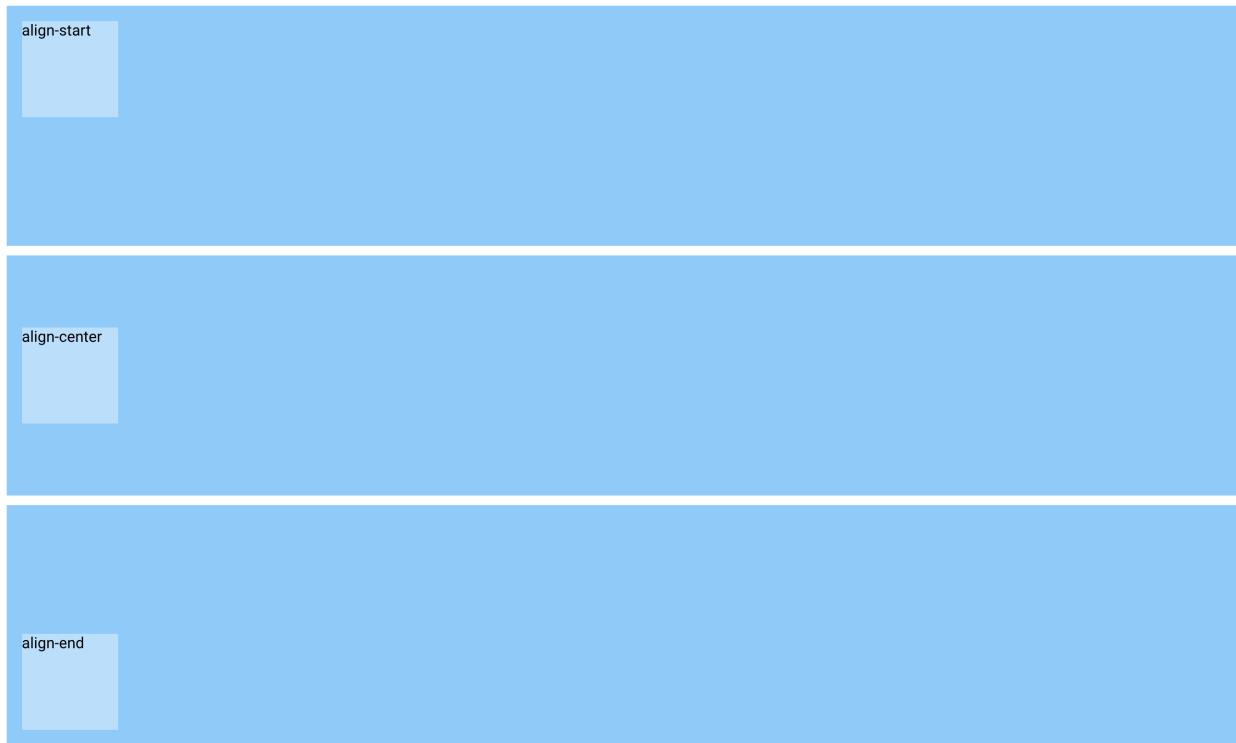
vertical alignment When you have for example a horizontal layout you might like to also control how the elements in the layout behave vertically. Same goes the other way, you might like to control horizontal behavior of elements in a vertical layout.

Add the following code to the HTML template

```

1 <div horizontal layout blue-200 align-start style="height: 250px; margin: 10px;">
2   <div square blue-100>align-start</div>
3 </div>
4 <div horizontal layout blue-200 align-center style="height: 250px; margin: 10px;">
5   <div square blue-100>align-center</div>
6 </div>
7 <div horizontal layout blue-200 align-end style="height: 250px; margin: 10px;">
8   <div square blue-100>align-end</div>
9 </div>
```

It will render this:



flex alignment You can also specify options for behavior of elements in the layout direction. For example you might want to horizontally center a set of horizontally aligned elements.

Add the following code to the HTML template

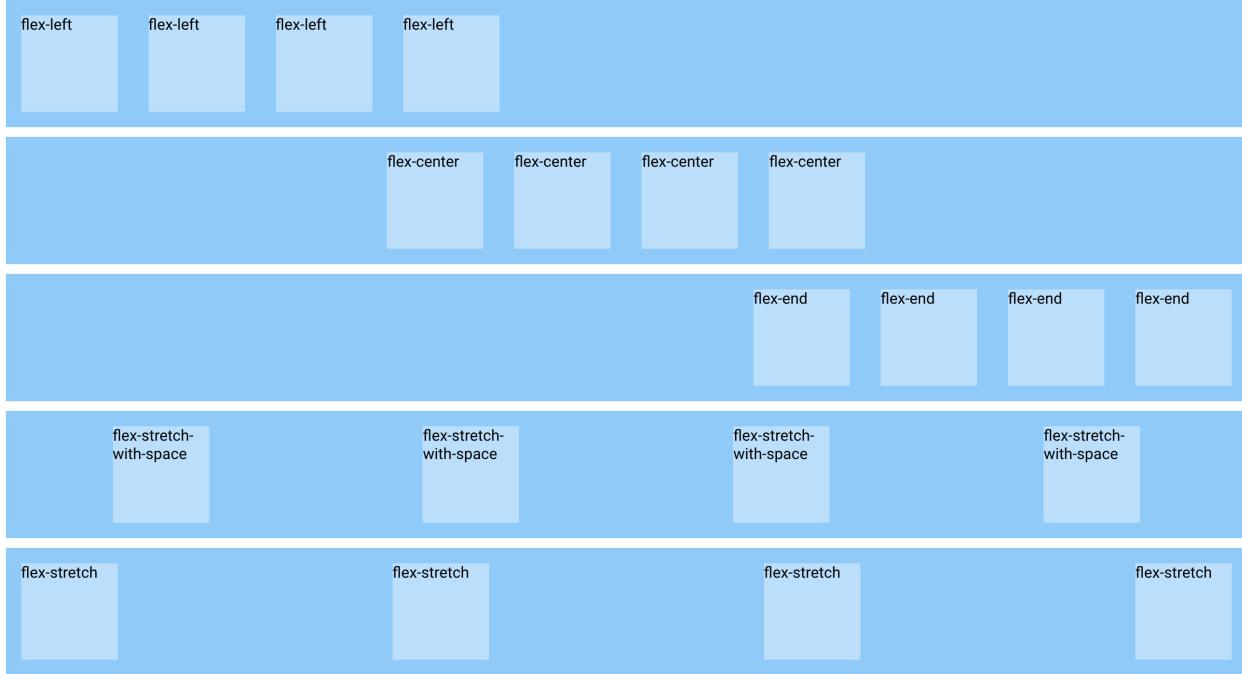
```
1 <div horizontal layout blue-200 flex-left style="margin: 10px;">>
2   <div square blue-100>flex-left</div>
3   <div square blue-100>flex-left</div>
4   <div square blue-100>flex-left</div>
5   <div square blue-100>flex-left</div>
6 </div>
7 <div horizontal layout blue-200 flex-center style="margin: 10px;">>
8   <div square blue-100>flex-center</div>
9   <div square blue-100>flex-center</div>
10  <div square blue-100>flex-center</div>
11  <div square blue-100>flex-center</div>
12 </div>
13 <div horizontal layout blue-200 flex-end style="margin: 10px;">>
14   <div square blue-100>flex-end</div>
15   <div square blue-100>flex-end</div>
16   <div square blue-100>flex-end</div>
17   <div square blue-100>flex-end</div>
18 </div>
19 <div horizontal layout blue-200 flex-stretch-with-space style="margin: 10px;">>
20   <div square blue-100>flex-stretch-with-space</div>
21   <div square blue-100>flex-stretch-with-space</div>
22   <div square blue-100>flex-stretch-with-space</div>
23   <div square blue-100>flex-stretch-with-space</div>
24 </div>
25 <div horizontal layout blue-200 flex-stretch style="margin: 10px;">>
26   <div square blue-100>flex-stretch</div>
```

```

27 <div square blue-100>flex-stretch</div>
28 <div square blue-100>flex-stretch</div>
29 <div square blue-100>flex-stretch</div>
30 </div>

```

It will render this:



wrapped content blocks

You can instruct a layout to break its set of elements into multiple lines.

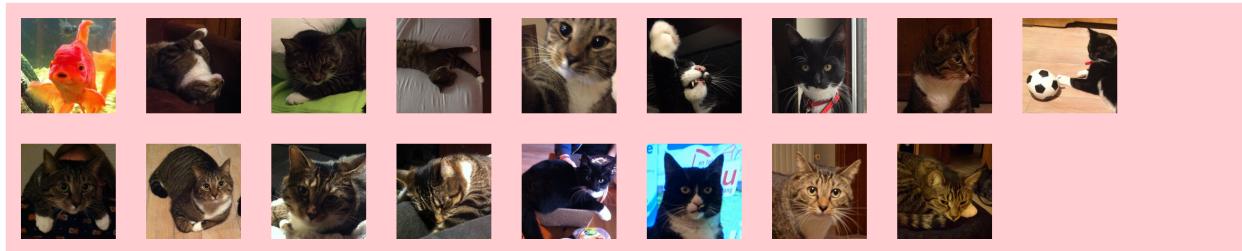
Add the following code to the HTML template

```

1 <div horizontal layout wrap red-100>
2   <div square red-300></div>
3   <div square red-300></div>
4   <div square red-300></div>
5   <div square red-300></div>
6   <div square red-300></div>
7   <div square red-300></div>
8   <div square red-300></div>
9   <div square red-300></div>
10  <div square red-300></div>
11  <div square red-300></div>
12  <div square red-300></div>
13  <div square red-300></div>
14  <div square red-300></div>
15  <div square red-300></div>
16  <div square red-300></div>
17  <div square red-300></div>
18  <div square red-300></div>
19 </div>

```

It will render this:

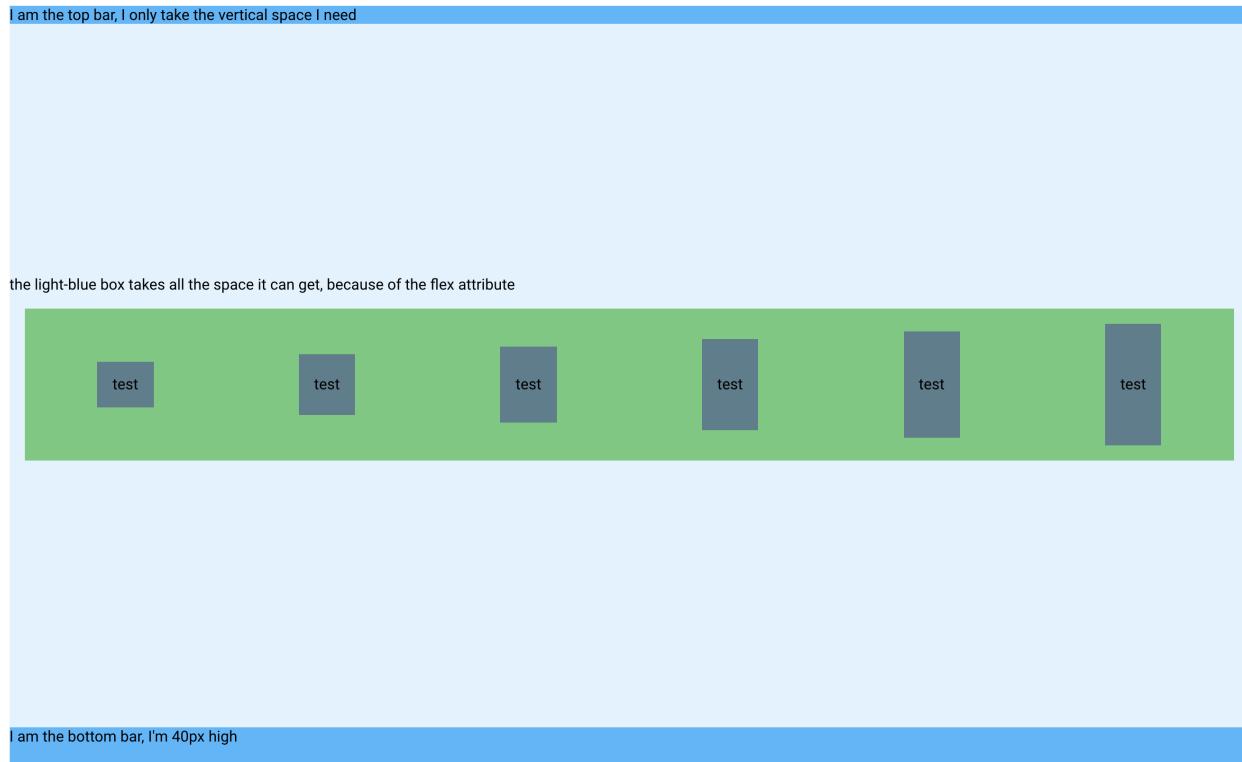


header and footer, vertically centered content

This layout generates a header and a footer, the content has the flex attribute and takes the rest of the vertical space.

```
1 <div vertical layout style="height:800px;">
2   <div blue-300>I am the top bar, I only take the vertical space I need</div>
3   <div blue-50 flex vertical layout flex-center>
4     <span>the light-blue box takes all the space it can get, because of the flex_
5       ↪attribute</span>
6
7     <div horizontal layout
8       green-300
9       flex-stretch-with-space
10      align-center
11      style="height: 10em; margin: 1em;">
12       <span blue-grey-500 style="margin:1em;padding:1em;line-height:1em;">test</span>
13       <span blue-grey-500 style="margin:1em;padding:1em;line-height:2em;">test</span>
14       <span blue-grey-500 style="margin:1em;padding:1em;line-height:3em;">test</span>
15       <span blue-grey-500 style="margin:1em;padding:1em;line-height:4em;">test</span>
16       <span blue-grey-500 style="margin:1em;padding:1em;line-height:5em;">test</span>
17       <span blue-grey-500 style="margin:1em;padding:1em;line-height:6em;">test</span>
18     </div>
19   </div>
20   <div blue-300 style="height: 40px;">I am the bottom bar, I'm 40px high</div>
21 </div>
```

It will render this:



Impressively complex layout 1

This code:

```

1  <!-- big box, children will be put next to each other --&gt;
2  &lt;div horizontal layout style="height: 300px;"&gt;
3      &lt;div blue-400&gt;left&lt;/div&gt;
4
5      &lt;!-- second big box, children will be put on top of each other --&gt;
6      &lt;div vertical layout flex&gt;
7          &lt;div blue-300&gt;top&lt;/div&gt;
8
9          &lt;div horizontal layout flex&gt;
10
11             &lt;div vertical layout flex&gt;
12
13                 &lt;div horizontal layout flex&gt;
14                     &lt;div blue-200&gt;left&lt;/div&gt;
15                     &lt;div blue-100 flex&gt;left&lt;/div&gt;
16                     &lt;div blue-50 flex-5&gt;
17                         &lt;p&gt;note that this box is 5 times larger than the box to the ↴
18                         left, thanks to the flex-5 attribute&lt;/p&gt;
19                         &lt;p&gt;&lt;a href="https://www.google.be"&gt;Google&lt;/a&gt;&lt;/p&gt;
20                         &lt;p&gt;Lorem ipsum dolor sit amet, consectetur adipisicing elit, ↴
    ↴sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim ↴
    ↴veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo ↴
    ↴consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum ↴
    ↴dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, ↴
    ↴sunt in culpa qui officia deserunt mollit anim id est laborum.&lt;/p&gt;
21             &lt;/div&gt;
22         &lt;/div&gt;
23     &lt;/div&gt;
24 </pre>

```

```

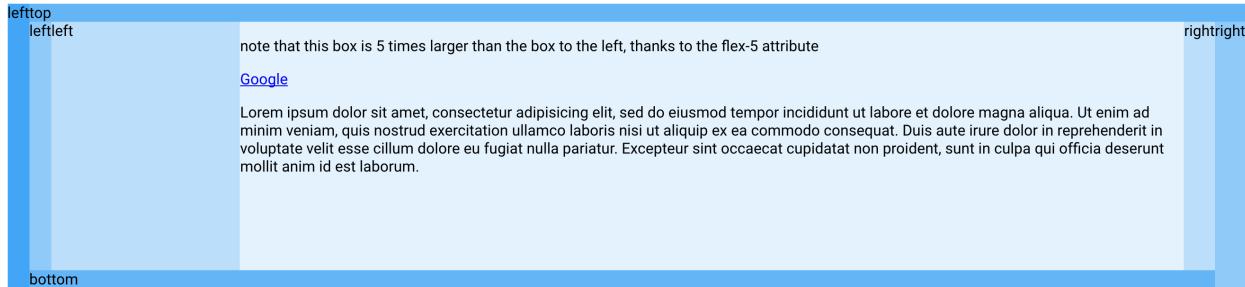
21      <div blue-100>right</div>
22    </div>

23
24      <div blue-300>bottom</div>
25    </div>

26
27      <div blue-200>right</div>
28    </div>

29
30  </div>
31
32</div>
```

Will render this:



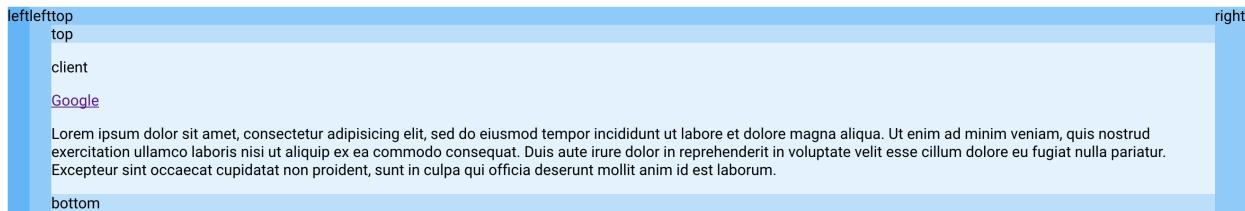
Impossibly complex layout 2

This code:

```

1  <div horizontal layout>
2    <div blue-300>left</div>
3    <div blue-200>left</div>
4    <div vertical layout>
5      <div blue-200>top</div>
6      <div blue-100>top</div>
7      <div blue-50>
8        <p>client</p>
9        <p><a href="#">Google</a></p>
10       <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.</p>
11     </div>
12     <div blue-100>bottom</div>
13   </div>
14   <div blue-200>right</div>
15 </div>
```

Will render this:



Now execute Grunt to build our new Polymer element.

```
1 cd src/html
2 grunt
```

Make the flexbox-layout panel

Make a new c++ file /src/common/panels/FlexboxLayout.cc

```
1 #include "subsystem/supervisor/panels/FlexboxLayout.h"
2 #include "ajax/PolymerElement.h"
3
4 using namespace subsystempanels;
5 FlexboxLayout::FlexboxLayout( tsframework::CellAbstractContext* context, log4cplus::
6     →Logger& logger)
7 :tsframework::CellPanel(context, logger) {
8     logger_ = log4cplus::Logger::getInstance(logger.getName() + ".FlexboxLayout");
9 }
10 void FlexboxLayout::layout(cgicc::Cgicc& cgi) {
11     remove();
12     add(new ajax::PolymerElement("flexbox-layout"));
13 }
```

Make the include/subsystem/supervisor/panels/FlexboxLayout.h file.

```
1 ifndef _subsystem_supervisor_panels_FlexboxLayout_h_
2 define _subsystem_supervisor_panels_FlexboxLayout_h_
3
4 include "ts/framework/CellPanel.h"
5 include "log4cplus/logger.h"
6 include "cgicc/Cgicc.h"
7
8 namespace subsystempanels {
9     class FlexboxLayout: public tsframework::CellPanel {
10         public:
11             FlexboxLayout(tsframework::CellAbstractContext* context, log4cplus::Logger&_
12             →logger);
13             void layout(cgicc::Cgicc& cgi);
14     };
15 }
#endif
```

Register the new class in the Makefile.

```
1 Sources=\
2     version.cc\
3     Cell.cc\
4     CellContext.cc\
```

```
5 Configuration.cc \
6 ...
7 panels/FlexboxLayout.cc \
8 ...
```

Now register your new panel in the menu so users can access it.

Edit src/common/Cell.cc

```
1 #include "subsystem/supervisor/panels/FlexboxLayout.h"
2 ...
3 void subsystemsupervisor::Cell::init() {
4 ...
5     tsframework::CellPanelFactory* panelF = getContext()->getPanelFactory();
6 ...
7     panelF->add<subsystempanels::FlexboxLayout>("Flexbox layout examples");
```

Now you can compile your cell and you should see the “Flexbox layout examples” panel in the menu under the ‘control-panels’ section.

Also your element has created some documentation. Surf to <hostname>:<port>/<package-name>/html/index.html and you will see the package documentation for your cell. *flexbox-layout* will be in there, and clicking it brings up the documentation for your *flexbox-layout* element.

1.2.6 Demo 3: Sending data to the server

The *ts-ajax* element can also be used to send data back to the server. To demonstrate this we’ll build a simple form.

Make the form-example element

In your cell, run:

```
1 cd src/html/elements
2 ./new-element.js
3 name of the new element: form-example
4 creating new element <form-example>...
5 removing any .svn folders in  form-example
6 Finished
```

Register the form-example element

Edit src/html/elements/elements.html and add the following line

```
1 <link rel="import" href="form-example/form-example.html">
```

This will tell AjaXell to load our new element.

Edit the form-example element

Edit src/html/elements/form-example/form-example.html
/src/html/elements/form-example/form-example.html

```

1 <link rel="import" href="/extern/bower_components/polymer/polymer.html">
2 <link rel="import" href="/ts/common-elements/reset-css/reset-css.html">
3 <link rel="import" href="/ts/common-elements/iron-flex-layout-attributes/iron-flex-
4   ↪layout-attributes.html">
5 <link rel="import" href="/extern/bower_components/paper-material/paper-material.html">
6 <link rel="import" href="/extern/bower_components/paper-input/paper-input.html">
7 <link rel="import" href="/extern/bower_components/paper-input/paper-textarea.html">
8 <link rel="import" href="/extern/bower_components/iron-icon/iron-icon.html">
9 <link rel="import" href="/extern/bower_components/iron-icons/iron-icons.html">
10 <link rel="import" href="/extern/bower_components/paper-checkbox/paper-checkbox.html">
11 <link rel="import" href="/extern/bower_components/paper-radio-button/paper-radio-
12   ↪button.html">
13 <link rel="import" href="/extern/bower_components/paper-radio-group/paper-radio-group.
14   ↪html">
15 <link rel="import" href="/extern/bower_components/paper-slider/paper-slider.html">
16 <link rel="import" href="/extern/bower_components/paper-fab/paper-fab.html">
17 <link rel="import" href="/extern/bower_components/paper-toggle-button/paper-toggle-
18   ↪button.html">
19 <link rel="import" href="/ts/common-elements/ts-ajax/ts-ajax.html">
20 <link rel="import" href="complex-input.html">

21 <!-- for animations -->
22 <link rel="import" href="/extern/bower_components/neon-animation/neon-shared-element-
23   ↪animatable-behavior.html">
24 <link rel="import" href="/extern/bower_components/neon-animation/neon-animatable-
25   ↪behavior.html">
26 <link rel="import" href="/extern/bower_components/neon-animation/neon-animation-
27   ↪runner-behavior.html">
28 <link rel="import" href="/extern/bower_components/neon-animation/animations/fade-in-
29   ↪animation.html">
30 <link rel="import" href="/extern/bower_components/neon-animation/animations/cascaded-
31   ↪animation.html">
32 <link rel="import" href="/extern/bower_components/neon-animation/animations/transform-
33   ↪animation.html">

34 <!--
35 `<polymer-formexample>` is an element that demonstrates the most common things found
36 in a form made in a polymer fashion.
37
38 It uses flexbox to setup the layout. This layout is set via attribute selectors.
39 with a media query these attributes can be changed, and so achieve a responsive_
40   ↪layout
41
42 data binding is used to:
43   - bind the media query and the attributes
44   - bind some components together to make the form interactive
45     example: the checkboxes aren't visible if the toggle-button isn't checked
46

@authors Glenn Dirkx
-->
<dom-module id="form-example">
  <template>
    <style include="reset-css"></style>
    <style include="iron-flex-layout-attributes"></style>
    <link rel="stylesheet" type="text/css" href="css/form-example-min.css?__
      ↪inline=true">
```

```

47 <ts-ajax id="ajax"
48   callback="submit"
49   data="{{response}}"
50   parameters='["sometextinput", "defaultvaluetext", "password", "textarea"
51   ↪", "charcounter", "charcounter10", "letters", "letters2", "username", "ssn",
52   ↪"likespizza", "withcheese", "withsalami", "withpineapple", "withonion", "withkebab
53   ↪", "favoritetvstuff", "slider1", "slider2", "slider3", "slider4"]'
54     sometextinput={{text1}}
55     defaultvaluetext="{{text2}}"
56     password="{{text3}}"
57     textarea="{{text4}}"
58     charcounter="{{text6}}"
59     charcounter10="{{text7}}"
60     letters="{{text8}}"
61     letters2="{{text9}}"
62     username="{{text10}}"
63     ssn="{{text11}}"
64     likespizza="{{likesPizza}}"
65     withcheese="{{withCheese}}"
66     withsalami="{{withSalami}}"
67     withpineapple="{{withPineapple}}"
68     withonion="{{withOnion}}"
69     withkebab="{{withKebab}}"
70     favoritetvstuff="{{favoriteTVstuff}}"
71     slider1="{{slider1}}"
72     slider2="{{slider2}}"
73     slider3="{{slider3}}"
74     slider4="{{slider4}}">></ts-ajax>
75   <!-- notice the single &amp; double quotes in the parameters variable
76   do yourself a favor and do not use capital letters in parameter names --&gt;
77
78 &lt;section horizontal layout&gt;
79
80   &lt;paper-material elevation="1" flex&gt;
81     &lt;paper-input label="some text input"
82       value="{{text1}}"&gt;&lt;/paper-input&gt;
83
84     &lt;paper-input label="text input with default value"
85       value="{{text2}}"&gt;&lt;/paper-input&gt;
86
87     &lt;paper-input label="type your current CERN password here"
88       type="password"
89       value="{{text3}}"&gt;&lt;/paper-input&gt;
90
91     &lt;paper-textarea label="this is actually a text area, it will grow as needed"
92       value="{{text4}}"&gt;&lt;/paper-textarea&gt;
93
94     &lt;paper-input label="this one is disabled"
95       disabled
96       value="{{text5}}"&gt;&lt;/paper-input&gt;
97
98     &lt;paper-input label="simple character counter"
99       char-counter
100      value="{{text6}}"&gt;&lt;/paper-input&gt;
101
102     &lt;paper-input label="input with at most 10 characters"
103       char-counter
104       maxlength="10"&gt;</pre>

```

```

102                               value="{{text7}}">></paper-input>
103
104 <paper-input label="this input requires letters only"
105   auto-validate pattern="[a-zA-Z]*"
106   error-message="letters only!"
107   value="{{text8}}">></paper-input>
108
109 <paper-input label="this input will only let you type letters"
110   auto-validate
111   allowed-pattern="[a-zA-Z]"
112   value="{{text9}}">></paper-input>
113
114 <paper-input label="username">
115   value="{{text10}}>
116   <iron-icon icon="mail" prefix></iron-icon>
117   <div suffix>@email.com</div>
118 </paper-input>
119
120 <paper-input-container always-float-label auto-validate attr-for-value="value
121   " >
122   <label>Social Security Number</label>
123   <complex-input class="paper-input-input" value="{{text11}}></complex-input>
124   <paper-input-error>SSN invalid!</paper-input-error>
125 </paper-input-container>
126
127 </paper-material>
128
129 <div vertical layout flex>
130
131   <paper-material elevation="1" flex vertical layout>
132     <paper-toggle-button checked="{{likesPizza}}>I like pizza</paper-toggle-
133   <button>
134     <template is="dom-if" if="{{likesPizza}}>
135       <paper-checkbox checked="{{withCheese}}>with cheese</paper-checkbox>
136       <paper-checkbox checked="{{withSalami}}>with salami</paper-checkbox>
137       <paper-checkbox checked="{{withPineapple}}>with pineapple</paper-
138     <checkbox>
139       <template is="dom-if" if="{{withPineapple}}" restamp="true">
140         <p pineapple-note>You have terrible taste</p>
141       </template>
142       <paper-checkbox checked="{{withOnion}}>with onion</paper-checkbox>
143       <paper-checkbox checked="{{withKebab}}>with kebab</paper-checkbox>
144     </template>
145   </paper-material>
146
147   <paper-material elevation="1">
148     <paper-radio-group selected="{{favoriteTVstuff}}> vertical layout>
149       <paper-radio-button name="starwars">Star Wars</paper-radio-button>
150       <paper-radio-button name="startrek">Star Trek</paper-radio-button>
151       <paper-radio-button name="drwho">Dr Who</paper-radio-button>
152       <paper-radio-button name="jp">Jurassic Park</paper-radio-button>
153       <paper-radio-button name="td">Tenacious D in The Pick of Destiny</paper-
154     <radio-button>
155       </paper-radio-group>
156   </paper-material>
157
158 </div>

```

```
156 </section>
157
158 <paper-material elevation="1">
159   <paper-slider name="slider 1"
160     value="{{slider1}}></paper-slider>
161   <paper-slider name="slider 2"
162     value="{{slider2}}"
163     max="100"
164     editable></paper-slider>
165   <paper-slider name="slider 3"
166     pin
167     value="{{slider3}}></paper-slider>
168   <paper-slider name="slider 4"
169     max-markers="100"
170     pin
171     snaps
172     max="300"
173     step="20"
174     value="{{slider4}}></paper-slider>
175 </paper-material>
176
177 <paper-fab icon="send" title="send" on-click="submit"></paper-fab>
178
179 <h1>server response:</h1>
180
181 <template is="dom-repeat" items="[[response]]" as="item">
182   <p>[[item.name]] = [[item.value]]</p>
183 </template>
184
185 </template>
186 <script src="javascript/form-example-min.js?__inline=true"></script>
187 </dom-module>
```

Now edit /src/html/elements/form-example/css/form-example.scss

/src/html/elements/form-example/css/form-example.scss

```
1 :host {
2   display: block;
3 }
4
5 paper-material {
6   margin: 1em;
7   padding: 1em;
8 }
9
10 paper-fab {
11   width: 5em;
12   height: 5em;
13   position: absolute;
14   bottom: 1em;
15   right: 1em;
16 }
17
18 p[pineapple-note] {
19   border: 3px solid red;
20   padding: 5px;
21 }
```

Now edit /src/html/elements/form-example/javascript/form-example.js

/src/html/elements/form-example/javascript/form-example.js

```

56         name: 'fade-in-animation',
57         node: this,
58         timing: {duration: 500}
59     } ],
60     'cascade':[ {
61         name: 'cascaded-animation',
62         animation: 'transform-animation',
63         transformFrom: 'translateY(100%)',
64         transformTo: 'none'
65     }]
66 }
67 }
68 }
69 },
70
71 submit: function() {
72     this.$.ajax.generateRequest();
73 },
74
75
76 attached: function() {
77     // creates the animation during panel-load
78     this.async(function() {
79         var nodeList = Polymer.dom(this.root).querySelectorAll('paper-material');
80         this.animationConfig['cascade'][0].nodes = Array.prototype.slice.
81         ↪call(nodeList);
82         this.style.display = 'block';
83         this.playAnimation('fadein');
84         this.playAnimation('cascade');
85     });
86 },
87
88 behaviors: [
89     Polymer.NeonSharedElementAnimatableBehavior,
90     Polymer.NeonAnimatableBehavior,
91     Polymer.NeonAnimationRunnerBehavior
92 ]
93 );

```

Now execute Grunt to build our new Polymer element.

```

1 cd src/html
2 grunt

```

Make the form-example panel

Make a new c++ file /src/common/panels/FormExample.cc

/src/common/panels/FormExample.cc

```

1 #include "subsystem/supervisor/panels/FormExample.h"
2 #include "ajax/PolymerElement.h"
3 #include "ajax/toolbox.h"
4 #include "json/json.h"
5
6 using namespace subsystempanels;
7 FormExample::FormExample( tsframework::CellAbstractContext* context, log4cplus::
8 ↪Logger& logger)

```

```

8 :tsframework::CellPanel(context,logger) {
9     logger_ = log4cplus::Logger::getInstance(logger.getName() +".FormExample");
10 }
11 FormExample::~FormExample() {
12     remove();
13 }
14
15 void FormExample::layout(cgicc::Cgicc& cgi) {
16     remove();
17     setEvent("submit", ajax::Eventable::OnClick, this, &FormExample::submit);
18     add(new ajax::PolymerElement("form-example"));
19 }
20
21 void FormExample::submit(cgicc::Cgicc& cgi, std::ostream& out) {
22     std::map<std::string, std::string> values(ajax::toolbox::getSubmittedValues(cgi));
23
24     Json::Value root(Json::arrayValue);
25
26     for(std::map<std::string, std::string>::iterator i(values.begin()); i != values.
27     end(); ++i) {
28         Json::Value someresult;
29         someresult["name"] = i->first;
30         someresult["value"] = i->second;
31         root.append(someresult);
32     }
33
34     out << root;
35 }
```

Make the /include/subsystem/supervisor/panels/FormExample.h file.

/include/subsystem/supervisor/panels/FormExample.h

```

1 #ifndef _subsystem_supervisor_panels_FormExample_h_
2 #define _subsystem_supervisor_panels_FormExample_h_
3
4 #include "ts/framework/CellPanel.h"
5 #include "log4cplus/logger.h"
6 #include "cgicc/Cgicc.h"
7
8 namespace subsystempanels {
9     class FormExample: public tsframework::CellPanel {
10     public:
11         FormExample(tsframework::CellAbstractContext* context, log4cplus::Logger& logger);
12         ~FormExample();
13         void layout(cgicc::Cgicc& cgi);
14         private:
15         void submit(cgicc::Cgicc& cgi, std::ostream& out);
16     };
17 }
18 #endif
```

Register the new class in the Makefile.

```

1 Sources=\
2     version.cc\
3     Cell.cc\
```

```
4   CellContext.cc \
5   Configuration.cc \
6   ...
7   panels/FormExample.cc \
8   ...
```

Now register your new panel in the menu so users can access it.

Edit src/common/Cell.cc

```
1 #include "subsystem/supervisor/panels/FormExample.h"
2 ...
3 void subsystemsupervisor::Cell::init() {
4     ...
5     tsframework::CellPanelFactory* panelF = getContext()->getPanelFactory();
6     ...
7     panelF->add<subsystempanels::FormExample>("Form example");
```

Now you can compile your cell and you should see the “Form example” panel in the menu under the ‘control-panels’ section.

Also your element has created some documentation. Surf to <hostname>:<port>/<package-name>/html/index.html and you will see the package documentation for your cell. *form-example* will be in there, and clicking it brings up the documentation for your *form-example* element.

1.2.7 Demo 4: Refresh

The *refresh-example* element behaves much like the *ts-ajax* element we saw in the previous demo. The big difference is that this element implements periodic updating.

Make the refresh-example element

In your cell, run:

```
1 cd src/html/elements
2 ./new-element.js
3 name of the new element: refresh-example
4 creating new element <refresh-example>...
5 removing any .svn folders in refresh-example
6 Finished
```

Register the refresh-example element

Edit src/html/elements/elements.html and add the following line

```
1 <link rel="import" href="refresh-example/refresh-example.html">
```

This will tell AjaXell to load our new element.

Edit the refresh-example element

Edit src/html/elements/refresh-example/refresh-example.html

/src/html/elements/refresh-example/refresh-example.html

```

1 <link rel="import" href="/extern/bower_components/polymer/polymer.html">
2 <link rel="import" href="/extern/bower_components/iron-media-query/iron-media-query.
  ↪html">
3 <link rel="import" href="/extern/bower_components/paper-checkbox/paper-checkbox.html">
4 <link rel="import" href="/extern/bower_components/paper-input/paper-input.html">
5 <link rel="import" href="/ts/common-elements/reset-css/reset-css.html">
6 <link rel="import" href="/ts/common-elements/auto-update/auto-update.html">
7 <link rel="import" href="/ts/common-elements/iron-flex-layout-attributes/iron-flex-
  ↪layout-attributes.html">

8
9
10 <dom-module id="refresh-example">
11   <template>
12     <link rel="import" type="css" href="css/refresh-example-min.css?__inline=true">
13     <style include="iron-flex-layout-attributes"></style>
14     <style include="reset-css"></style>
15
16     <paper-checkbox checked="{{refreshtoggle}}>Pauze refresh</paper-checkbox>
17     <br>
18     <auto-update data="{{test}}"
19       interval="1000"
20       callback="RefreshExample::timesExecuted"
21       handle-as="text"
22       timeout="10000"
23       no-refresh$="{{refreshtoggle}}></auto-update>
24     <span>{{test}}</span>
25
26     <hr>
27
28     <paper-input value="{{favDino}}" label="your favorite dinosaur"></paper-input>
29     <auto-update data="{{test2}}"
30       interval="1000"
31       callback="RefreshExample::timesExecutedWithParameters"
32       handle-as="text"
33       parameters='["favoritedino"]'
34       favoritedino="{{favDino}}></auto-update>
35     <!-- notice the single & double quotes in the parameters variable
36     do yourself a favor and do not use capital letters in parameter names -->
37     <span>{{test2}}</span>
38   </template>
39   <script src="javascript/refresh-example-min.js?__inline=true"></script>
40 </dom-module>

```

Now edit /src/html/elements/refresh-example/css/refresh-example.scss

/src/html/elements/refresh-example/css/refresh-example.scss

```

1 :host {
2   color: black;
3 }

```

Now edit /src/html/elements/refresh-example/javascript/refresh-example.js

/src/html/elements/refresh-example/javascript/refresh-example.js

```

1 Polymer({
2   is: "refresh-example",
3   properties: {
4     test: String,

```

```

5     favDino: {
6         type: String,
7         value: "Velociraptor"
8     }
9 },
10 doTimesExecuted: function() {
11     var parameters = {};
12     parameters._eventType_ = "OnClick";
13     parameters._id_ = "RefreshExample-timesExecuted";
14     this.$.ajax_timesExecuted.params = parameters;
15     this.$.ajax_timesExecuted.url = window.location.origin + window.location.
→ pathname;
16     this.$.ajax_timesExecuted.generateRequest();
17 },
18 timesExecutedResponse: function(event, detail, sender) {
19     console.log(detail.response);
20     Polymer.dom(this.$.timesExecutedContainer).innerHTML = detail.response;
21 }
22 });

```

Now execute Grunt to build our new Polymer element.

```

1 cd src/html
2 grunt

```

Make the refresh-example panel

Make a new c++ file /src/common/panels/RefreshExample.cc

/src/common/panels/RefreshExample.cc

```

1 ****
2 * Trigger Supervisor Component
3 *
4 * Project Manager: Ildfons Magrans de Abril
5 *
6 * developer: Marc Magrans de Abril
7 ****
8 #include "subsystem/supervisor/panels/RefreshExample.h"
9
10 #include "ajax/PolymerElement.h"
11 #include "ajax/toolbox.h"
12
13 #include <iostream>
14 using namespace subsystempanels;
15 RefreshExample::RefreshExample(tsframework::CellAbstractContext* context, log4cplus::
→ Logger& logger)
16     : tsframework::CellPanel(context, logger)
17 {
18     logger_ = log4cplus::Logger::getInstance(logger.getName() + ".RefreshExample");
19 }
20
21 RefreshExample::~RefreshExample()
22 {
23     ;
24 }

```

```

26
27 void RefreshExample::timesExecuted(cgicc::Cgicc& cgi, std::ostream& out)
28 {
29     static int times(0);
30     times++;
31     std::ostringstream msg;
32     msg << "The RefreshExample::timesExecuted method has been executed " << times
33     << " times";
34     out << msg.str();
35 }
36 void RefreshExample::timesExecutedWithParameters(cgicc::Cgicc& cgi, std::ostream& out)
37 {
38     std::ostringstream msg;
39
40     std::map<std::string, std::string> values(ajax::toolbox::
41     <>getSubmittedValues(cgi));
42
43     msg << "Your favorite dinosaur is the ";
44
45     for(std::map<std::string, std::string>::iterator i(values.begin()); i !=_
46     <>values.end(); ++i) {
47         if ( i->first.compare("favoritedino") == 0 ) {
48             msg << i->second;
49         }
50     }
51
52     out << msg.str();
53 }
54
55 void RefreshExample::layout(cgicc::Cgicc& cgi)
56 {
57     //clear the output buffer of this panel, we start fresh
58     remove();
59
60     setEvent("RefreshExample::timesExecuted", ajax::Eventable::OnTime, this, &
61     <>RefreshExample::timesExecuted);
62     setEvent("RefreshExample::timesExecutedWithParameters", ajax::Eventable::
63     <>OnTime, this, &RefreshExample::timesExecutedWithParameters);
64 }
```

Make the /include/subsystem/supervisor/panels/RefreshExample.h file.

/include/subsystem/supervisor/panels/RefreshExample.h

```

1  ****
2  * Trigger Supervisor Component
3  *
4  * Project Manager: Ildfons Magrans de Abril
5  * Authors: Marc Magrans de Abril
6  *
7  ****/
```

```

8 #ifndef _subsystem_supervisor_panels_RefreshExample_h_
9 #define _subsystem_supervisor_panels_RefreshExample_h_
10
11 #include "ts/framework/CellPanel.h"
12
13 #include "ts/framework/CellAbstractContext.h"
14
15 #include "log4cplus/logger.h"
16
17 #include "cgicc/Cgicc.h"
18
19 #include <iostream>
20
21 namespace subsystempanels
22 {
23
24 class RefreshExample: public tsframework::CellPanel
25 {
26 public:
27     RefreshExample(tsframework::CellAbstractContext* context, log4cplus::Logger& logger);
28     ~RefreshExample();
29
30     void layout(cgicc::Cgicc& cgi);
31
32 private:
33     void timesExecuted(cgicc::Cgicc& cgi, std::ostream& out);
34     void timesExecutedWithParameters(cgicc::Cgicc& cgi, std::ostream& out);
35 };
36 } //ns subsystempanels
37 #endif

```

Register the new class in the Makefile.

```

1 Sources=\
2     version.cc\
3     Cell.cc\
4     CellContext.cc\
5     Configuration.cc\
6     ...
7     panels/RefreshExample.cc \
8     ...

```

Now register your new panel in the menu so users can access it.

Edit src/common/Cell.cc

```

1 #include "subsystem/supervisor/panels/RefreshExample.h"
2 ...
3 void subsystemsupervisor::Cell::init() {
4     ...
5     tsframework::CellPanelFactory* panelF = getContext()->getPanelFactory();
6     ...
7     panelF->add<subsystempanels::RefreshExample>("Refresh example");

```

Now you can compile your cell and you should see the “Form example” panel in the menu under the ‘control-panels’ section.

Also your element has created some documentation. Surf to <hostname>:<port>/<package-name>/html/index.html

and you will see the package documentation for your cell. *refresh-example* will be in there, and clicking it brings up the documentation for your *refresh-example* element.

1.2.8 Demo 5: Tables

Make the table-example element

In your cell, run:

```
1 cd src/html/elements
2 ./new-element.js
3 name of the new element: table-example
4 creating new element <table-example>...
5 removing any .svn folders in table-example
6 Finished
```

Register the table-example element

Edit src/html/elements/elements.html and add the following line

```
1 <link rel="import" href="table-example/table-example.html">
```

This will tell AjaXell to load our new element.

Edit the table-example element

Edit src/html/elements/table-example/table-example.html

/src/html/elements/table-example/table-example.html

```
1 <link rel="import" href="/extern/bower_components/polymer/polymer.html">
2 <link rel="import" href="/ts/common-elements/reset-css/reset-css.html">
3 <link rel="import" href="/ts/common-elements/iron-flex-layout-attributes/iron-flex-
4   layout-attributes.html">
5 <link rel="import" href="/extern/bower_components/vaadin-grid/vaadin-grid.html">
6 <link rel="import" href="/extern/bower_components/paper-material/paper-material.html">
7
8 <!--
9 This panel uses [vaadin-grid] (/extern/bower_components/vaadin-grid/index.html)
10 `table-example` demonstrates the uses of the vaadin-grid element.
11
12 -->
13 <dom-module id="table-example">
14   <template>
15     <style include="reset-css"></style>
16     <style include="iron-flex-layout-attributes"></style>
17
18     <link rel="stylesheet" type="text/css" href="css/table-example-min.css?__
19       inline=true">
20
21   </template>
22   <script src="javascript/table-example-min.js?__inline=true"></script>
</dom-module>
```

We'll add more stuff as we go along.

Now edit /src/html/elements/table-example/css/table-example.scss

/src/html/elements/table-example/css/table-example.scss

```
1 :host {  
2     display: block;  
3 }  
4  
5 paper-material {  
6     margin: 1em;  
7 }
```

Now execute Grunt to build our new Polymer element.

```
1 cd src/html  
2 grunt
```

Make the table-example panel

Make a new c++ file /src/common/panels/TableExample.cc

/src/common/panels/TableExample.cc

```
1 #include "subsystem/supervisor/panels/TableExample.h"  
2  
3 #include "ajax/toolbox.h"  
4 #include "ajax/PolymerElement.h"  
5 #include "ajax/toolbox.h"  
6 #include "json/json.h"  
7  
8 #include "log4cplus/loggingmacros.h"  
9  
10 #include <iostream>  
11 #include <time.h>  
12 #include <string>  
13 using namespace subsystempanels;  
14  
15 TableExample::TableExample( tsframework::CellAbstractContext* context, log4cplus::  
16     ~Logger& logger ) : tsframework::CellPanel(context,logger) {  
17     logger_ = log4cplus::Logger::getInstance(logger.getName() + ".TableExample");  
18 }  
19 TableExample::~TableExample() {  
20     remove();  
21 }  
22 void TableExample::layout(cgicc::Cgicc& cgi)  
23 {  
24     remove();  
25     add(new ajax::PolymerElement("table-example"));  
26 }
```

Make the /include/subsystem/supervisor/panels/TableExample.h file.

/include/subsystem/supervisor/panels/TableExample.h

```

1 #ifndef _subsystem_supervisor_panels_TableExample_h_
2 #define _subsystem_supervisor_panels_TableExample_h_
3
4 #include "ts/framework/CellPanel.h"
5
6 #include "ts/framework/CellAbstractContext.h"
7
8 #include "log4cplus/logger.h"
9
10 #include "cgicc/Cgicc.h"
11
12 #include <iostream>
13
14 namespace subsystempanels
15 {
16
17 class TableExample: public tsframework::CellPanel
18 {
19     public:
20         TableExample(tsframework::CellAbstractContext* context, log4cplus::Logger& logger);
21         ~TableExample();
22
23     void layout(cgicc::Cgicc& cgi);
24
25     private:
26
27 };
28 } //subsystempanels
29 #endif

```

Register the new class in the Makefile.

```

1 Sources=\
2     version.cc\
3     Cell.cc\
4     CellContext.cc\
5     Configuration.cc\
6     ...
7     panels/TableExample.cc \
8     ...

```

Now register your new panel in the menu so users can access it.

Edit src/common/Cell.cc

```

1 #include "subsystem/supervisor/panels/TableExample.h"
2 ...
3 void subsystemsupervisor::Cell::init() {
4     ...
5     tsframework::CellPanelFactory* panelF = getContext()->getPanelFactory();
6     ...
7     panelF->add<subsystempanels::TableExample>("Table example");

```

Now you can compile your cell and you should see the “Form example” panel in the menu under the ‘control-panels’ section.

Table1: A basic table with sorting

The first example will fetch a JSON file containing 1000 rows. Also we will make it sortable, so we will write our own sorting function.

C++ code

Edit /src/common/panels/TableExample.cc

/src/common/panels/TableExample.cc

```
1 void TableExample::layout(cgicc::Cgicc& cgi)
2 {
3     remove();
4     setEvent("getTable1", ajax::Eventable::OnClick, this, &TableExample::getTable1);
5     add(new ajax::PolymerElement("table-example"));
6 }
7
8 void TableExample::getTable1(cgicc::Cgicc& cgi, std::ostream& out) {
9     Json::Value root;
10
11     Json::Value items(Json::arrayValue);
12     for (int i = 0; i < 1000; i++) {
13         Json::Value row;
14
15         std::ostringstream msg;
16         msg << "row" << i;
17         row["some string"] = msg.str();
18
19         msg << "@cern.ch";
20         row["email"] = msg.str();
21
22         row["group"] = i % 3;
23
24         items.append(row);
25     }
26     root["items"] = items;
27
28     Json::Value columns(Json::arrayValue);
29
30     Json::Value column1;
31     column1["name"] = "some string";
32     column1["sortable"] = true;
33     columns.append(column1);
34
35     Json::Value column2;
36     column2["name"] = "email";
37     column2["sortable"] = true;
38     columns.append(column2);
39
40     Json::Value column3;
41     column3["name"] = "group";
42     column3["sortable"] = true;
43     columns.append(column3);
44
45     root["columns"] = columns;
46 }
```

```

47     out << root;
48 }
```

The `getTable1` call will provide the data of our first table. It will provide both column definitions and table rows.

Edit `/include/subsystem/supervisor/panels/TableExample.h`.

`/include/subsystem/supervisor/panels/TableExample.h`

```

1 namespace subsystempanels
2 {
3
4 class TableExample: public tsframework::CellPanel
5 {
6
7 private:
8     void getTable1(cgicc::Cgicc& cgi, std::ostream& out);
9
10 };
11 //subsystempanels
12 #endif
```

HTML

Edit `src/html/elements/table-example/table-example.html`

`/src/html/elements/table-example/table-example.html`

```

1 <dom-module id="table-example">
2   <template>
3
4     <h1>basic table, also sortable</h1>
5     <p>
6       This dataset has 1000 records, however this table is smart and will only render ↴
7       the rows currently in view.
8       This will keep the page performant.
9     </p>
10    <p>
11      Click on one of the columns to sort it, shift + click a column to use secondary ↴
12      sort
13    </p>
14    <paper-material elevation="1">
15      <ts-ajax data="{{table1}}" callback="getTable1" handle-as="json" auto></ts-ajax>
16      <vaadin-grid id="table1"
17        selection-mode="multi"
18        items="{{table1.items}}"
19        columns="{{table1.columns}}"></vaadin-grid>
20    </paper-material>
21  </template>
</dom-module>
```

`ts-ajax` will execute the `getTable1` callback (notice the ‘auto’ attribute). the table will populate itself as soon as `ts-ajax` completed the callback.

Making it sortable

In the C++ code, our generated JSON contains the “sortable”: true, but we still have to supply a sorting function ourselves.

This gets a bit complicated as we made more than one column sortable, so we have to support secondary sorting (the user can shift + click on a column to do that), and we have data that we need to parse to make properly sortable (like the email column).

Edit /src/html/elements/table-example/javascript/table-example.js

/src/html/elements/table-example/javascript/table-example.js

```
1 Polymer({
2     attached: function() {
3         this.$.table1.addEventListener('sort-order-changed', this.table1sort.bind(this));
4     },
5
6     table1sort: function() {
7         // secondary sort
8         var secondarySort = function() {return 0};
9         if (this.$.table1.sortOrder[1]) {
10             var columnNameSecondary = this.$.table1.columns[this.$.table1.sortOrder[1].
11             ↪column].name;
12             var directionSecondary = this.$.table1.sortOrder[1].direction == 'asc' ? -1 : 1;
13             secondarySort = function(row1, row2) {
14                 return (row1[columnNameSecondary] < row2[columnNameSecondary]) ?
15                     ↪directionSecondary : -directionSecondary;
16             }
17
18             if (columnNameSecondary == "some string") {
19                 secondarySort = function(row1, row2) {
20                     var a = parseInt(row1[columnNameSecondary].substring(3));
21                     var b = parseInt(row2[columnNameSecondary].substring(3));
22                     return (a < b) ? directionSecondary : -directionSecondary;
23                 }
24             }
25             if (columnNameSecondary == "email") {
26                 secondarySort = function(row1, row2) {
27                     var a = parseInt( row1[columnNameSecondary].substring(3).substring(0,
28                         ↪row1[columnNameSecondary].indexOf("@") - 3) );
29                     var b = parseInt( row2[columnNameSecondary].substring(3).substring(0,
30                         ↪row2[columnNameSecondary].indexOf("@") - 3) );
31                     return (a < b) ? directionSecondary : -directionSecondary;
32                 }
33             }
34         }
35
36         // primary sort
37         var columnName = this.$.table1.columns[this.$.table1.sortOrder[0].column].name;
38         var direction = this.$.table1.sortOrder[0].direction == 'asc' ? -1 : 1;
39
40         var sort = function(row1, row2) {
41             var result = (row1[columnName] < row2[columnName]) ? direction : -direction;
42             if (row1[columnName] == row2[columnName]) {
43                 result = secondarySort(row1, row2);
44             }
45             return result;
46         }
47     }
48 }
```

```

43
44     if (columnName == "some string") {
45         sort = function(row1, row2) {
46             var a = parseInt(row1[columnName].substring(3));
47             var b = parseInt(row2[columnName].substring(3));
48             var result = (a < b) ? direction : -direction;
49             if (a == b) {
50                 result = secondarySort(row1, row2);
51             }
52             return result;
53         }
54     }
55     if (columnName == "email") {
56         sort = function(row1, row2) {
57             var a = parseInt( row1[columnName].substring(3).substring(0, row1[columnName].
58             ↪indexOf("@") - 3) );
59             var b = parseInt( row2[columnName].substring(3).substring(0, row2[columnName].
60             ↪indexOf("@") - 3) );
61             var result = (a < b) ? direction : -direction;
62             if (a == b) {
63                 result = secondarySort(row1, row2);
64             }
65             return result;
66         }
67         this.$.table1.items.sort(sort);
68     }
69 } );

```

The result will look like this:

basic table, also sortable

This dataset has 1000 records, however this table is smart and will only render the rows currently in view. This will keep the page performant.

Click on one of the columns to sort it, shift + click a column to use secondary sort

<input type="checkbox"/>	Some String ↓ ₂	Email	Group ↑ ₁
<input type="checkbox"/>	row15	row15@cern.ch	0
<input type="checkbox"/>	row12	row12@cern.ch	0
<input type="checkbox"/>	row9	row9@cern.ch	0
<input type="checkbox"/>	row6	row6@cern.ch	0
<input type="checkbox"/>	row3	row3@cern.ch	0
<input type="checkbox"/>	row0	row0@cern.ch	0
<input type="checkbox"/>	row997	row997@cern.ch	1
<input type="checkbox"/>	row994	row994@cern.ch	1
<input type="checkbox"/>	row991	row991@cern.ch	1
<input type="checkbox"/>	row988	row988@cern.ch	1

Table2: An AJAX table

This dataset will be 5000 rows big, and is loaded asynchronously. This means the table will make requests for more data as the user scrolls.

C++ code

Edit /src/common/panels/TableExample.cc

/src/common/panels/TableExample.cc

```
1 void TableExample::layout(cgicc::Cgicc& cgi)
2 {
3     remove();
4     setEvent("getTable1", ajax::Eventable::OnClick, this, &TableExample::getTable1);
5     setEvent("getTable2", ajax::Eventable::OnClick, this, &TableExample::getTable2);
6     add(new ajax::PolymerElement("table-example"));
7 }
8
9 void TableExample::getTable2(cgicc::Cgicc& cgi, std::ostream& out) {
10    std::map<std::string, std::string> values(ajax::toolbox::getSubmittedValues(cgi));
11    int index;
12    std::stringstream indexs(values.find("index")->second);
13    indexs >> index;
14    int count;
15    std::stringstream counts(values.find("count")->second);
16    counts >> count;
17    if (indexs.fail() || counts.fail()) {
18        return;
19    }
20
21    Json::Value rows(Json::arrayValue);
22    for (size_t i = index; i <= count + index; i++) {
23        Json::Value row;
24
25        row["random number"] = rand();
26        std::ostringstream msg;
27        msg << "item " << i;
28        row["some string"] = msg.str();
29
30        rows.append(row);
31    }
32
33    out << rows;
34 }
```

Edit /include/subsystem/supervisor/panels/TableExample.h.

/include/subsystem/supervisor/panels/TableExample.h

```
1 namespace subsystempanels
2 {
3
4     class TableExample: public tsframework::CellPanel
5     {
6
7         private:
8         void getTable2(cgicc::Cgicc& cgi, std::ostream& out);
```

```

9
10 } ;
11 } //subsystempanels
12 #endif

```

Unlike the first example, were we sent both items and column info, we only send the items in this example. The column info will now be declared in JavaScript.

HTML & Javascript

Edit src/html/elements/table-example/table-example.html

/src/html/elements/table-example/table-example.html

```

1 <dom-module id="table-example">
2   <template>
3
4     <h1>Asynchronous table</h1>
5     <p>
6       This dataset is not fetched in one go from the server, but in chunks.
7       Use this if the data is expensive to render server side.
8       Scroll fast to see the data being loaded.
9     </p>
10    <p>
11      Unfortunately, async data means we cannot use sorting...
12    </p>
13    <paper-button raised primary on-click="table2_scrollend">Scroll to end</paper-
14    ↵button>
15    <paper-button raised primary on-click="table2_scrollstart">Scroll to start</paper-
16    ↵button>
17    <paper-button raised primary on-click="table2_scroll3000">Scroll to line 3000</
18    ↵paper-button>
19    <paper-material elevation="1">
20      <ts-ajax id="ajax_table2"
21        data="{{tsajax_table2}}"
22        callback="getTable2"
23        handle-as="json"
24        parameters='["index", "count"]'></ts-ajax>
25      <vaadin-grid id="table2"
26        selection-mode="multi"
27        items="{{table2items}}"
28        columns="{{table2columns}}"
29        size="5000"></vaadin-grid>
30    </paper-material>
31
32  </template>
33 </dom-module>

```

Now edit /src/html/elements/table-example/javascript/table-example.js

/src/html/elements/table-example/javascript/table-example.js

```

1 Polymer({
2   properties: {
3     /**
4      * The callback for the asynchronous data request
5     */

```

```
6     table2callback: Function,
7     /**
8      * The ajax response from ts-ajax
9      */
10    tsajax_table2: {
11      type: Object,
12      observer: 'newTable2data'
13    },
14    /**
15     * The `items` dataset for our table. It is a function because it will fetch
16     * data for us rather than just be a dumb array containing all the data.
17     */
18    table2items: {
19      type: Function,
20      value: function() {
21        return function(params, callback) {
22          this.table2callback = callback;
23          var ajax = this.$.ajax_table2;
24          ajax.index = params.index;
25          ajax.count = params.count;
26          ajax.generateRequest();
27        }.bind(this);
28      }
29    },
30    /**
31     * The columns of our data
32     */
33    table2columns: {
34      type: Array,
35      value: function() {
36        return [{name: "some string"}, {name: "random number"}]
37      }
38    }
39  },
40
41  newTable2data: function(newdata) {
42    if (this.table2callback) {
43      // note that the callback can also take a second parameter that updates the size
44      // this can be used to implement infinite scrolling or datasets with changing
45      // sizes
46      this.table2callback(newdata);
47    }
48  },
49
50  table2_scrollend: function() {
51    this.$.table2.scrollToEnd();
52  },
53  table2_scrollstart: function() {
54    this.$.table2.scrollToStart();
55  },
56  table2_scroll3000: function() {
57    this.$.table2.scrollToRow(3000);
58  },
59});
```

Notice the asynchronous data fetching makes our JavaScript quite a bit more complex. Use this approach when generating data on server-side is slow or otherwise expensive.

The result will look like this:

Asynchronous table

This dataset is not fetched in one go from the server, but in chunks. Use this if the data is expensive to render server side. Scroll fast to see the data being loaded.

Unfortunately, async data means we cannot use sorting...

	Some String	Random Number
<input type="checkbox"/>	item 2991	454249139
<input type="checkbox"/>	item 2992	334040127
<input type="checkbox"/>	item 2993	3791705
<input type="checkbox"/>	item 2994	1123807238
<input type="checkbox"/>	item 2995	404166632
<input type="checkbox"/>	item 2996	852936502
<input type="checkbox"/>	item 2997	1934410235
<input type="checkbox"/>	item 2998	578210321
<input type="checkbox"/>	item 2999	175150901
<input type="checkbox"/>	item 3000	1065668260

Table3: Frozen columns & styling

This table will have a ‘frozen’ column that will always be visible. Also some columns are ‘hidable’, so we will have the option to hide them.

C++ code

Edit /src/common/panels/TableExample.cc

/src/common/panels/TableExample.cc

```

1 void TableExample::layout(cgicc::Cgicc& cgi)
2 {
3     remove();
4     setEvent("getTable1", ajax::Eventable::OnClick, this, &TableExample::getTable1);
5     setEvent("getTable2", ajax::Eventable::OnClick, this, &TableExample::getTable2);
6     setEvent("getTable3", ajax::Eventable::OnClick, this, &TableExample::getTable3);
7     add(new ajax::PolymerElement("table-example"));
8 }
9
10 void TableExample::getTable3(cgicc::Cgicc& cgi, std::ostream& out) {
11     std::map<std::string, std::string> values(ajax::toolbox::getSubmittedValues(cgi));
12     int index;
13     std::stringstream indexs(values.find("index")->second);
14     indexs >> index;
15     int count;
16     std::stringstream counts(values.find("count")->second);
17     counts >> count;
18     if (indexs.fail() || counts.fail()) {
19         return;

```

```
20 }
21
22     std::string states[] = {"ok", "warning", "error"};
23     Json::Value rows(Json::arrayValue);
24     for (size_t i = index; i <= count + index; i++) {
25         Json::Value row;
26
27         std::ostringstream msg;
28         msg << "cell #" << i;
29         row["cell"] = msg.str();
30
31         row["CPU (%)"] = rand() % 100;
32         row["memory (%)"] = rand() % 100;
33         row["network (%)"] = rand() % 100;
34         row["PID"] = rand() % 700;
35         row["RSS"] = rand();
36         row["VSZ"] = rand();
37         row["state"] = states[i % 3];
38
39         time_t now = time(0);
40         struct tm tstruct;
41         char buf[80];
42         tstruct = *localtime(&now);
43         // Visit http://en.cppreference.com/w/cpp/chrono/c strftime
44         // for more information about date/time format
45         strftime(buf, sizeof(buf), "%Y-%m-%d.%X", &tstruct);
46         row["timestamp"] = buf;
47
48         rows.append(row);
49     }
50
51     out << rows;
52 }
```

Edit /include/subsystem/supervisor/panels/TableExample.h.

/include/subsystem/supervisor/panels/TableExample.h

```
1 namespace subsystempanels
2 {
3
4     class TableExample: public tsframework::CellPanel
5     {
6
7         private:
8         void getTable3(cgicc::Cgicc& cgi, std::ostream& out);
9     };
10 } // subsystempanels
11 #endif
```

HTML & Javascript

Edit src/html/elements/table-example/table-example.html

/src/html/elements/table-example/table-example.html

```

1 <dom-module id="table-example">
2   <template>
3
4     <h1>Frozen columns && styling</h1>
5     <p>
6       The first column in this dataset will always be visible.
7       Errors and warnings will be very visible.
8     </p>
9     <p>
10      You can also choose to hide some columns using the icon in the upper right.
11    </p>
12    <paper-material elevation="1">
13      <ts-ajax id="ajax_table3"
14        data="{{tsajax_table3}}"
15        callback="getTable3"
16        handle-as="json"
17        parameters='["index", "count"]'></ts-ajax>
18      <vaadin-grid id="table3"
19        selection-mode="multi"
20        items="{{table3items}}"
21        columns='{{table3columns}}'
22        size="5000"
23        row-class-generator="[[table3rowclass]]"
24        frozen-columns="1"></vaadin-grid>
25    </paper-material>
26
27  </template>
28</dom-module>

```

Now edit /src/html/elements/table-example/javascript/table-example.js

/src/html/elements/table-example/javascript/table-example.js

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17 Polymer({
18   properties: {
19     /**
20      * The callback for the asynchronous data request
21     */
22     table3callback: Function,
23     /**
24      * The ajax response from ts-ajax
25     */

```

```
26     tsajax_table3: {
27         type: Object,
28         observer: 'newTable3data'
29     },
30     /**
31      * The `items` dataset for our table. It is a function because it will fetch
32      * data for us rather than just be a dumb array containing all the data.
33      */
34     table3items: {
35         type: Function,
36         value: function() {
37             return function(params, callback) {
38                 this.table3callback = callback;
39                 var ajax = this.$.ajax_table3;
40                 ajax.index = params.index;
41                 ajax.count = params.count;
42                 ajax.generateRequest();
43             }.bind(this);
44         }
45     },
46     /**
47      * The columns of our data
48      */
49     table3columns: {
50         type: Array,
51         value: function() {
52             return [
53                 {name: "cell"},
54                 {name: "CPU (%)", hidable: true},
55                 {name: "memory (%)", hidable: true},
56                 {name: "network (%)", hidable: true},
57                 {name: "PID", hidable: true},
58                 {name: "RSS", hidable: true},
59                 {name: "VSZ", hidable: true},
60                 {name: "state"},
61                 {name: "timestamp", hidable: true}
62             ];
63         }
64     },
65     table3rowclass: {
66         type: Function,
67         value: function() {
68             return function(row) {
69                 var bgcolor = "";
70                 var color = "";
71                 // var fontweight = "";
72                 var DOMrow = row.element;
73
74                 if (row.data.state == "error") {
75                     bgcolor = "#EC4343";
76                     color = "white";
77                     DOMrow.childNodes[8].style.textTransform = "uppercase";
78                 } else if (row.data.state == "warning") {
79                     bgcolor = "#EC9943";
80                     color = "white";
81                     DOMrow.childNodes[8].style.textTransform = "uppercase";
82                 }
83             }
84         }
85     }
86 }
```

```

84
85     DOMrow.style.color = color;
86     for (var i = 0; i < DOMrow.childNodes.length; i++) {
87         DOMrow.childNodes[i].style.backgroundColor = bgcolor;
88     }
89
90     return "";
91 }
92 }
93 }
94 ,
95
96 newTable3data: function(newdata) {
97     if (this.table3callback && newdata) {
98         this.table3callback(newdata);
99     }
100 }
101
102 }) ;

```

The result will look like this:
Frozen columns & styling

The first 2 columns in this dataset will always be visible. Errors and warnings will be very visible.

You can also choose to hide some columns using the icon in the upper right.

<input type="checkbox"/> Cell	CP U (%)	Memory (%)	Network (%)	PID	RSS	VSZ	State	Timestamp	
<input type="checkbox"/> cell #0	56	98	11	69	1143739824	601430824	OK	2016-04-07.14:52:10	
<input type="checkbox"/> cell #1	93	74	88	526	175664460	100668754	WARNING	2016-04-07.14:52:10	
<input type="checkbox"/> cell #2	58	84	77	544	255169849	951214646	ERROR	2016-04-07.14:52:10	
<input type="checkbox"/> cell #3	33	25	44	194	698797658	874485379	OK	2016-04-07.14:52:10	
<input type="checkbox"/> cell #4	60	40	35	458	798758151	24551057	WARNING	2016-04-07.14:52:10	
<input type="checkbox"/> cell #5	83	62	99	360	41542630	857441135	ERROR	2016-04-07.14:52:11	
<input type="checkbox"/> cell #6	16	71	48	492	1644694092	1513223639	OK	2016-04-07.14:52:11	
<input type="checkbox"/> cell #7	91	69	0	633	183946714	1117362635	WARNING	2016-04-07.14:52:11	
<input type="checkbox"/> cell #8	82	60	72	629	2025758855	696556657	ERROR	2016-04-07.14:52:11	
<input type="checkbox"/> cell #9	12	28	36	473	1518401987	1368301838	OK	2016-04-07.14:52:11	

Table4: Custom cell content

Sometimes you want to display something fancy instead of just data. This example will replace the ‘progress’ value (0-100) with an html progress bar.

C++ code

Edit /src/common/panels/TableExample.cc

/src/common/panels/TableExample.cc

```
1 void TableExample::layout(cgicc::Cgicc& cgi)
2 {
3     remove();
4     setEvent("getTable1", ajax::Eventable::OnClick, this, &TableExample::getTable1);
5     setEvent("getTable2", ajax::Eventable::OnClick, this, &TableExample::getTable2);
6     setEvent("getTable3", ajax::Eventable::OnClick, this, &TableExample::getTable3);
7     setEvent("getTable4", ajax::Eventable::OnClick, this, &TableExample::getTable4);
8     add(new ajax::PolymerElement("table-example"));
9 }
10
11 void TableExample::getTable4(cgicc::Cgicc& cgi, std::ostream& out) {
12     Json::Value root;
13
14     Json::Value items(Json::arrayValue);
15     for (int i = 1; i <= 100; i++) {
16         Json::Value row;
17
18         std::ostringstream msg;
19         msg << "row " << i;
20         row["rowNumber"] = msg.str();
21
22         row["progress"] = rand() % 100;
23
24         row["extra info"] = "some extra info about row #" + msg.str();
25
26         items.append(row);
27     }
28     root["items"] = items;
29
30     Json::Value columns(Json::arrayValue);
31
32     Json::Value column1;
33     column1["name"] = "rowNumber";
34     columns.append(column1);
35
36     Json::Value column3;
37     column3["name"] = "progress";
38     columns.append(column3);
39
40     root["columns"] = columns;
41
42     out << root;
43 }
```

Edit /include/subsystem/supervisor/panels/TableExample.h.

/include/subsystem/supervisor/panels/TableExample.h

```
1 namespace subsystempanels
2 {
3
4     class TableExample: public tsframework::CellPanel
5     {
6
7         private:
8         void getTable4(cgicc::Cgicc& cgi, std::ostream& out);
9
10    };
```

```

11 } //subsystempanels
12 #endif

```

HTML & Javascript

Edit src/html/elements/table-example/table-example.html

/src/html/elements/table-example/table-example.html

```

1 <dom-module id="table-example">
2   <template>
3
4     <h1>Custom HTML instead of pure data</h1>
5     <p>
6       Instead of some progress number (e.g. 10 or 80), you can show a progress bar.
7     </p>
8     <paper-material elevation="1">
9       <ts-ajax data="{{table4}}" callback="getTable4" handle-as="json" auto></ts-ajax>
10      <vaadin-grid id="table4"
11        items="{{table4.items}}"
12        columns="{{table4.columns}}"></vaadin-grid>
13    </paper-material>
14
15  </template>
16</dom-module>

```

Now edit /src/html/elements/table-example/javascript/table-example.js

/src/html/elements/table-example/javascript/table-example.js

```

1 Polymer({
2   properties: {
3     table4: {
4       type: Object,
5       observer: "table4Changed"
6     }
7   },
8
9   table4Changed: function(table4) {
10     if (table4) {
11       table4.columns[1].renderer = this.table4ProgressRenderer;
12     }
13   },
14
15   table4ProgressRenderer: function(cell) {
16     cell.element.innerHTML = '';
17     var child = document.createElement('progress');
18     child.setAttribute('value', cell.data);
19     child.setAttribute('max', 100);
20     cell.element.appendChild(child);
21   }
22 }
23 );

```

The result will look like this:

Custom HTML instead of pure data

Instead of some progress number (e.g. 10 or 80), you can show a progress bar.

Row Number	Progress
row 1	<div style="width: 50%;"><div style="width: 100%;"> </div></div>
row 2	<div style="width: 80%;"><div style="width: 100%;"> </div></div>
row 3	<div style="width: 20%;"><div style="width: 100%;"> </div></div>
row 4	<div style="width: 5%;"><div style="width: 100%;"> </div></div>
row 5	<div style="width: 40%;"><div style="width: 100%;"> </div></div>
row 6	<div style="width: 60%;"><div style="width: 100%;"> </div></div>
row 7	<div style="width: 30%;"><div style="width: 100%;"> </div></div>
row 8	<div style="width: 70%;"><div style="width: 100%;"> </div></div>
row 9	<div style="width: 90%;"><div style="width: 100%;"> </div></div>
row 10	<div style="width: 100%;"><div style="width: 100%;"> </div></div>

Table5: Details on selection

You can show some extra details when one of the rows is clicked.

HTML & Javascript

Edit src/html/elements/table-example/table-example.html

/src/html/elements/table-example/table-example.html

```
1 <dom-module id="table-example">
2   <template>
3
4     <h1>details on selection</h1>
5     <p>
6       Select an item, and a detail view will appear
7     </p>
8     <paper-material elevation="1">
9       <ts-ajax data="{{table5}}" callback="getTable4" handle-as="json" auto></ts-ajax>
10      <vaadin-grid id="table5"
11        items="{{table5.items}}"
12        columns="{{table5.columns}}"></vaadin-grid>
13    </paper-material>
14
15  </template>
16</dom-module>
```

Now edit /src/html/elements/table-example/javascript/table-example.js

/src/html/elements/table-example/javascript/table-example.js

```
1 Polymer({
2   properties: {
3     table5_detailSelected: {
4       type: Number,
```

```

5         value: 0
6     }
7 },
8
9     attached: function() {
10
11     this.$.table5.addEventListener('selected-items-changed', function() {
12         this.$.table5.setRowDetailsVisible(this.table5_detailSelected, false);
13         var selected = this.$.table5.selection.selected();
14         if (selected.length == 1) {
15             this.$.table5.setRowDetailsVisible(selected[0], true);
16             this.table5_detailSelected = selected[0];
17         }
18     }.bind(this));
19
20     this.$.table5.rowDetailsGenerator = function(rowIndex) {
21         var elem = document.createElement('table-detail');
22
23         this.$.table5.getItem(rowIndex, function(error, item) {
24             if (!error) {
25                 elem.item = item;
26             }
27         });
28
29         return elem;
30     }.bind(this);
31 }
32
33 }) ;

```

Now our code will show a ‘table-detail’ element when a row is selected. We still need to make this element.

Create src/html/elements/table-example/table-detail.html

/src/html/elements/table-example/table-detail.html

```

1 <link rel="import" href="/extern/bower_components/paper-material/paper-material.html">
2
3 <dom-module id="table-detail">
4     <template>
5         <link rel="stylesheet" type="text/css" href="css/table-detail-min.css?__
6             ↪inline=true">
7
8         <paper-material elevation="1">
9             <p>
10                [ [getExtraInfo(item)] ]
11            </p>
12        </paper-material>
13
14    </template>
15    <script src="javascript/table-detail-min.js?__inline=true"></script>
</dom-module>

```

Now create /src/html/elements/table-example/javascript/table-detail.js

/src/html/elements/table-example/javascript/table-detail.js

```

1 Polymer({
2     is: "table-detail",

```

```
3 properties: {
4     item: {
5         type: Object
6     }
7 },
8
9 getExtraInfo: function(item) {
10     // data binding can't handle spaces
11     return item['extra info'];
12 }
13 );
```

Now create /src/html/elements/table-example/css/table-detail.scss

/src/html/elements/table-example/css/table-detail.scss

```
1 :host {
2     display: block;
3     height: 8em;
4 }
5
6 paper-material {
7     padding: 1em;
8     margin: 1em;
9 }
```

The result will look like this:

details on selection

Select an item, and a detail view will appear

Row Number	Progress
row 1	97
row 2	24
row 3	48
row 4	65
some extra info about row #row 4	
row 5	71
row 6	21
row 7	13
row 8	40

1.2.9 Demo 6: Charts

For a complete list of options available for each type of chart, look at the official NVD3 docs (<https://nvd3-community.github.io/nvd3/examples/documentation.html>)

Make the chart-examples element

In your cell, run:

```
1 cd src/html/elements
2 ./new-element.js
3 name of the new element: chart-examples
4 creating new element <chart-examples>...
5 removing any .svn folders in chart-examples
6 Finished
```

Register the chart-examples element

Edit src/html/elements/elements.html and add the following line

```
1 <link rel="import" href="chart-examples/chart-examples.html">
```

This will tell AjaXell to load our new element.

Edit the chart-examples element

Edit src/html/elements/chart-examples/chart-examples.html

/src/html/elements/chart-examples/chart-examples.html

```
1 <link rel="import" href="/extern/bower_components/polymer/polymer.html">
2 <link rel="import" href="/ts/common-elements/reset-css/reset-css.html">
3 <link rel="import" href="/ts/common-elements/iron-flex-layout-attributes/iron-flex-
4   ↪layout-attributes.html">
5 <link rel="import" href="/ts/common-elements/math-equation/math-equation.html">
6 <link rel="import" href="/ts/common-elements/charts/line-chart/line-chart.html">
7 <link rel="import" href="/ts/common-elements/charts/cumulative-line-chart/cumulative-
8   ↪line-chart.html">
9 <link rel="import" href="/ts/common-elements/charts/focus-line-chart/focus-line-chart.
10  ↪html">
11 <link rel="import" href="/ts/common-elements/charts/horizontal-stacked-bar-chart/
12   ↪horizontal-stacked-bar-chart.html">
13 <link rel="import" href="/ts/common-elements/charts/historical-bar-chart/historical-
14   ↪bar-chart.html">
15 <link rel="import" href="/ts/common-elements/charts/discrete-bar-chart/discrete-bar-
16   ↪chart.html">
17 <link rel="import" href="/ts/common-elements/charts/stacked-bar-chart/stacked-bar-
18   ↪chart.html">
19 <link rel="import" href="/ts/common-elements/charts/pie-chart/pie-chart.html">
20 <link rel="import" href="/ts/common-elements/charts/scatter-chart/scatter-chart.html">
21 <link rel="import" href="/ts/common-elements/charts/stacked-area-chart/stacked-area-
22   ↪chart.html">
23 <link rel="import" href="/ts/common-elements/charts/parallel-chart/parallel-chart.html
24   ↪">
25 <link rel="import" href="/ts/common-elements/charts/candlestick-chart/candlestick-
26   ↪chart.html">
27
28 <!--
29 `chart-examples` gives some examples of how to use chart elements in the
30 common-elements package.
31 -->
```

```
22 <dom-module id="chart-exampless">
23   <template>
24     <style include="reset-css"></style>
25     <style include="iron-flex-layout-attributes"></style>
26     <link rel="stylesheet" type="text/css" href="css/chart-exampless-min.css?__
27     ↪inline=true">
28   </template>
29   <script src="javascript/chart-exampless-min.js?__inline=true"></script>
30 </dom-module>
```

We'll add more stuff as we go along.

Now edit /src/html/elements/chart-examples/css/chart-examples.scss

/src/html/elements/chart-examples/css/chart-examples.scss

```
1 :host {
2   display: block;
3   height: 100%;
4 }
5
6 paper-material {
7   height: 400px;
8   margin: 1em;
9 }
```

Now execute Grunt to build our new Polymer element.

```
1 cd src/html
2 grunt
```

Make the chart-examples panel

Make a new c++ file /src/common/panels/ChartExamples.cc

/src/common/panels/ChartExamples.cc

```
1 #include "subsystem/supervisor/panels/ChartExamples.h"
2 #include "ajax/PolymerElement.h"
3 #include "json/json.h"
4 #include <math.h>
5
6 using namespace subsystempanels;
7 ChartExamples::ChartExamples( tsframework::CellAbstractContext* context, log4cplus::
8   ↪Logger& logger)
9   :tsframework::CellPanel(context, logger) {
10     logger_ = log4cplus::Logger::getInstance(logger.getName() + ".ChartExamples");
11   }
12
13 void ChartExamples::layout(cgicc::Cgicc& cgi) {
14   remove();
15   add(new ajax::PolymerElement("chart-examples"));
16 }
```

Make the /include/subsystem/supervisor/panels/ChartExamples.h file.

/include/subsystem/supervisor/panels/ChartExamples.h

```

1 #ifndef _subsystem_supervisor_panels_ChartExamples_h_
2 #define _subsystem_supervisor_panels_ChartExamples_h_
3
4 #include "ts/framework/CellPanel.h"
5 #include "log4cplus/logger.h"
6 #include "cgicc/Cgicc.h"
7
8 namespace subsystempanels {
9     class ChartExamples: public tsframework::CellPanel {
10     public:
11         ChartExamples(tsframework::CellAbstractContext* context, log4cplus::Logger& logger);
12         void layout(cgicc::Cgicc& cgi);
13     private:
14 };
15 }
16 #endif

```

Register the new class in the Makefile.

```

1 Sources=\
2     version.cc \
3     Cell.cc \
4     CellContext.cc \
5     Configuration.cc \
6     ...
7     panels/ChartExamples.cc \
8     ...

```

Now register your new panel in the menu so users can access it.

Edit src/common/Cell.cc

```

1 #include "subsystem/supervisor/panels/ChartExamples.h"
2 ...
3 void subsystemsupervisor::Cell::init() {
4     ...
5     tsframework::CellPanelFactory* panelF = getContext()->getPanelFactory();
6     ...
7     panelF->add<subsystempanels::ChartExamples>("Chart examples");

```

Now you can compile your cell and you should see the “Form example” panel in the menu under the ‘control-panels’ section.

Chart1: A line chart

This example will make a basic line chart, and will show a simple line. Data is rendered client-side for simplicity.

Also note the use of the *math-equation* element to display LaTeX math.

HTML

Edit src/html/elements/chart-examples/chart-examples.html
/src/html/elements/chart-examples/chart-examples.html

```

1 <dom-module id="chart-examples">
2   <template>
3
4     <div horizontal layout>
5       <paper-material elevation="1" flex vertical layout>
6         <math-equation big>f(x) = x^5 + 3.5x^4 - 2.4x^3 -12.5x^2 + 1.5x + 9</math-
7           ↵equation>
8           <line-chart flex data="{{chart1data}}" config="{{chart1config}}" configure-
9             ↵chart="{{chart1JSConfig}}"></line-chart>
10          </paper-material>
11        </div>
12
13      </template>
14    </dom-module>

```

JavaScript

For this first example, we generate our data client side.

Edit /src/html/elements/chart-examples/javascript/chart-examples.js

```

1 Polymer({
2   is: 'chart-examples',
3   properties: {
4     chart1data: {
5       type: Array,
6       value: function() {
7         var data = []
8         for (var i = -3.5; i < 2.2; i = i+0.01) {
9           data.push({
10             x: i,
11             y: Math.pow(i,5) + 3.5*Math.pow(i,4) - 2.5*Math.pow(i,3) - 12.5*Math.
12               ↵pow(i,2) + 1.5*i + 9
13             });
14           }
15
16         return [
17           {
18             values: data,
19             key: 'fifth degree polynomial',
20             color: '#00671A'
21           }
22         ];
23       }
24     },
25     chart1config: {
26       type: Object,
27       value: function() {
28         return {
29           useInteractiveGuideline: true,
30           margin: {
31             left: 70
32           }
33         }
34       }
35     }
36   }
37 }

```

```

34     }
35   },
36
37   chart1JSConfig: {
38     type: Function,
39     value: function() {
40       return function() {
41         var superscript = "$^0$$^1$$^2$$^3$$^4$$^5$$^6$$^7$$^8$$^9$";
42         var formatPower = function(d) {
43           return (d + "").split("").map(function(c) {
44             return superscript[c];
45           }).join("");
46         };
47         this._chart.xAxis
48           .axisLabel("x")
49           .tickFormat(d3.format(',.2f'));
50
51         this._chart.yAxis
52           .axisLabel("x" + formatPower(5) + " + 3.5x" + formatPower(4) + " - 2.5x" +
53           →+ formatPower(3) + " - 12.5x" + formatPower(2) + " + 1.5x + 9")
54           .tickFormat(d3.format(',.2f'));
55       }
56     }
57   }
58 } ;

```

The result will look like this:

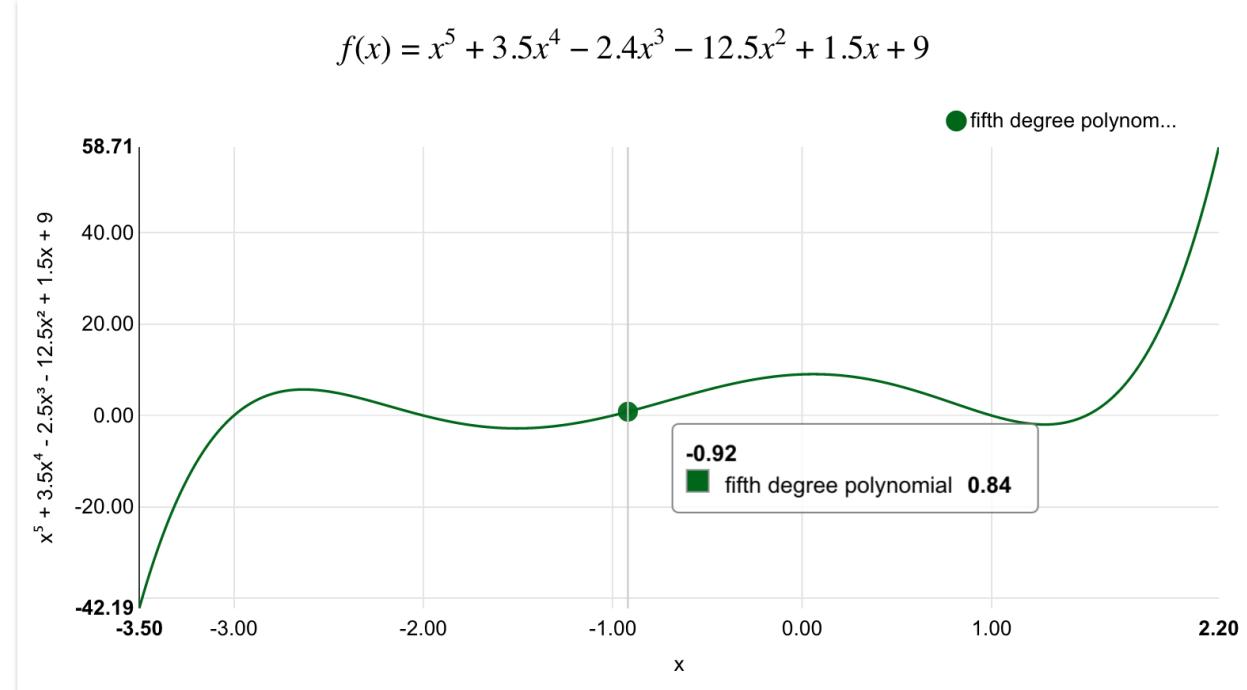


Chart2: A more advanced line chart

This line chart will feature multiple plots, missing data gaps, and a date for the x axis.

Data is still generated client side.

HTML & Javascript

Edit src/html/elements/chart-examples/chart-examples.html

/src/html/elements/chart-examples/chart-examples.html

```

1  <dom-module id="chart-examples">
2    <template>
3
4      <div horizontal layout>
5        <paper-material elevation="1" flex vertical layout>
6          <math-equation big>f(x) = x^5 + 3.5x^4 - 2.4x^3 -12.5x^2 + 1.5x + 9</math-
7          ↪equation>
8            <line-chart flex data="{{chart1data}}" config="{{chart1config}}" configures-
9            ↪chart="{{chart1JSConfig}}"></line-chart>
10           </paper-material>
11           <paper-material elevation="1" flex horizontal layout>
12             <line-chart flex data="{{chart2data}}" config="{{chart1config}}" configures-
13             ↪chart="{{chart2JSConfig}}"></line-chart>
14           </paper-material>
15     </div>
16
17   </template>
18 </dom-module>
```

Note that we put the second chart next to the first chart, this to demonstrate the ability to use the layout attributes.

Now edit /src/html/elements/chart-examples/javascript/chart-examples.js

/src/html/elements/chart-examples/javascript/chart-examples.js

```

1 Polymer({
2   is: 'chart-examples',
3   properties: {
4     chart2data: {
5       type: Array,
6       value: function() {
7         var sin = [],
8           sin2 = [],
9           cos = [],
10          rand = [],
11          rand2 = []
12          ;
13         for (var i = 0; i < 100; i++) {
14           sin.push({x: i, y: i % 10 == 5 ? null : Math.sin(i/10)}); //the nulls ↪
15           ↪are to show how defined works
16           sin2.push({x: i, y: Math.sin(i/5) * 0.4 - 0.25});
17           cos.push({x: i, y: .5 * Math.cos(i/10)});
18           rand.push({x:i, y: Math.random() / 10});
19           rand2.push({x: i, y: Math.cos(i/10) + Math.random() / 10 })
20         }
21         return [
22           {
23             area: true,
24             values: sin,
25             key: "Sine Wave",
26           }
27         ]
28       }
29     }
30   }
31 }
```

```

25         color: "#ff7f0e",
26         strokeWidth: 4,
27         classed: 'dashed'
28     },
29     {
30         values: cos,
31         key: "Cosine Wave",
32         color: "#2ca02c"
33     },
34     {
35         values: rand,
36         key: "Random Points",
37         color: "#2222ff"
38     },
39     {
40         values: rand2,
41         key: "Random Cosine",
42         color: "#667711",
43         strokeWidth: 3.5
44     },
45     {
46         area: true,
47         values: sin2,
48         key: "Fill opacity",
49         color: "#EF9CFB",
50         fillOpacity: .1
51     }
52 ];
53 },
54
55 },
56
57 chart2JSConfig: {
58     type: Function,
59     value: function() {
60         return function() {
61             this._chart.xAxis
62                 .showMaxMin(false)
63                 .tickFormat(function(d) { return d3.time.format('%x')(new Date(d)) });
64
65             this._chart.yAxis
66                 .tickFormat(d3.format(',.2f'));
67         }
68     }
69 }
70 }
71 );

```

The result will look like this:

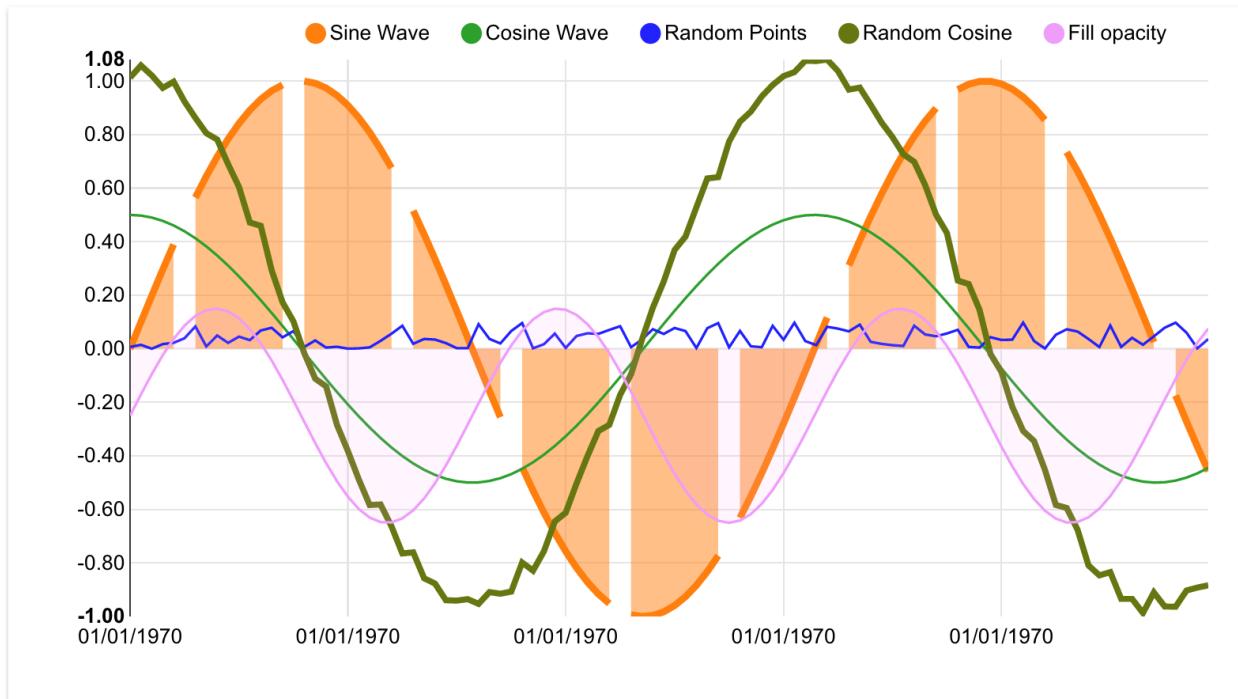


Chart3: A cumulative line chart

This type of chart shows a relative change of data. You will notice the data in this chart starts from 0. The user can also click on any point to instruct the chart to take that point in the x axis as the new relative zero.

HTML & Javascript

Edit `src/html/elements/chart-examples/chart-examples.html`

`/src/html/elements/chart-examples/chart-examples.html`

```

1  <dom-module id="chart-examples">
2    <template>
3
4      <div horizontal layout>
5        <paper-material elevation="1" flex horizontal layout>
6          <cumulative-line-chart data="{{chart3data}}" config="{{chart3config}}"\u202a
7            &configure-chart="{{chart3JSConfig}}></cumulative-line-chart>
8        </div>
9      </template>
10     </dom-module>
11

```

Now edit `/src/html/elements/chart-examples/javascript/chart-examples.js`

`/src/html/elements/chart-examples/javascript/chart-examples.js`

```

1 Polymer({
2   properties: {
3     chart3data: {

```

```

4   type: Array,
5   value: function() {
6     return [
7       {
8         key: "Long",
9         values: [ [ 1083297600000 , -2.974623048543] , [ 1085976000000 , -1.
→7740300785979] , [ 1088568000000 , 4.4681318138177] , [ 1091246400000 , 7.
→0242541001353] , [ 1093924800000 , 7.5709603667586] , [ 1096516800000 , 20.
→612245065736] , [ 1099195200000 , 21.698065237316] , [ 1101790800000 , 40.
→501189458018] , [ 1104469200000 , 50.464679413194] , [ 1107147600000 , 48.
→917421973355] , [ 1109566800000 , 63.750936549160] , [ 1112245200000 , 59.
→072499126460] , [ 1114833600000 , 43.373158880492] , [ 1117512000000 , 54.
→490918947556] , [ 1120104000000 , 56.661178852079] , [ 1122782400000 , 73.
→450103545496] , [ 1125460800000 , 71.714526354907] , [ 1128052800000 , 85.
→221664349607] , [ 1130734800000 , 77.769261392481] , [ 1133326800000 , 95.
→966528716500] , [ 1136005200000 , 107.59132116397] , [ 1138683600000 , 127.
→25740096723] , [ 1141102800000 , 122.13917498830] , [ 1143781200000 , 126.
→53657279774] , [ 1146369600000 , 132.39300992970] , [ 1149048000000 , 120.
→11238242904] , [ 1151640000000 , 118.41408917750] , [ 1154318400000 , 107.
→92918924621] , [ 1156996800000 , 110.28057249569] , [ 1159588800000 , 117.
→20485334692] , [ 1162270800000 , 141.33556756948] , [ 1164862800000 , 159.
→59452727893] , [ 1167541200000 , 167.09801853304] , [ 1170219600000 , 185.
→46849659215] , [ 1172638800000 , 184.82474099990] , [ 1175313600000 , 195.
→63155213887] , [ 1177905600000 , 207.40597044171] , [ 1180584000000 , 230.
→55966698196] , [ 1183176000000 , 239.55649035292] , [ 1185854400000 , 241.
→35915085208] , [ 1188532800000 , 239.89428956243] , [ 1191124800000 , 260.
→47781917715] , [ 1193803200000 , 276.39457482225] , [ 1196398800000 , 258.
→66530682672] , [ 1199077200000 , 250.98846121893] , [ 1201755600000 , 226.
→89902618127] , [ 1204261200000 , 227.29009273807] , [ 1206936000000 , 218.
→66476654350] , [ 1209528000000 , 232.46605902918] , [ 1212206400000 , 253.
→25667081117] , [ 1214798400000 , 235.82505363925] , [ 1217476800000 , 229.
→70112774254] , [ 1220155200000 , 225.18472705952] , [ 1222747200000 , 189.
→13661746552] , [ 1225425600000 , 149.46533007301] , [ 1228021200000 , 131.
→00340772114] , [ 1230699600000 , 135.18341728866] , [ 1233378000000 , 109.
→15296887173] , [ 1235797200000 , 84.614772549760] , [ 1238472000000 , 100.
→60810015326] , [ 1241064000000 , 141.50134895610] , [ 1243742400000 , 142.
→50405083675] , [ 1246334400000 , 139.81192372672] , [ 1249012800000 , 177.
→78205544583] , [ 1251691200000 , 194.73691933074] , [ 1254283200000 , 209.
→00838460225] , [ 1256961600000 , 198.19855877420] , [ 1259557200000 , 222.
→37102417812] , [ 1262235600000 , 234.24581081250] , [ 1264914000000 , 228.
→26087689346] , [ 1267333200000 , 248.81895126250] , [ 1270008000000 , 270.
→57301075186] , [ 1272600000000 , 292.64604322550] , [ 1275278400000 , 265.
→94088520518] , [ 1277870400000 , 237.82887467569] , [ 1280548800000 , 265.
→55973314204] , [ 1283227200000 , 248.30877330928] , [ 1285819200000 , 278.
→14870066912] , [ 1288497600000 , 292.69260960288] , [ 1291093200000 , 300.
→84263809599] , [ 1293771600000 , 326.17253914628] , [ 12964500000000 , 337.
→69335966505] , [ 1298869200000 , 339.73260965121] , [ 1301544000000 , 346.
→87865120765] , [ 1304136000000 , 347.92991526628] , [ 1306814400000 , 342.
→04627502669] , [ 1309406400000 , 333.45386231233] , [ 1312084800000 , 323.
→15034181243] , [ 1314763200000 , 295.66126882331] , [ 1317355200000 , 251.
→48014579253] , [ 1320033600000 , 295.15424257905] , [ 1322629200000 , 294.
→54766764397] , [ 1325307600000 , 295.72906119051] , [ 1327986000000 , 325.
→73351347613] , [ 1330491600000 , 340.16106061186] , [ 1333166400000 , 345.
→15514071490] , [ 1335758400000 , 337.10259395679] , [ 1338436800000 , 318.
→68216333837] , [ 1341028800000 , 317.03683945246] , [ 1343707200000 , 318.
→53549659997] , [ 1346385600000 , 332.85381464104] , [ 1348977600000 , 337.
→36534373477] , [ 1351656000000 , 350.27872156161] , [ 1354251600000 , 349.
→45128876100]]]

```

```

10
11         mean: 250
12     },
13     {
14         key: "Short",
15         values: [ [ 1083297600000 , -0.77078283705125] , [ 1085976000000 , -1.
→8356366650335] , [ 1088568000000 , -5.3121322073127] , [ 1091246400000 , -4.
→9320975829662] , [ 1093924800000 , -3.9835408823225] , [ 1096516800000 , -6.
→8694685316805] , [ 1099195200000 , -8.4854877428545] , [ 1101790800000 , -15.
→933627197384] , [ 1104469200000 , -15.920980069544] , [ 1107147600000 , -12.
→478685045651] , [ 1109566800000 , -17.297761889305] , [ 1112245200000 , -15.
→247129891020] , [ 1114833600000 , -11.336459046839] , [ 1117512000000 , -13.
→298990907415] , [ 1120104000000 , -16.360027000056] , [ 1122782400000 , -18.
→527929522030] , [ 1125460800000 , -22.176516738685] , [ 1128052800000 , -23.
→309665368330] , [ 1130734800000 , -21.629973409748] , [ 1133326800000 , -24.
→186429093486] , [ 1136005200000 , -29.116707312531] , [ 1138683600000 , -37.
→188037874864] , [ 1141102800000 , -34.689264821198] , [ 1143781200000 , -39.
→505932105359] , [ 1146369600000 , -45.339572492759] , [ 1149048000000 , -43.
→849353192764] , [ 1151640000000 , -45.418353922571] , [ 1154318400000 , -44.
→579281059919] , [ 1156996800000 , -44.027098363370] , [ 1159588800000 , -41.
→261306759439] , [ 1162270800000 , -47.446018534027] , [ 1164862800000 , -53.
→413782948909] , [ 1167541200000 , -50.700723647419] , [ 1170219600000 , -56.
→374090913296] , [ 1172638800000 , -61.754245220322] , [ 1175313600000 , -66.
→246241587629] , [ 1177905600000 , -75.351650899999] , [ 1180584000000 , -81.
→699058262032] , [ 1183176000000 , -82.487023368081] , [ 1185854400000 , -86.
→230055113277] , [ 1188532800000 , -84.746914818507] , [ 1191124800000 , -100.
→77134971977] , [ 1193803200000 , -109.95435565947] , [ 1196398800000 , -99.
→605672965057] , [ 1199077200000 , -99.607249394382] , [ 1201755600000 , -94.
→874614950188] , [ 1204261200000 , -105.35899063105] , [ 1206936000000 , -106.
→01931193802] , [ 1209528000000 , -110.28883571771] , [ 1212206400000 , -119.
→60256203030] , [ 1214798400000 , -115.62201315802] , [ 1217476800000 , -106.
→63824185202] , [ 1220155200000 , -99.848746318951] , [ 1222747200000 , -85.
→631219602987] , [ 1225425600000 , -63.547909262067] , [ 1228021200000 , -59.
→753275364457] , [ 1230699600000 , -63.874977883542] , [ 1233378000000 , -56.
→865697387488] , [ 1235797200000 , -54.285579501988] , [ 1238472000000 , -56.
→474659581885] , [ 1241064000000 , -63.847137745644] , [ 1243742400000 , -68.
→754247867325] , [ 1246334400000 , -69.474257009155] , [ 1249012800000 , -75.
→084828197067] , [ 1251691200000 , -77.101028237237] , [ 1254283200000 , -80.
→454866854387] , [ 1256961600000 , -78.984349952220] , [ 1259557200000 , -83.
→041230807854] , [ 1262235600000 , -84.529748348935] , [ 1264914000000 , -83.
→837470195508] , [ 1267333200000 , -87.174487671969] , [ 1270008000000 , -90.
→342293007487] , [ 1272600000000 , -93.550928464991] , [ 1275278400000 , -85.
→833102140765] , [ 1277870400000 , -79.326501831592] , [ 1280548800000 , -87.
→986196903537] , [ 1283227200000 , -85.397862121771] , [ 1285819200000 , -94.
→738167050020] , [ 1288497600000 , -98.661952897151] , [ 1291093200000 , -99.
→609665952708] , [ 1293771600000 , -103.57099836183] , [ 1296450000000 , -104.
→04353411322] , [ 1298869200000 , -108.21382792587] , [ 1301544000000 , -108.
→74006900920] , [ 1304136000000 , -112.07766650960] , [ 1306814400000 , -109.
→63328199118] , [ 1309406400000 , -106.53578966772] , [ 1312084800000 , -103.
→16480871469] , [ 1314763200000 , -95.945078001828] , [ 1317355200000 , -81.
→226687340874] , [ 1320033600000 , -90.782206596168] , [ 1322629200000 , -89.
→484445370113] , [ 1325307600000 , -88.514723135326] , [ 1327986000000 , -93.
→381292724320] , [ 1330491600000 , -97.529705609172] , [ 1333166400000 , -99.
→520481439189] , [ 1335758400000 , -99.430184898669] , [ 1338436800000 , -93.
→349934521973] , [ 1341028800000 , -95.858475286491] , [ 1343707200000 , -95.
→522755836605] , [ 1346385600000 , -98.503848862036] , [ 1348977600000 , -101.
→49415251896] , [ 1351656000000 , -101.50099325672] , [ 1354251600000 , -99.
→487094927489]]
```

```

16
17         mean: -60
18     },
19     {
20         key: "Gross",
21         mean: 125,
22         values: [ [ 1083297600000, -3.7454058855943], [ 1085976000000, -3.
→6096667436314], [ 1088568000000, -0.8440003934950], [ 1091246400000, 2.
→0921565171691], [ 1093924800000, 3.5874194844361], [ 1096516800000, 13.
→742776534056], [ 1099195200000, 13.212577494462], [ 1101790800000, 24.
→567562260634], [ 1104469200000, 34.543699343650], [ 1107147600000, 36.
→438736927704], [ 1109566800000, 46.453174659855], [ 1112245200000, 43.
→825369235440], [ 1114833600000, 32.036699833653], [ 1117512000000, 41.
→191928040141], [ 1120104000000, 40.301151852023], [ 1122782400000, 54.
→922174023466], [ 1125460800000, 49.538009616222], [ 1128052800000, 61.
→911998981277], [ 1130734800000, 56.139287982733], [ 1133326800000, 71.
→780099623014], [ 1136005200000, 78.474613851439], [ 1138683600000, 90.
→069363092366], [ 1141102800000, 87.449910167102], [ 1143781200000, 87.
→030640692381], [ 1146369600000, 87.053437436941], [ 1149048000000, 76.
→263029236276], [ 1151640000000, 72.995735254929], [ 1154318400000, 63.
→349908186291], [ 1156996800000, 66.253474132320], [ 1159588800000, 75.
→943546587481], [ 1162270800000, 93.889549035453], [ 1164862800000, 106.
→18074433002], [ 1167541200000, 116.39729488562], [ 1170219600000, 129.
→09440567885], [ 1172638800000, 123.07049577958], [ 1175313600000, 129.
→38531055124], [ 1177905600000, 132.05431954171], [ 1180584000000, 148.
→86060871993], [ 1183176000000, 157.06946698484], [ 1185854400000, 155.
→12909573880], [ 1188532800000, 155.14737474392], [ 1191124800000, 159.
→70646945738], [ 1193803200000, 166.44021916278], [ 1196398800000, 159.
→05963386166], [ 1199077200000, 151.38121182455], [ 1201755600000, 132.
→02441123108], [ 1204261200000, 121.93110210702], [ 1206936000000, 112.
→64545460548], [ 1209528000000, 122.17722331147], [ 1212206400000, 133.
→65410878087], [ 1214798400000, 120.20304048123], [ 1217476800000, 123.
→06288589052], [ 1220155200000, 125.33598074057], [ 1222747200000, 103.
→50539786253], [ 1225425600000, 85.917420810943], [ 1228021200000, 71.
→250132356683], [ 1230699600000, 71.308439405118], [ 1233378000000, 52.
→287271484242], [ 1235797200000, 30.329193047772], [ 1238472000000, 44.
→133440571375], [ 1241064000000, 77.654211210456], [ 1243742400000, 73.
→749802969425], [ 1246334400000, 70.337666717565], [ 1249012800000, 102.
→69722724876], [ 1251691200000, 117.63589109350], [ 1254283200000, 128.
→55351774786], [ 1256961600000, 119.21420882198], [ 1259557200000, 139.
→32979337027], [ 1262235600000, 149.71606246357], [ 1264914000000, 144.
→42340669795], [ 1267333200000, 161.64446359053], [ 1270008000000, 180.
→23071774437], [ 1272600000000, 199.09511476051], [ 1275278400000, 180.
→10778306442], [ 1277870400000, 158.50237284410], [ 1280548800000, 177.
→57353623850], [ 1283227200000, 162.91091118751], [ 1285819200000, 183.
→41053361910], [ 1288497600000, 194.03065670573], [ 1291093200000, 201.
→23297214328], [ 1293771600000, 222.60154078445], [ 1296450000000, 233.
→35556801977], [ 1298869200000, 231.22452435045], [ 1301544000000, 237.
→84432503045], [ 1304136000000, 235.55799131184], [ 1306814400000, 232.
→11873570751], [ 1309406400000, 226.62381538123], [ 1312084800000, 219.
→34811113539], [ 1314763200000, 198.69242285581], [ 1317355200000, 168.
→90235629066], [ 1320033600000, 202.64725756733], [ 1322629200000, 203.
→05389378105], [ 1325307600000, 204.85986680865], [ 1327986000000, 229.
→77085616585], [ 1330491600000, 239.65202435959], [ 1333166400000, 242.
→33012622734], [ 1335758400000, 234.11773262149], [ 1338436800000, 221.
→47846307887], [ 1341028800000, 216.98308827912], [ 1343707200000, 218.
→37781386755], [ 1346385600000, 229.39368622736], [ 1348977600000, 230.
→54656412916], [ 1351656000000, 243.06087025523], [ 1354251600000, 244.
→24733578385]]]

```

```

23     },
24     {
25         key: "S&P 1500",
26         values: [ [ 1083297600000 , -1.7798428181819 ] , [ 1085976000000 , -0.
27         ↪36883324836999 ] , [ 1088568000000 , 1.7312581046040 ] , [ 1091246400000 , -1.
28         ↪8356125950460 ] , [ 1093924800000 , -1.5396564170877 ] , [ 1096516800000 , -0.
29         ↪16867791409247 ] , [ 1099195200000 , 1.3754263993413 ] , [ 1101790800000 , 5.
30         ↪8171640898041 ] , [ 1104469200000 , 9.4350145241608 ] , [ 1107147600000 , 6.
31         ↪7649081510160 ] , [ 1109566800000 , 9.1568499314776 ] , [ 1112245200000 , 7.
32         ↪2485090994419 ] , [ 1114833600000 , 4.876222306595 ] , [ 1117512000000 , 8.
33         ↪5992339354652 ] , [ 1120104000000 , 9.0896517982086 ] , [ 1122782400000 , 13.
34         ↪394644048577 ] , [ 1125460800000 , 12.311842010760 ] , [ 1128052800000 , 13.
35         ↪221003650717 ] , [ 1130734800000 , 11.218481009206 ] , [ 1133326800000 , 15.
36         ↪565352598445 ] , [ 1136005200000 , 15.623703865926 ] , [ 1138683600000 , 19.
37         ↪275255326383 ] , [ 1141102800000 , 19.432433717836 ] , [ 1143781200000 , 21.
38         ↪232881244655 ] , [ 1146369600000 , 22.798299192958 ] , [ 1149048000000 , 19.
39         ↪006125095476 ] , [ 1151640000000 , 19.151889158536 ] , [ 1154318400000 , 19.
40         ↪340022855452 ] , [ 1156996800000 , 22.027934841859 ] , [ 1159588800000 , 24.
41         ↪903300681329 ] , [ 1162270800000 , 29.146492833877 ] , [ 1164862800000 , 31.
42         ↪781626082589 ] , [ 1167541200000 , 33.358770738428 ] , [ 1170219600000 , 35.
43         ↪622684613497 ] , [ 1172638800000 , 33.332821711366 ] , [ 1175313600000 , 34.
44         ↪878748635832 ] , [ 1177905600000 , 40.582332613844 ] , [ 1180584000000 , 45.
45         ↪719535502920 ] , [ 1183176000000 , 43.239344722386 ] , [ 1185854400000 , 38.
46         ↪550955100342 ] , [ 1188532800000 , 40.585368816283 ] , [ 1191124800000 , 45.
47         ↪601374057981 ] , [ 1193803200000 , 48.051404337892 ] , [ 1196398800000 , 41.
48         ↪582581696032 ] , [ 1199077200000 , 40.650580792748 ] , [ 1201755600000 , 32.
49         ↪252222066493 ] , [ 1204261200000 , 28.106390258553 ] , [ 1206936000000 , 27.
50         ↪532698196687 ] , [ 1209528000000 , 33.986390463852 ] , [ 1212206400000 , 36.
51         ↪302660526438 ] , [ 1214798400000 , 25.015574480172 ] , [ 1217476800000 , 23.
52         ↪989494069029 ] , [ 1220155200000 , 25.934351445531 ] , [ 1222747200000 , 14.
53         ↪627592011699 ] , [ 1225425600000 , -5.2249403809749 ] , [ 1228021200000 , -12.
54         ↪330933408050 ] , [ 1230699600000 , -11.000291508188 ] , [ 1233378000000 , -18.
55         ↪563864948088 ] , [ 1235797200000 , -27.213097001687 ] , [ 1238472000000 , -20.
56         ↪834133840523 ] , [ 1241064000000 , -12.717886701719 ] , [ 1243742400000 , -8.
57         ↪1644613083526 ] , [ 1246334400000 , -7.9108408918201 ] , [ 1249012800000 , -0.
58         ↪77002391591209 ] , [ 1251691200000 , 2.8243816569672 ] , [ 1254283200000 , 6.
59         ↪8761411421070 ] , [ 1256961600000 , 4.5060912230294 ] , [ 1259557200000 , 10.
60         ↪487179794349 ] , [ 1262235600000 , 13.251375597594 ] , [ 1264914000000 , 9.
61         ↪2207594803415 ] , [ 1267333200000 , 12.836276936538 ] , [ 1270008000000 , 19.
62         ↪816793904978 ] , [ 1272600000000 , 22.156787167211 ] , [ 1275278400000 , 12.
63         ↪518039090576 ] , [ 1277870400000 , 6.4253587440854 ] , [ 1280548800000 , 13.
64         ↪847372028409 ] , [ 1283227200000 , 8.5454736090364 ] , [ 1285819200000 , 18.
65         ↪542801953304 ] , [ 1288497600000 , 23.037064683183 ] , [ 1291093200000 , 23.
66         ↪517422401888 ] , [ 1293771600000 , 31.804723416068 ] , [ 1296450000000 , 34.
67         ↪778247386072 ] , [ 1298869200000 , 39.584883855230 ] , [ 1301544000000 , 40.
68         ↪080647664875 ] , [ 1304136000000 , 44.180050667889 ] , [ 1306814400000 , 42.
69         ↪533535927221 ] , [ 1309406400000 , 40.105374449011 ] , [ 1312084800000 , 37.
70         ↪014659267156 ] , [ 1314763200000 , 29.263745084262 ] , [ 1317355200000 , 19.
71         ↪637463417584 ] , [ 1320033600000 , 33.157645345770 ] , [ 1322629200000 , 32.
72         ↪895053150988 ] , [ 1325307600000 , 34.111544824647 ] , [ 1327986000000 , 40.
73         ↪453985817473 ] , [ 1330491600000 , 46.435700783313 ] , [ 1333166400000 , 51.
74         ↪062385488671 ] , [ 1335758400000 , 50.130448220658 ] , [ 1338436800000 , 41.
75         ↪035476682018 ] , [ 1341028800000 , 46.591932296457 ] , [ 1343707200000 , 48.
76         ↪349391180634 ] , [ 1346385600000 , 51.913011286919 ] , [ 1348977600000 , 55.
77         ↪747238313752 ] , [ 1351656000000 , 52.991824077209 ] , [ 1354251600000 , 49.
78         ↪556311883284 ] ]
27             }
28         ];

```

```

29     }
30   },
31
32   chart3config: {
33     type: Object,
34     value: function() {
35       return {
36         showLegend : true,
37
38         x: function(d) { return d[0] },
39         y: function(d) { return d[1]/100 },
40         color: d3.scale.category10().range(),
41         average: function(d) { return d.mean/100; },
42         duration: 300,
43         clipVoronoi: false,
44         useInteractiveGuideline: true,
45
46         //showcase that the config is idiot-proof
47         donut : true
48       }
49     }
50   },
51
52   chart3JSConfig: {
53     type: Function,
54     value: function() {
55       return function() {
56         this._chart.xAxis
57           .tickFormat(function(d) { return d3.time.format('%m/%d/%y')(new Date(d)) })
58         ;
59
60         this._chart.yAxis
61           .tickFormat(d3.format(',.1%'));
62       }
63     }
64   }
65 } );

```

The result will look like this:

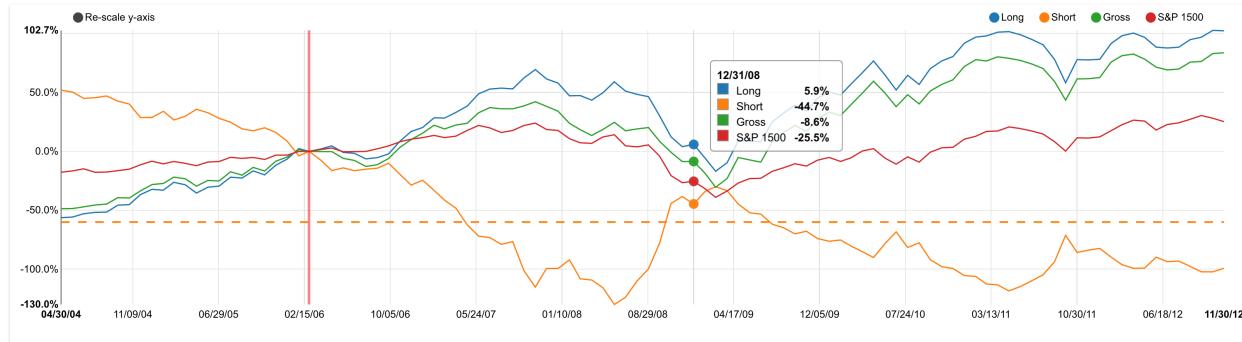


Chart4: A focusable line chart

This line chart features a bar on the bottom the user can use to zoom in or out of the dataset.

Also this chart has been configured with an upper limit on the y axis to prevent very high initial values from scaling down the rest of the chart.

This example will be the first to have data generated on server-side.

C++ code

Edit /src/common/panels/ChartExamples.cc

/src/common/panels/ChartExamples.cc

```
1 void ChartExamples::layout(cgicc::Cgicc& cgi) {
2     remove();
3     setEvent("getChart4", ajax::Eventable::OnClick, this, &ChartExamples::getChart4);
4     add(new ajax::PolymerElement("chart-examples"));
5 }
6
7 void ChartExamples::getChart4(cgicc::Cgicc& cgi, std::ostream& out) {
8     int pix = 0;
9     Json::Value root(Json::arrayValue);
10
11    Json::Value stream0;
12    stream0["area"] = true;
13    stream0["color"] = "#00695C";
14    stream0["key"] = "prime density  $\pi(x)/x$ ";
15    Json::Value stream0values(Json::arrayValue);
16    for (int i = 2; i < 200; i++) {
17        if (ChartExamples::isPrime(i)) {
18            pix += 1;
19        }
20        Json::Value item;
21        item["x"] = i;
22        item["y"] = (float)pix/(float)i;
23        stream0values.append(item);
24    }
25    stream0["values"] = stream0values;
26    root.append(stream0);
27
28    Json::Value stream1;
29    stream1["area"] = false;
30    stream1["color"] = "#E57373";
31    stream1["key"] = "1/(ln(x)-1)";
32    Json::Value stream1values(Json::arrayValue);
33    for (int i = 2; i < 200; i++) {
34        Json::Value item;
35        item["x"] = i;
36        item["y"] = (float)1/(float)(log(i) - 1);
37        stream1values.append(item);
38    }
39    stream1["values"] = stream1values;
40    root.append(stream1);
41
42    out << root;
43 }
44
45 bool ChartExamples::isPrime(int num) {
46     if (num <= 1)
47         return false;
```

```

48     else if (num == 2)
49         return true;
50     else if (num % 2 == 0)
51         return false;
52     else {
53         bool prime = true;
54         int divisor = 3;
55         double num_d = static_cast<double>(num);
56         int upperLimit = static_cast<int>(sqrt(num_d) + 1);
57
58         while (divisor <= upperLimit) {
59             if (num % divisor == 0)
60                 prime = false;
61             divisor += 2;
62         }
63         return prime;
64     }
65 }
```

Edit /include/subsystem/supervisor/panels/ChartExamples.h.

/include/subsystem/supervisor/panels/ChartExamples.h

```

1 namespace subsystempanels {
2     class ChartExamples: public tsframework::CellPanel {
3         private:
4             void getChart4(cgicc::Cgicc& cgi, std::ostream& out);
5             bool isPrime(int num);
6     };
7 }
```

HTML & Javascript

Edit src/html/elements/chart-examples/chart-examples.html

/src/html/elements/chart-examples/chart-examples.html

```

1 <dom-module id="chart-examples">
2     <template>
3
4         <div horizontal layout>
5             <paper-material elevation="1" flex vertical layout>
6                 <ts-ajax data="{{chart4data}}" callback="getChart4" handle-as="json" auto></
7                 <ts-ajax>
8                     <p>
9                         The number of primes with value lower than x ( $\pi(x)$ ) can be approximated using  $\frac{1}{\ln(x)} - 1$ 
10                    </p>
11                    <focus-line-chart flex data="{{chart4data}}" config="{{chart4config}}></
12                    <configure-chart="{{chart4JSConfig}}></focus-line-chart>
13                    </div>
14                </div>
15            </template>
</dom-module>
```

Now edit /src/html/elements/chart-examples/javascript/chart-examples.js

/src/html/elements/chart-examples/javascript/chart-examples.js

```

1 Polymer({
2   properties: {
3     chart4data: {
4       type: Array,
5       value: function() {
6         return [];
7       }
8     },
9   },
10  chart4config: {
11    type: Object,
12    value: function() {
13      return {
14        brushExtent: [2, 100],
15        useInteractiveGuideline: true
16      }
17    }
18  },
19  chart4JSConfig: {
20    type: Function,
21    value: function() {
22      return function() {
23        this._chart.xAxis
24          .tickFormat(d3.format(',f'));
25        this._chart.x2Axis
26          .tickFormat(d3.format(',f'));
27
28        this._chart.yAxis
29          .tickFormat(d3.format(',.2f'));
30        this._chart.y2Axis
31          .tickFormat(d3.format(',.2f'));
32
33        this._chart.yDomain([0.2, 1]);
34      }
35    }
36  }
37 }
38 });
39 ) ;

```

The result will look like this:

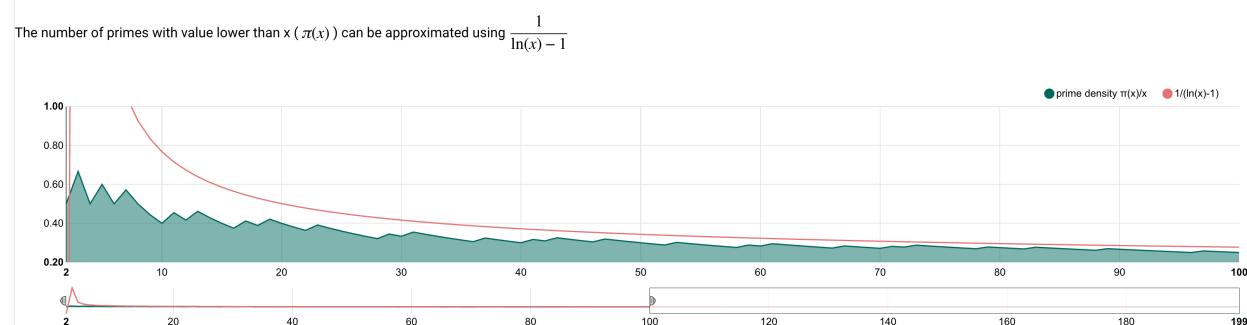


Chart5: Bar charts

For simplicity we will generate the data on client-side again.

In this example we will make a bar chart, a horizontal bar chart, a discrete bar chart, and a historical bar chart.

HTML & Javascript

Edit src/html/elements/chart-examples/chart-examples.html

/src/html/elements/chart-examples/chart-examples.html

```

1  <dom-module id="chart-examples">
2    <template>
3
4      <div vertical layout>
5        <div horizontal layout flex>
6          <paper-material elevation="1" flex horizontal layout>
7            <discrete-bar-chart flex data="{{chart5data}}" config="{{chart5config}}">
8              <!--configure-chart="{{chart5JSConfig}}"--></discrete-bar-chart>
9            </paper-material>
10           <paper-material elevation="1" flex horizontal layout>
11             <historical-bar-chart flex data="{{chart6data}}" config="{{chart6config}}">
12               <!--configure-chart="{{chart6JSConfig}}"--></historical-bar-chart>
13             </paper-material>
14           </div>
15           <div horizontal layout flex>
16             <paper-material elevation="1" flex horizontal layout>
17               <stacked-bar-chart flex data="{{chart5data}}" config="{{chart5config}}">
18                 <!--configure-chart="{{chart5JSConfig}}"--></stacked-bar-chart>
19               </paper-material>
20             <paper-material elevation="1" flex horizontal layout>
21               <horizontal-stacked-bar-chart flex data="{{chart5data}}" config="{
22                 chart5config
23               }" configure-chart="{{chart5JSConfig}}"></horizontal-stacked-bar-
24             chart>
25           </paper-material>
26         </div>
27       </div>
28     </template>
29   </dom-module>
```

Now edit /src/html/elements/chart-examples/javascript/chart-examples.js

/src/html/elements/chart-examples/javascript/chart-examples.js

```

1 Polymer({
2   properties: {
3     chart5data: {
4       type: Array,
5       value: function() {
6         return [
7           { key: "Cumulative Return",
8             values: [
9               { label: "A",
10                 value: 29.765957771107
11               },
12               { label: "B",
```

```
13         "value": 0
14     }, {
15         "label": "C",
16         "value": 32.807804682612
17     }, {
18         "label": "D",
19         "value": 196.45946739256
20     }, {
21         "label": "E",
22         "value": 0.19434030906893
23     }, {
24         "label": "F",
25         "value": 98.079782601442
26     }, {
27         "label": "G",
28         "value": 13.925743130903
29     }, {
30         "label": "H",
31         "value": 5.1387322875705
32     }]
33   ];
34 }
35 },
36
37 chart5config: {
38   type: Object,
39   value: function() {
40     return {
41       duration: 250,
42       x: function(d) { return d.label },
43       y: function(d) { return d.value },
44     }
45   }
46 },
47
48 chart5JSConfig: {
49   type: Function,
50   value: function() {
51     return function() {
52     }
53   }
54 },
55
56 chart6data: {
57   type: Array,
58   value: function() {
59     var sin = [];
60     for (var i = 0; i < 100; i++) {
61       sin.push({x: i, y: Math.sin(i/10) * Math.random() * 100});
62     }
63     return [
64       values: sin,
65       key: "Sine Wave",
66       color: "#ff7f0e"
67     ];
68   }
69 },
```

```

71   chart6config: {
72     type: Object,
73     value: function() {
74       return {
75         margin: {left: 100, bottom: 100},
76         useInteractiveGuideline: true,
77         duration: 250
78       }
79     }
80   },
81
82   chart6JSConfig: {
83     type: Function,
84     value: function() {
85       return function() {
86         this._chart.xAxis
87           .axisLabel("Time (s)")
88           .tickFormat(d3.format(',.1f'));
89         this._chart.yAxis
90           .axisLabel('Voltage (v)')
91           .tickFormat(d3.format(',.2f'));
92         this._chart.showXAxis(true);
93       }
94     }
95   }
96 }
97 } );

```

The result will look like this:



Chart6: Pie charts

HTML & Javascript

Edit src/html/elements/chart-examples/chart-examples.html

/src/html/elements/chart-examples/chart-examples.html

```
1 <dom-module id="chart-examples">
2   <template>
3
4     <div vertical layout>
5       <div horizontal layout flex>
6         <paper-material elevation="1" flex horizontal layout>
7           <pie-chart flex data="{{chart7data}}" config="{{chart7config}}"></pie-chart>
8         </paper-material>
9         <paper-material elevation="1" flex horizontal layout>
10           <pie-chart flex data="{{chart7data}}" config="{{chart7config}}" configure-
11             chart="{{chart7JSConfig}}"></pie-chart>
12           </paper-material>
13         </div>
14       </div>
15   </template>
16 </dom-module>
```

Now edit /src/html/elements/chart-examples/javascript/chart-examples.js

/src/html/elements/chart-examples/javascript/chart-examples.js

```
1 Polymer({
2   properties: {
3     chart7data: {
4       type: Array,
5       value: function() {
6         var sin = [];
7         for (var i = 0; i < 100; i++) {
8           sin.push({x: i, y: Math.sin(i/10) * Math.random() * 100});
9         }
10        return [
11          {key: "One", y: 5},
12          {key: "Two", y: 2},
13          {key: "Three", y: 9},
14          {key: "Four", y: 7},
15          {key: "Five", y: 4},
16          {key: "Six", y: 3},
17          {key: "Seven", y: 0.5}
18        ];
19      }
20    },
21
22    chart7config: {
23      type: Object,
24      value: function() {
25        return {
26          x: function(d) { return d.key },
27          y: function(d) { return d.y },
28          donut: true,
29          padAngle: 0.08,
```

```

30     cornerRadius: 5,
31     labelsOutside: true
32   }
33 }
34 },
35
36 chart7JSConfig: {
37   type: Function,
38   value: function() {
39     return function() {
40       this._chart.pie
41         .startAngle(function(d) { return d.startAngle/2 -Math.PI/2 })
42         .endAngle(function(d) { return d.endAngle/2 -Math.PI/2 });
43     }
44   }
45 }
46
47 }) ;

```

The result will look like this:

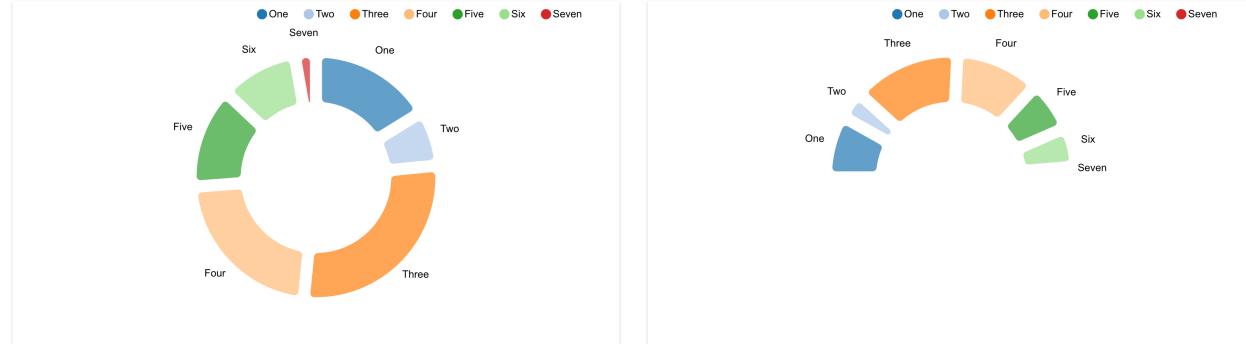


Chart7: Scatter charts

HTML & Javascript

Edit src/html/elements/chart-examples/chart-examples.html

/src/html/elements/chart-examples/chart-examples.html

```

1 <dom-module id="chart-examples">
2   <template>
3
4     <div horizontal layout flex>
5       <paper-material elevation="1" flex horizontal layout>
6         <scatter-chart flex data="{{chart8data}}" config="{{chart8config}}></scatter-chart>
7         </paper-material>
8       </div>
9
10    </template>
11 </dom-module>

```

Now edit /src/html/elements/chart-examples/javascript/chart-examples.js

/src/html/elements/chart-examples/javascript/chart-examples.js

```
1 Polymer({
2   properties: {
3     chart8data: {
4       type: Array,
5       value: function() {
6         var groups = 4;
7         var points = 40;
8         var data = [],
9             shapes = ['thin-x', 'circle', 'cross', 'triangle-up', 'triangle-down',
10            ↵'diamond', 'square'],
11            random = d3.random.normal();
12         for (i = 0; i < groups; i++) {
13           data.push({
14             key: 'Group ' + i,
15             values: []
16           });
17           for (j = 0; j < points; j++) {
18             data[i].values.push({
19               x: random(),
20               y: random(),
21               size: Math.round(Math.random() * 100) / 100,
22               shape: shapes[j % shapes.length]
23             });
24           }
25         }
26       return data;
27     }
28   },
29   chart8config: {
30     type: Object,
31     value: function() {
32       return {
33         showDistX: true,
34         showDistY: true,
35         useVoronoi: true,
36         duration: 300,
37         color: d3.scale.category10().range()
38       }
39     }
40   },
41   chart8JSConfig: {
42     type: Function,
43     value: function() {
44       return function() {
45         nv.utils.symbolMap.set('thin-x', function(size) {
46           size = Math.sqrt(size);
47           return 'M' + (-size/2) + ',' + (-size/2) +
48             '1' + size + ',' + size +
49             'm0,' + -(size) +
50             'l' + (-size) + ',' + size;
51         });
52         this._chart.xAxis
53           .tickFormat(d3.format('.02f'));
54         this._chart.yAxis
55           .tickFormat(d3.format('.02f'));
56       }
57     }
58   }
59 }
```

```

58     }
59   }
60 }
61 );

```

The result will look like this:

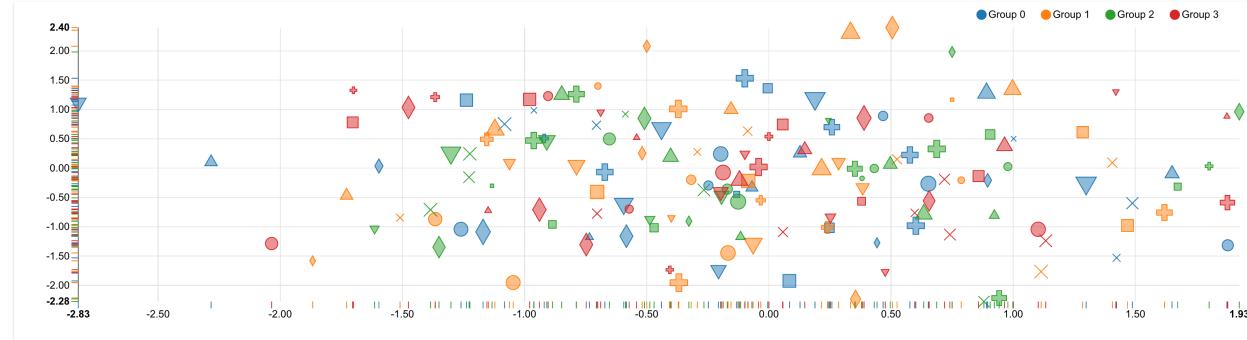


Chart8: Stacked area chart

HTML & Javascript

Edit src/html/elements/chart-examples/chart-examples.html

/src/html/elements/chart-examples/chart-examples.html

```

1 <dom-module id="chart-examples">
2   <template>
3
4     <div horizontal layout flex>
5       <paper-material elevation="1" flex horizontal layout>
6         <stacked-area-chart flex data="{{chart9data}}" config="{{chart9config}}"
7           &configure-chart="{{chart9JSConfig}}></stacked-area-chart>
8         </paper-material>
9       </div>
10
11   </template>
12 </dom-module>

```

Now edit /src/html/elements/chart-examples/javascript/chart-examples.js

/src/html/elements/chart-examples/javascript/chart-examples.js

```

1 Polymer({
2   properties: {
3     chart9data: {
4       type: Array,
5       value: function() {
6         return [
7           { "key": "Consumer Discretionary",
8             "values": [
9               [1138683600000, 27.38478809681],
10              [1141102800000, 27.371377218208],
11              [1143781200000, 26.309915460827],
12              [1146369600000, 26.425199957521],
13              [1149048000000, 26.823411519395],

```

```
14 [115164000000, 23.850443591584],  
15 [115431840000, 23.158355444054],  
16 [1156996800000, 22.998689393694],  
17 [1159588800000, 27.977128511299],  
18 [1162270800000, 29.073672469721],  
19 [1164862800000, 28.587640408904],  
20 [1167541200000, 22.788453687638],  
21 [1170219600000, 22.429199073597],  
22 [1172638800000, 22.324103271051],  
23 [1175313600000, 17.558388444186],  
24 [1177905600000, 16.769518096208],  
25 [1180584000000, 16.214738201302],  
26 [1183176000000, 18.729632971228],  
27 [1185854400000, 18.814523318848],  
28 [1188532800000, 19.789986451358],  
29 [11911124800000, 17.070049054933],  
30 [1193803200000, 16.121349575715],  
31 [1196398800000, 15.141659430091],  
32 [1199077200000, 17.175388025298],  
33 [1201755600000, 17.286592443521],  
34 [1204261200000, 16.323141626569],  
35 [1206936000000, 19.231263773952],  
36 [1209528000000, 18.446256391094],  
37 [1212206400000, 17.822632399764],  
38 [1214798400000, 15.539366475979],  
39 [1217476800000, 15.255131790216],  
40 [1220155200000, 15.660963922593],  
41 [1222747200000, 13.254482273697],  
42 [1225425600000, 11.920796202299],  
43 [1228021200000, 12.122809090925],  
44 [1230699600000, 15.691026271393],  
45 [1233378000000, 14.720881635107],  
46 [1235797200000, 15.387939360044],  
47 [1238472000000, 13.765436672229],  
48 [1241064000000, 14.6314458648],  
49 [1243742400000, 14.292446536221],  
50 [1246334400000, 16.170071367016],  
51 [1249012800000, 15.948135554337],  
52 [1251691200000, 16.612872685134],  
53 [1254283200000, 18.778338719091],  
54 [1256961600000, 16.75602606542],  
55 [1259557200000, 19.385804443147],  
56 [1262235600000, 22.950590240168],  
57 [1264914000000, 23.61159018141],  
58 [1267333200000, 25.708586989581],  
59 [1270008000000, 26.883915999885],  
60 [1272600000000, 25.893486687065],  
61 [1275278400000, 24.678914263176],  
62 [1277870400000, 25.937275793023],  
63 [1280548800000, 29.46138169384],  
64 [1283227200000, 27.357322961862],  
65 [1285819200000, 29.057235285673],  
66 [1288497600000, 28.549434189386],  
67 [1291093200000, 28.506352379723],  
68 [1293771600000, 29.449241421597],  
69 [1296450000000, 25.796838168807],  
70 [1298869200000, 28.740145449189],  
71 [1301544000000, 22.091744141872],
```

```

72      [1304136000000, 25.079662545409],
73      [1306814400000, 23.674906973064],
74      [1309406400000, 23.41800274293],
75      [1312084800000, 23.243644138871],
76      [1314763200000, 31.591854066817],
77      [1317355200000, 31.497112374114],
78      [1320033600000, 26.672380820431],
79      [1322629200000, 27.297080015495],
80      [1325307600000, 20.174315530051],
81      [1327986000000, 19.631084213899],
82      [1330491600000, 20.366462219462],
83      [1333166400000, 17.429019937289],
84      [1335758400000, 16.75543633539],
85      [1338436800000, 16.182906906042]
86    ]
87  },
88  {
89    "key": "Consumer Staples",
90    "values": [
91      [1138683600000, 7.2800122043237],
92      [1141102800000, 7.1187787503354],
93      [1143781200000, 8.351887016482],
94      [1146369600000, 8.4156698763993],
95      [1149048000000, 8.1673298604231],
96      [1151640000000, 5.5132447126042],
97      [1154318400000, 6.1152537710599],
98      [1156996800000, 6.076765091942],
99      [1159588800000, 4.6304473798646],
100     [1162270800000, 4.6301068469402],
101     [1164862800000, 4.3466656309389],
102     [1167541200000, 6.830104897003],
103     [1170219600000, 7.241633040029],
104     [1172638800000, 7.1432372054153],
105     [1175313600000, 10.608942063374],
106     [1177905600000, 10.914964549494],
107     [1180584000000, 10.933223880565],
108     [1183176000000, 8.3457524851265],
109     [1185854400000, 8.1078413081882],
110     [1188532800000, 8.2697185922474],
111     [1191124800000, 8.4742436475968],
112     [1193803200000, 8.4994601179319],
113     [1196398800000, 8.7387319683243],
114     [1199077200000, 6.8829183612895],
115     [1201755600000, 6.984133637885],
116     [1204261200000, 7.0860136043287],
117     [1206936000000, 4.3961787956053],
118     [1209528000000, 3.8699674365231],
119     [1212206400000, 3.6928925238305],
120     [1214798400000, 6.7571718894253],
121     [1217476800000, 6.4367313362344],
122     [1220155200000, 6.4048441521454],
123     [1222747200000, 5.4643833239669],
124     [1225425600000, 5.3150786833374],
125     [1228021200000, 5.3011272612576],
126     [1230699600000, 4.1203601430809],
127     [1233378000000, 4.0881783200525],
128     [1235797200000, 4.1928665957189],
129     [1238472000000, 7.0249415663205],
130     [1241064000000, 7.006530880769],
131   ]
132 }
```

```

130      [1243742400000, 6.994835633224],
131      [1246334400000, 6.1220222336254],
132      [1249012800000, 6.1177436137653],
133      [1251691200000, 6.1413396231981],
134      [1254283200000, 4.8046006145874],
135      [1256961600000, 4.6647600660544],
136      [1259557200000, 4.544865006255],
137      [1262235600000, 6.0488249316539],
138      [1264914000000, 6.3188669540206],
139      [1267333200000, 6.5873958262306],
140      [1270008000000, 6.2281189839578],
141      [1272600000000, 5.8948915746059],
142      [1275278400000, 5.5967320482214],
143      [1277870400000, 0.99784432084837],
144      [1280548800000, 1.0950794175359],
145      [1283227200000, 0.94479734407491],
146      [1285819200000, 1.222093988688],
147      [1288497600000, 1.335093106856],
148      [1291093200000, 1.3302565104985],
149      [1293771600000, 1.340824670897],
150      [1296450000000, 0],
151      [1298869200000, 0],
152      [1301544000000, 0],
153      [1304136000000, 0],
154      [1306814400000, 0],
155      [1309406400000, 0],
156      [1312084800000, 0],
157      [1314763200000, 0],
158      [1317355200000, 4.4583692315],
159      [1320033600000, 3.6493043348059],
160      [1322629200000, 3.8610064091761],
161      [1325307600000, 5.5144800685202],
162      [1327986000000, 5.1750695220792],
163      [1330491600000, 5.6710066952691],
164      [1333166400000, 8.5658461590953],
165      [1335758400000, 8.6135447714243],
166      [1338436800000, 8.0231460925212]
167    ]
168  },
169  {
170    "key": "Energy",
171    "values": [
172      [1138683600000, 1.544303464167],
173      [1141102800000, 1.4387289432421],
174      [1143781200000, 0],
175      [1146369600000, 0],
176      [1149048000000, 0],
177      [1151640000000, 1.328626801128],
178      [1154318400000, 1.2874050802627],
179      [1156996800000, 1.0872743105593],
180      [1159588800000, 0.96042562635813],
181      [1162270800000, 0.93139372870616],
182      [1164862800000, 0.94432167305385],
183      [1167541200000, 1.277750166208],
184      [1170219600000, 1.2204893886811],
185      [1172638800000, 1.207489123122],
186      [1175313600000, 1.2490651414113],
187      [1177905600000, 1.2593129913052],
188      [1180584000000, 1.373329808388],

```

```
188 [1183176000000, 0],  
189 [1185854400000, 0],  
190 [1188532800000, 0],  
191 [1191124800000, 0],  
192 [1193803200000, 0],  
193 [1196398800000, 0],  
194 [1199077200000, 0],  
195 [1201755600000, 0],  
196 [1204261200000, 0],  
197 [1206936000000, 0],  
198 [1209528000000, 0],  
199 [1212206400000, 0],  
200 [1214798400000, 0],  
201 [1217476800000, 0],  
202 [1220155200000, 0],  
203 [1222747200000, 1.4516108933695],  
204 [1225425600000, 1.1856025268225],  
205 [1228021200000, 1.3430470355439],  
206 [1230699600000, 2.2752595354509],  
207 [1233378000000, 2.4031560010523],  
208 [1235797200000, 2.0822430731926],  
209 [1238472000000, 1.5640902826938],  
210 [1241064000000, 1.5812873972356],  
211 [1243742400000, 1.9462448548894],  
212 [1246334400000, 2.9464870223957],  
213 [1249012800000, 3.0744699383222],  
214 [1251691200000, 2.9422304628446],  
215 [1254283200000, 2.7503075599999],  
216 [1256961600000, 2.6506701800427],  
217 [1259557200000, 2.8005425319977],  
218 [1262235600000, 2.6816184971185],  
219 [1264914000000, 2.681206271327],  
220 [1267333200000, 2.8195488011259],  
221 [1270008000000, 0],  
222 [1272600000000, 0],  
223 [1275278400000, 0],  
224 [1277870400000, 1.0687057346382],  
225 [1280548800000, 1.2539400544134],  
226 [1283227200000, 1.1862969445955],  
227 [1285819200000, 0],  
228 [1288497600000, 0],  
229 [1291093200000, 0],  
230 [1293771600000, 0],  
231 [1296450000000, 1.941972859484],  
232 [1298869200000, 2.1142247697552],  
233 [1301544000000, 2.3788590206824],  
234 [1304136000000, 2.5337302877545],  
235 [1306814400000, 2.3163370395199],  
236 [1309406400000, 2.0645451843195],  
237 [1312084800000, 2.1004446672411],  
238 [1314763200000, 3.6301875804303],  
239 [1317355200000, 2.454204664652],  
240 [1320033600000, 2.196082370894],  
241 [1322629200000, 2.3358418255202],  
242 [1325307600000, 0],  
243 [1327986000000, 0],  
244 [1330491600000, 0],  
245 [1333166400000, 0.39001201038526],
```

```
246             [1335758400000, 0.30945472725559],  
247             [1338436800000, 0.31062439305591]  
248         ]  
249     }, {  
250         "key": "Financials",  
251         "values": [  
252             [1138683600000, 13.356778764352],  
253             [1141102800000, 13.611196863271],  
254             [1143781200000, 6.895903006119],  
255             [1146369600000, 6.9939633271352],  
256             [1149048000000, 6.7241510257675],  
257             [1151640000000, 5.5611293669516],  
258             [1154318400000, 5.6086488714041],  
259             [1156996800000, 5.4962849907033],  
260             [1159588800000, 6.9193153169279],  
261             [1162270800000, 7.0016334389777],  
262             [1164862800000, 6.7865422443273],  
263             [1167541200000, 9.0006454225383],  
264             [1170219600000, 9.2233916171431],  
265             [1172638800000, 8.8929316009479],  
266             [1175313600000, 10.345937520404],  
267             [1177905600000, 10.075914677026],  
268             [1180584000000, 10.089006188111],  
269             [1183176000000, 10.598330295008],  
270             [1185854400000, 9.968954653301],  
271             [1188532800000, 9.7740580198146],  
272             [1191124800000, 10.558483060626],  
273             [1193803200000, 9.9314651823603],  
274             [1196398800000, 9.3997715873769],  
275             [1199077200000, 8.4086493387262],  
276             [1201755600000, 8.9698309085926],  
277             [1204261200000, 8.2778357995396],  
278             [1206936000000, 8.8585045600123],  
279             [1209528000000, 8.7013756413322],  
280             [1212206400000, 7.7933605469443],  
281             [1214798400000, 7.0236183483064],  
282             [1217476800000, 6.9873088186829],  
283             [1220155200000, 6.8031713070097],  
284             [1222747200000, 6.6869531315723],  
285             [1225425600000, 6.138256993963],  
286             [1228021200000, 5.6434994016354],  
287             [1230699600000, 5.495220262512],  
288             [1233378000000, 4.6885326869846],  
289             [1235797200000, 4.4524349883438],  
290             [1238472000000, 5.6766520778185],  
291             [1241064000000, 5.7675774480752],  
292             [1243742400000, 5.7882863168337],  
293             [1246334400000, 7.2666010034924],  
294             [1249012800000, 7.519182132226],  
295             [1251691200000, 7.849651451445],  
296             [1254283200000, 10.383992037985],  
297             [1256961600000, 9.0653691861818],  
298             [1259557200000, 9.6705248324159],  
299             [1262235600000, 10.856380561349],  
300             [1264914000000, 11.27452370892],  
301             [1267333200000, 11.754156529088],  
302             [1270008000000, 8.2870811422456],  
303             [1272600000000, 8.0210264360699],
```

```

304     [1275278400000, 7.5375074474865],
305     [1277870400000, 8.3419527338039],
306     [1280548800000, 9.4197471818443],
307     [1283227200000, 8.7321733185797],
308     [1285819200000, 9.6627062648126],
309     [1288497600000, 10.187962234549],
310     [1291093200000, 9.8144201733476],
311     [1293771600000, 10.275723361713],
312     [1296450000000, 16.796066079353],
313     [1298869200000, 17.543254984075],
314     [1301544000000, 16.673660675084],
315     [1304136000000, 17.963944353609],
316     [1306814400000, 16.637740867211],
317     [1309406400000, 15.84857094609],
318     [1312084800000, 14.767303362182],
319     [1314763200000, 24.778452182432],
320     [1317355200000, 18.370353229999],
321     [1320033600000, 15.2531374291],
322     [1322629200000, 14.989600840649],
323     [1325307600000, 16.052539160125],
324     [1327986000000, 16.424390322793],
325     [1330491600000, 17.884020741105],
326     [1333166400000, 7.1424929577921],
327     [1335758400000, 7.8076213051482],
328     [1338436800000, 7.2462684949232]
329   ]
330 },
331   {
332     "key": "Health Care",
333     "values": [
334       [1138683600000, 14.212410956029],
335       [1141102800000, 13.973193618249],
336       [1143781200000, 15.218233920665],
337       [1146369600000, 14.38210972745],
338       [1149048000000, 13.894310878491],
339       [1151640000000, 15.593086090032],
340       [1154318400000, 16.244839695188],
341       [1156996800000, 16.017088850646],
342       [1159588800000, 14.183951830055],
343       [1162270800000, 14.148523245697],
344       [1164862800000, 13.424326059972],
345       [1167541200000, 12.974450435753],
346       [1170219600000, 13.23247041802],
347       [1172638800000, 13.318762655574],
348       [1175313600000, 15.961407746104],
349       [1177905600000, 16.287714639805],
350       [1180584000000, 16.246590583889],
351       [1183176000000, 17.564505594809],
352       [1185854400000, 17.872725373165],
353       [1188532800000, 18.018998508757],
354       [1191124800000, 15.584518016603],
355       [1193803200000, 15.480850647181],
356       [1196398800000, 15.699120036984],
357       [1199077200000, 19.184281817226],
358       [1201755600000, 19.691226605207],
359       [1204261200000, 18.982314051295],
360       [1206936000000, 18.707820309008],
361       [1209528000000, 17.459630929761],
362       [1212206400000, 16.500616076782],
363     ]
364   }

```

```
362     [1214798400000, 18.086324003979],  
363     [1217476800000, 18.929464156258],  
364     [1220155200000, 18.233728682084],  
365     [1222747200000, 16.315776297325],  
366     [1225425600000, 14.63289219025],  
367     [1228021200000, 14.667835024478],  
368     [1230699600000, 13.946993947308],  
369     [1233378000000, 14.394304684397],  
370     [1235797200000, 13.724462792967],  
371     [1238472000000, 10.930879035806],  
372     [1241064000000, 9.8339915513708],  
373     [1243742400000, 10.053858541872],  
374     [1246334400000, 11.786998438287],  
375     [1249012800000, 11.780994901769],  
376     [1251691200000, 11.305889670276],  
377     [1254283200000, 10.918452290083],  
378     [1256961600000, 9.6811395055706],  
379     [1259557200000, 10.971529744038],  
380     [1262235600000, 13.330210480209],  
381     [1264914000000, 14.592637568961],  
382     [1267333200000, 14.605329141157],  
383     [1270008000000, 13.936853794037],  
384     [1272600000000, 12.189480759072],  
385     [1275278400000, 11.676151385046],  
386     [1277870400000, 13.058852800017],  
387     [1280548800000, 13.62891543203],  
388     [1283227200000, 13.811107569918],  
389     [1285819200000, 13.786494560787],  
390     [1288497600000, 14.04516285753],  
391     [1291093200000, 13.697412447288],  
392     [1293771600000, 13.677681376221],  
393     [1296450000000, 19.961511864531],  
394     [1298869200000, 21.049198298158],  
395     [1301544000000, 22.687631094008],  
396     [1304136000000, 25.469010617433],  
397     [1306814400000, 24.883799437121],  
398     [1309406400000, 24.203843814248],  
399     [1312084800000, 22.138760964038],  
400     [1314763200000, 16.034636966228],  
401     [1317355200000, 15.394958944556],  
402     [1320033600000, 12.625642461969],  
403     [1322629200000, 12.973735699739],  
404     [1325307600000, 15.786018336149],  
405     [1327986000000, 15.227368020134],  
406     [1330491600000, 15.899752650734],  
407     [1333166400000, 18.994731295388],  
408     [1335758400000, 18.450055817702],  
409     [1338436800000, 17.863719889669]  
410   ]  
411 }, {  
412   "key": "Industrials",  
413   "values": [  
414     [1138683600000, 7.1590087090398],  
415     [1141102800000, 7.1297210970108],  
416     [1143781200000, 5.5774588290586],  
417     [1146369600000, 5.4977254491156],  
418     [1149048000000, 5.5138153113634],  
419     [1151640000000, 4.3198084032122],
```

```

420 [1154318400000, 3.9179295839125],
421 [1156996800000, 3.8110093051479],
422 [1159588800000, 5.5629020916939],
423 [1162270800000, 5.7241673711336],
424 [1164862800000, 5.4715049695004],
425 [1167541200000, 4.9193763571618],
426 [1170219600000, 5.136053947247],
427 [1172638800000, 5.1327258759766],
428 [1175313600000, 5.1888943925082],
429 [1177905600000, 5.5191481293345],
430 [1180584000000, 5.6093625614921],
431 [1183176000000, 4.2706312987397],
432 [1185854400000, 4.4453235132117],
433 [1188532800000, 4.6228003109761],
434 [1191124800000, 5.0645764756954],
435 [1193803200000, 5.0723447230959],
436 [1196398800000, 5.1457765818846],
437 [1199077200000, 5.4067851597282],
438 [1201755600000, 5.4722419168161],
439 [1204261200000, 5.3742740389688],
440 [1206936000000, 6.251751933664],
441 [1209528000000, 6.1406852153472],
442 [1212206400000, 5.8164385627465],
443 [1214798400000, 5.4255846656171],
444 [1217476800000, 5.3738499417204],
445 [1220155200000, 5.1815627753979],
446 [1222747200000, 5.0305983235349],
447 [1225425600000, 4.6823058607165],
448 [1228021200000, 4.5941481589093],
449 [1230699600000, 5.4669598474575],
450 [1233378000000, 5.1249037357],
451 [1235797200000, 4.3504421250742],
452 [1238472000000, 4.6260881026002],
453 [1241064000000, 5.0140402458946],
454 [1243742400000, 4.7458462454774],
455 [1246334400000, 6.0437019654564],
456 [1249012800000, 6.4595216249754],
457 [1251691200000, 6.6420468254155],
458 [1254283200000, 5.8927271960913],
459 [1256961600000, 5.4712108838003],
460 [1259557200000, 6.1220254207747],
461 [1262235600000, 5.5385935169255],
462 [1264914000000, 5.7383377612639],
463 [1267333200000, 6.1715976730415],
464 [1270008000000, 4.0102262681174],
465 [1272600000000, 3.7693896796921],
466 [1275278400000, 3.5301571031152],
467 [1277870400000, 2.7660252652526],
468 [1280548800000, 3.1409983385775],
469 [1283227200000, 3.0528024863055],
470 [1285819200000, 4.3126123157971],
471 [1288497600000, 4.594654041683],
472 [1291093200000, 4.5424126126793],
473 [1293771600000, 4.7790043987302],
474 [1296450000000, 7.4969154058289],
475 [1298869200000, 7.9424751557821],
476 [1301544000000, 7.1560736250547],
477 [1304136000000, 7.9478117337855],

```

```
478 [1306814400000, 7.4109214848895],  
479 [1309406400000, 7.5966457641101],  
480 [1312084800000, 7.165754444071],  
481 [1314763200000, 5.4816702524302],  
482 [1317355200000, 4.9893656089584],  
483 [1320033600000, 4.498385105327],  
484 [1322629200000, 4.6776090358151],  
485 [1325307600000, 8.1350814368063],  
486 [1327986000000, 8.0732769990652],  
487 [1330491600000, 8.5602340387277],  
488 [1333166400000, 5.1293714074325],  
489 [1335758400000, 5.2586794619016],  
490 [1338436800000, 5.1100853569977]  
491 ]  
492 }, {  
493   "key": "Information Technology",  
494   "values": [  
495     [1138683600000, 13.242301508051],  
496     [1141102800000, 12.863536342042],  
497     [1143781200000, 21.034044171629],  
498     [1146369600000, 21.419084618803],  
499     [1149048000000, 21.142678863691],  
500     [1151640000000, 26.568489677529],  
501     [1154318400000, 24.839144939905],  
502     [1156996800000, 25.456187462167],  
503     [1159588800000, 26.350164502826],  
504     [1162270800000, 26.47833320519],  
505     [1164862800000, 26.425979547847],  
506     [1167541200000, 28.191461582256],  
507     [1170219600000, 28.930307448808],  
508     [1172638800000, 29.521413891117],  
509     [1175313600000, 28.188285966466],  
510     [1177905600000, 27.704619625832],  
511     [1180584000000, 27.490862424829],  
512     [1183176000000, 28.770679721286],  
513     [1185854400000, 29.060480671449],  
514     [1188532800000, 28.240998844973],  
515     [11911124800000, 33.004893194127],  
516     [1193803200000, 34.075180359928],  
517     [1196398800000, 32.548560664833],  
518     [1199077200000, 30.629727432728],  
519     [1201755600000, 28.642858788159],  
520     [1204261200000, 27.973575227842],  
521     [1206936000000, 27.393351882726],  
522     [1209528000000, 28.476095288523],  
523     [1212206400000, 29.296678664261],  
524     [1214798400000, 29.222333802896],  
525     [1217476800000, 28.092966093843],  
526     [1220155200000, 28.107159262922],  
527     [1222747200000, 25.482974832098],  
528     [1225425600000, 21.208115993834],  
529     [1228021200000, 20.295043095268],  
530     [1230699600000, 15.925754618401],  
531     [1233378000000, 17.162864628346],  
532     [1235797200000, 17.084345773174],  
533     [1238472000000, 22.246007102281],  
534     [1241064000000, 24.530543998509],  
535     [1243742400000, 25.084184918242],
```

```

536     [1246334400000, 16.606166527358],
537     [1249012800000, 17.239620011628],
538     [1251691200000, 17.336739127379],
539     [1254283200000, 25.478492475753],
540     [1256961600000, 23.017152085245],
541     [1259557200000, 25.617745423683],
542     [1262235600000, 24.061133998642],
543     [1264914000000, 23.223933318644],
544     [1267333200000, 24.425887263937],
545     [1270008000000, 35.501471156693],
546     [1272600000000, 33.775013878676],
547     [1275278400000, 30.417993630285],
548     [1277870400000, 30.023598978467],
549     [1280548800000, 33.327519522436],
550     [1283227200000, 31.963388450371],
551     [1285819200000, 30.498967232092],
552     [1288497600000, 32.403696817912],
553     [1291093200000, 31.47736071922],
554     [1293771600000, 31.53259666241],
555     [1296450000000, 41.760282761548],
556     [1298869200000, 45.605771243237],
557     [1301544000000, 39.986557966215],
558     [1304136000000, 43.846330510051],
559     [1306814400000, 39.857316881857],
560     [1309406400000, 37.675127768208],
561     [1312084800000, 35.775077970313],
562     [1314763200000, 48.631009702577],
563     [1317355200000, 42.830831754505],
564     [1320033600000, 35.611502589362],
565     [1322629200000, 35.320136981738],
566     [1325307600000, 31.564136901516],
567     [1327986000000, 32.074407502433],
568     [1330491600000, 35.053013769976],
569     [1333166400000, 26.434568573937],
570     [1335758400000, 25.305617871002],
571     [1338436800000, 24.520919418236]
572   ]
573 }, {
574   "key": "Materials",
575   "values": [
576     [1138683600000, 5.5806167415681],
577     [1141102800000, 5.4539047069985],
578     [1143781200000, 7.6728842432362],
579     [1146369600000, 7.719946716654],
580     [1149048000000, 8.0144619912942],
581     [1151640000000, 7.942223133434],
582     [1154318400000, 8.3998279827444],
583     [1156996800000, 8.532324572605],
584     [1159588800000, 4.7324285199763],
585     [1162270800000, 4.7402397487697],
586     [1164862800000, 4.9042069355168],
587     [1167541200000, 5.9583963430882],
588     [1170219600000, 6.3693899239171],
589     [1172638800000, 6.261153903813],
590     [1175313600000, 5.3443942184584],
591     [1177905600000, 5.4932111235361],
592     [1180584000000, 5.5747393101109],
593     [1183176000000, 5.3833633060013],

```

```
594 [1185854400000, 5.5125898831832],  
595 [1188532800000, 5.8116112661327],  
596 [1191124800000, 4.3962296939996],  
597 [1193803200000, 4.6967663605521],  
598 [1196398800000, 4.7963004350914],  
599 [1199077200000, 4.1817985183351],  
600 [1201755600000, 4.3797643870182],  
601 [1204261200000, 4.6966642197965],  
602 [1206936000000, 4.3609995132565],  
603 [1209528000000, 4.4736290996496],  
604 [1212206400000, 4.3749762738128],  
605 [1214798400000, 3.3274661194507],  
606 [1217476800000, 3.0316184691337],  
607 [1220155200000, 2.5718140204728],  
608 [1222747200000, 2.7034994044603],  
609 [1225425600000, 2.2033786591364],  
610 [1228021200000, 1.9850621240805],  
611 [1230699600000, 0],  
612 [1233378000000, 0],  
613 [1235797200000, 0],  
614 [1238472000000, 0],  
615 [1241064000000, 0],  
616 [1243742400000, 0],  
617 [1246334400000, 0],  
618 [1249012800000, 0],  
619 [1251691200000, 0],  
620 [1254283200000, 0.44495950017788],  
621 [1256961600000, 0.33945469262483],  
622 [1259557200000, 0.38348269455195],  
623 [1262235600000, 0],  
624 [1264914000000, 0],  
625 [1267333200000, 0],  
626 [1270008000000, 0],  
627 [1272600000000, 0],  
628 [1275278400000, 0],  
629 [1277870400000, 0],  
630 [1280548800000, 0],  
631 [1283227200000, 0],  
632 [1285819200000, 0],  
633 [1288497600000, 0],  
634 [1291093200000, 0],  
635 [1293771600000, 0],  
636 [1296450000000, 0.52216435716176],  
637 [1298869200000, 0.59275786698454],  
638 [1301544000000, 0],  
639 [1304136000000, 0],  
640 [1306814400000, 0],  
641 [1309406400000, 0],  
642 [1312084800000, 0],  
643 [1314763200000, 0],  
644 [1317355200000, 0],  
645 [1320033600000, 0],  
646 [1322629200000, 0],  
647 [1325307600000, 0],  
648 [1327986000000, 0],  
649 [1330491600000, 0],  
650 [1333166400000, 0],  
651 [1335758400000, 0],
```

```

652     [1338436800000, 0]
653   ]
654 }, {
655   "key": "Telecommunication Services",
656   "values": [
657     [1138683600000, 3.7056975170243],
658     [1141102800000, 3.7561118692318],
659     [1143781200000, 2.861913700854],
660     [1146369600000, 2.9933744103381],
661     [1149048000000, 2.7127537218463],
662     [1151640000000, 3.1195497076283],
663     [1154318400000, 3.4066964004508],
664     [1156996800000, 3.3754571113569],
665     [1159588800000, 2.2965579982924],
666     [1162270800000, 2.4486818633018],
667     [1164862800000, 2.4002308848517],
668     [1167541200000, 1.9649579750349],
669     [1170219600000, 1.9385263638056],
670     [1172638800000, 1.9128975336387],
671     [1175313600000, 2.3412869836298],
672     [1177905600000, 2.4337870351445],
673     [1180584000000, 2.62179703171],
674     [1183176000000, 3.2642864957929],
675     [1185854400000, 3.3200396223709],
676     [1188532800000, 3.3934212707572],
677     [1191124800000, 4.2822327088179],
678     [1193803200000, 4.1474964228541],
679     [1196398800000, 4.1477082879801],
680     [1199077200000, 5.2947122916128],
681     [1201755600000, 5.2919843508028],
682     [1204261200000, 5.1989783050309],
683     [1206936000000, 3.5603057673513],
684     [1209528000000, 3.3009087690692],
685     [1212206400000, 3.1784852603792],
686     [1214798400000, 4.5889503538868],
687     [1217476800000, 4.401779617494],
688     [1220155200000, 4.2208301828278],
689     [1222747200000, 3.89396671475],
690     [1225425600000, 3.0423832241354],
691     [1228021200000, 3.135520611578],
692     [1230699600000, 1.9631418164089],
693     [1233378000000, 1.8963543874958],
694     [1235797200000, 1.8266636017025],
695     [1238472000000, 0.93136635895188],
696     [1241064000000, 0.92737801918888],
697     [1243742400000, 0.97591889805002],
698     [1246334400000, 2.6841193805515],
699     [1249012800000, 2.5664341140531],
700     [1251691200000, 2.3887523699873],
701     [1254283200000, 1.1737801663681],
702     [1256961600000, 1.0953582317281],
703     [1259557200000, 1.2495674976653],
704     [1262235600000, 0.36607452464754],
705     [1264914000000, 0.3548719047291],
706     [1267333200000, 0.36769242398939],
707     [1270008000000, 0],
708     [1272600000000, 0],
709     [1275278400000, 0],

```

```
710     [1277870400000, 0],  
711     [1280548800000, 0],  
712     [1283227200000, 0],  
713     [1285819200000, 0.85450741275337],  
714     [1288497600000, 0.91360317921637],  
715     [1291093200000, 0.89647678692269],  
716     [1293771600000, 0.87800687192639],  
717     [1296450000000, 0],  
718     [1298869200000, 0],  
719     [1301544000000, 0.43668720882994],  
720     [1304136000000, 0.4756523602692],  
721     [1306814400000, 0.46947368328469],  
722     [1309406400000, 0.45138896152316],  
723     [1312084800000, 0.43828726648117],  
724     [1314763200000, 2.0820861395316],  
725     [1317355200000, 0.9364411075395],  
726     [1320033600000, 0.60583907839773],  
727     [1322629200000, 0.61096950747437],  
728     [1325307600000, 0],  
729     [1327986000000, 0],  
730     [1330491600000, 0],  
731     [1333166400000, 0],  
732     [1335758400000, 0],  
733     [1338436800000, 0]  
734     ]  
735 }, {  
736   "key": "Utilities",  
737   "values": [  
738     [1138683600000, 0],  
739     [1141102800000, 0],  
740     [1143781200000, 0],  
741     [1146369600000, 0],  
742     [1149048000000, 0],  
743     [1151640000000, 0],  
744     [1154318400000, 0],  
745     [1156996800000, 0],  
746     [1159588800000, 0],  
747     [1162270800000, 0],  
748     [1164862800000, 0],  
749     [1167541200000, 0],  
750     [1170219600000, 0],  
751     [1172638800000, 0],  
752     [1175313600000, 0],  
753     [1177905600000, 0],  
754     [1180584000000, 0],  
755     [1183176000000, 0],  
756     [1185854400000, 0],  
757     [1188532800000, 0],  
758     [1191124800000, 0],  
759     [1193803200000, 0],  
760     [1196398800000, 0],  
761     [1199077200000, 0],  
762     [1201755600000, 0],  
763     [1204261200000, 0],  
764     [1206936000000, 0],  
765     [1209528000000, 0],  
766     [1212206400000, 0],  
767     [1214798400000, 0],
```

```

768     [1217476800000, 0],
769     [1220155200000, 0],
770     [1222747200000, 0],
771     [1225425600000, 0],
772     [1228021200000, 0],
773     [1230699600000, 0],
774     [1233378000000, 0],
775     [1235797200000, 0],
776     [1238472000000, 0],
777     [1241064000000, 0],
778     [1243742400000, 0],
779     [1246334400000, 0],
780     [1249012800000, 0],
781     [1251691200000, 0],
782     [1254283200000, 0],
783     [1256961600000, 0],
784     [1259557200000, 0],
785     [1262235600000, 0],
786     [1264914000000, 0],
787     [1267333200000, 0],
788     [1270008000000, 0],
789     [1272600000000, 0],
790     [1275278400000, 0],
791     [1277870400000, 0],
792     [1280548800000, 0],
793     [1283227200000, 0],
794     [1285819200000, 0],
795     [1288497600000, 0],
796     [1291093200000, 0],
797     [1293771600000, 0],
798     [1296450000000, 0],
799     [1298869200000, 0],
800     [1301544400000, 0],
801     [1304136000000, 0],
802     [1306814400000, 0],
803     [1309406400000, 0],
804     [1312084800000, 0],
805     [1314763200000, 0],
806     [1317355200000, 0],
807     [1320033600000, 0],
808     [1322629200000, 0],
809     [1325307600000, 0],
810     [1327986000000, 0],
811     [1330491600000, 0],
812     [1333166400000, 0],
813     [1335758400000, 0],
814     [1338436800000, 0]
815   ],
816 }
817 }
818 },
819
820 chart9config: {
821   type: Object,
822   value: function() {
823     return {
824       useInteractiveGuideline: true,
825       x: function(d) { return d[0] },

```

```

826         y: function(d) { return d[1] },
827         duration: 300,
828         controlLabels: {stacked: "Stacked"}
829     }
830 }
831 },
832
833 chart9JSConfig: {
834     type: Function,
835     value: function() {
836         return function() {
837             this._chart.xAxis
838                 .tickFormat(function(d) { return d3.time.format('%x')(new Date(d)) });
839             this._chart.yAxis
840                 .tickFormat(d3.format(',.4f'));
841             this._chart.legend.vers('furious');
842         }
843     }
844 }
845
846 } );

```

The result will look like this:

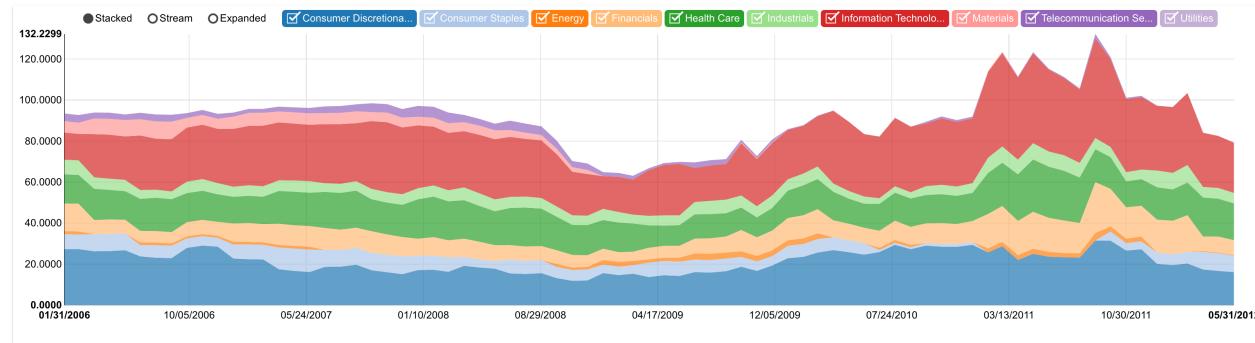


Chart9: A Candlestick charts

HTML & Javascript

Edit src/html/elements/chart-examples/chart-examples.html

/src/html/elements/chart-examples/chart-examples.html

```

1 <dom-module id="chart-examples">
2   <template>
3
4     <div horizontal layout flex>
5       <paper-material elevation="1" flex horizontal layout>
6         <candlestick-chart flex data="{{chart11data}}" config="{{chart11config}}"\_
7           configure-chart="{{chart11JSConfig}}></candlestick-chart>
8         </paper-material>
9       </div>
10
11   </template>
</dom-module>

```

Now edit /src/html/elements/chart-examples/javascript/chart-examples.js

/src/html/elements/chart-examples/javascript/chart-examples.js

```

1 Polymer({
2   properties: {
3     chart11data: {
4       type: Array,
5       value: function() {
6         return [
7           values: [
8             {
9               "date": 15854,
10              "open": 165.42,
11              "high": 165.8,
12              "low": 164.34,
13              "close": 165.22,
14              "volume": 160363400,
15              "adjusted": 164.35
16            },
17            {
18              "date": 15855,
19              "open": 165.35,
20              "high": 166.59,
21              "low": 165.22,
22              "close": 165.83,
23              "volume": 107793800,
24              "adjusted": 164.96
25            },
26            {
27              "date": 15856,
28              "open": 165.37,
29              "high": 166.31,
30              "low": 163.13,
31              "close": 163.45,
32              "volume": 176850100,
33              "adjusted": 162.59
34            },
35            {
36              "date": 15859,
37              "open": 163.83,
38              "high": 164.46,
39              "low": 162.66,
40              "close": 164.35,
41              "volume": 168390700,
42              "adjusted": 163.48
43            },
44            {
45              "date": 15860,
46              "open": 164.44,
47              "high": 165.1,
48              "low": 162.73,
49              "close": 163.56,
50              "volume": 157631500,
51              "adjusted": 162.7
52            },
53            {
54              "date": 15861,
55              "open": 163.09,
56              "high": 163.42,
57              "low": 161.13,
58              "close": 161.27,
59              "volume": 211737800,
60              "adjusted": 160.42
61            }
62          ]
63        }
64      }
65    }
66  }
67)

```

```
56      "date": 15862,
57      "open": 161.2,
58      "high": 162.74,
59      "low": 160.25,
60      "close": 162.73,
61      "volume": 200225500,
62      "adjusted": 161.87
63    },
64    {
65      "date": 15863,
66      "open": 163.85,
67      "high": 164.95,
68      "low": 163.14,
69      "close": 164.8,
70      "volume": 188337800,
71      "adjusted": 163.93
72    },
73    {
74      "date": 15866,
75      "open": 165.31,
76      "high": 165.4,
77      "low": 164.37,
78      "close": 164.8,
79      "volume": 105667100,
80      "adjusted": 163.93
81    },
82    {
83      "date": 15867,
84      "open": 163.3,
85      "high": 164.54,
86      "low": 162.74,
87      "close": 163.1,
88      "volume": 159505400,
89      "adjusted": 162.24
90    },
91    {
92      "date": 15868,
93      "open": 164.22,
94      "high": 164.39,
95      "low": 161.6,
96      "close": 161.75,
97      "volume": 177361500,
98      "adjusted": 160.9
99    },
100   {
101     "date": 15869,
102     "open": 161.66,
103     "high": 164.5,
104     "low": 161.3,
105     "close": 164.21,
106     "volume": 163587800,
107     "adjusted": 163.35
108   },
109   {
110     "date": 15870,
111     "open": 164.03,
112     "high": 164.67,
113     "low": 162.91,
```

```

114      "high": 165.22,
115      "low": 163.22,
116      "close": 164.44,
117      "volume": 136295600,
118      "adjusted": 163.57
119    },
120    {
121      "date": 15874,
122      "open": 164.53,
123      "high": 165.99,
124      "low": 164.52,
125      "close": 165.74,
126      "volume": 114695600,
127      "adjusted": 164.87
128    },
129    {
130      "date": 15875,
131      "open": 165.6,
132      "high": 165.89,
133      "low": 163.38,
134      "close": 163.45,
135      "volume": 206149500,
136      "adjusted": 162.59
137    },
138    {
139      "date": 15876,
140      "open": 161.86,
141      "high": 163.47,
142      "low": 158.98,
143      "close": 159.4,
144      "volume": 321255900,
145      "adjusted": 158.56
146    },
147    {
148      "date": 15877,
149      "open": 159.64,
150      "high": 159.76,
151      "low": 157.47,
152      "close": 159.07,
153      "volume": 271956800,
154      "adjusted": 159.07
155    },
156    {
157      "date": 15880,
158      "open": 157.41,
159      "high": 158.43,
160      "low": 155.73,
161      "close": 157.06,
162      "volume": 222329000,
163      "adjusted": 157.06
164    },
165    {
166      "date": 15881,
167      "open": 158.48,
168      "high": 160.1,
169      "low": 157.42,
170      "close": 158.57,
171      "volume": 162262200,
172      "adjusted": 158.57
173    },
174    {
175      "date": 15882,
176      "open": 159.87,
177      "high": 160.5,
178      "low": 159.25,
179      "close": 160.5,
180      "volume": 162262200,
181      "adjusted": 159.87
182    }
  
```

```
172     "close": 160.14,
173     "volume": 134848000,
174     "adjusted": 160.14
175   },
176   {
177     "date": 15883,
178     "open": 161.1,
179     "high": 161.82,
180     "low": 160.95,
181     "close": 161.08,
182     "volume": 129483700,
183     "adjusted": 161.08
184   },
185   {
186     "date": 15884,
187     "open": 160.63,
188     "high": 161.4,
189     "low": 159.86,
190     "close": 160.42,
191     "volume": 160402900,
192     "adjusted": 160.42
193   },
194   {
195     "date": 15887,
196     "open": 161.26,
197     "high": 162.48,
198     "low": 161.08,
199     "close": 161.36,
200     "volume": 131954800,
201     "adjusted": 161.36
202   },
203   {
204     "date": 15888,
205     "open": 161.12,
206     "high": 162.3,
207     "low": 160.5,
208     "close": 161.21,
209     "volume": 154863700,
210     "adjusted": 161.21
211   },
212   {
213     "date": 15889,
214     "open": 160.48,
215     "high": 161.77,
216     "low": 160.22,
217     "close": 161.28,
218     "volume": 75216400,
219     "adjusted": 161.28
220   },
221   {
222     "date": 15891,
223     "open": 162.47,
224     "high": 163.08,
225     "low": 161.3,
226     "close": 163.02,
227     "volume": 122416900,
228     "adjusted": 163.02
229   },
230   {
231     "date": 15894,
232     "open": 163.86,
233     "high": 164.39,
234     "low": 163.08,
235     "close": 163.95,
236     "volume": 108092500,
```

```
230     "adjusted": 163.95
231 }, {
232     "date": 15895,
233     "open": 164.98,
234     "high": 165.33,
235     "low": 164.27,
236     "close": 165.13,
237     "volume": 119298000,
238     "adjusted": 165.13
239 }, {
240     "date": 15896,
241     "open": 164.97,
242     "high": 165.75,
243     "low": 164.63,
244     "close": 165.19,
245     "volume": 121410100,
246     "adjusted": 165.19
247 }, {
248     "date": 15897,
249     "open": 167.11,
250     "high": 167.61,
251     "low": 165.18,
252     "close": 167.44,
253     "volume": 135592200,
254     "adjusted": 167.44
255 }, {
256     "date": 15898,
257     "open": 167.39,
258     "high": 167.93,
259     "low": 167.13,
260     "close": 167.51,
261     "volume": 104212700,
262     "adjusted": 167.51
263 }, {
264     "date": 15901,
265     "open": 167.97,
266     "high": 168.39,
267     "low": 167.68,
268     "close": 168.15,
269     "volume": 69450600,
270     "adjusted": 168.15
271 }, {
272     "date": 15902,
273     "open": 168.26,
274     "high": 168.36,
275     "low": 167.07,
276     "close": 167.52,
277     "volume": 88702100,
278     "adjusted": 167.52
279 }, {
280     "date": 15903,
281     "open": 168.16,
282     "high": 168.48,
283     "low": 167.73,
284     "close": 167.95,
285     "volume": 92873900,
286     "adjusted": 167.95
287 }, {
```

```
288     "date": 15904,
289     "open": 168.31,
290     "high": 169.27,
291     "low": 168.2,
292     "close": 168.87,
293     "volume": 103620100,
294     "adjusted": 168.87
295   },
296   {
297     "date": 15905,
298     "open": 168.52,
299     "high": 169.23,
300     "low": 168.31,
301     "close": 169.17,
302     "volume": 103831700,
303     "adjusted": 169.17
304   },
305   {
306     "date": 15908,
307     "open": 169.41,
308     "high": 169.74,
309     "low": 169.01,
310     "close": 169.5,
311     "volume": 79428600,
312     "adjusted": 169.5
313   },
314   {
315     "date": 15909,
316     "open": 169.8,
317     "high": 169.83,
318     "low": 169.05,
319     "close": 169.14,
320     "volume": 80829700,
321     "adjusted": 169.14
322   },
323   {
324     "date": 15910,
325     "open": 169.79,
326     "high": 169.86,
327     "low": 168.18,
328     "close": 168.52,
329     "volume": 112914000,
330     "adjusted": 168.52
331   },
332   {
333     "date": 15911,
334     "open": 168.22,
335     "high": 169.08,
336     "low": 167.94,
337     "close": 168.93,
338     "volume": 111088600,
339     "adjusted": 168.93
340   },
341   {
342     "date": 15912,
343     "open": 168.22,
344     "high": 169.16,
345     "low": 167.52,
```

```
346     "high": 169.06,
347     "low": 168.11,
348     "close": 168.59,
349     "volume": 79695000,
350     "adjusted": 168.59
351   },
352   {
353     "date": 15916,
354     "open": 169.1,
355     "high": 169.28,
356     "low": 168.19,
357     "close": 168.59,
358     "volume": 85209600,
359     "adjusted": 168.59
360   },
361   {
362     "date": 15917,
363     "open": 168.94,
364     "high": 169.85,
365     "low": 168.49,
366     "close": 168.71,
367     "volume": 142388700,
368     "adjusted": 168.71
369   },
370   {
371     "date": 15918,
372     "open": 169.99,
373     "high": 170.81,
374     "low": 169.9,
375     "close": 170.66,
376     "volume": 110438400,
377     "adjusted": 170.66
378   },
379   {
380     "date": 15919,
381     "open": 170.28,
382     "high": 170.97,
383     "low": 170.05,
384     "close": 170.95,
385     "volume": 91116700,
386     "adjusted": 170.95
387   },
388   {
389     "date": 15922,
390     "open": 170.57,
391     "high": 170.96,
392     "low": 170.35,
393     "close": 170.7,
394     "volume": 54072700,
395     "adjusted": 170.7
396   },
397   {
398     "date": 15923,
399     "open": 170.37,
400     "high": 170.74,
401     "low": 169.35,
402     "close": 169.73,
403     "volume": 87495000,
404     "adjusted": 169.73
405   },
406   {
407     "date": 15924,
408     "open": 169.19,
409     "high": 169.43,
410     "low": 168.55,
```

```
404     "close": 169.18,
405     "volume": 84854700,
406     "adjusted": 169.18
407   },
408   {
409     "date": 15925,
410     "open": 169.98,
411     "high": 170.18,
412     "low": 168.93,
413     "close": 169.8,
414     "volume": 102181300,
415     "adjusted": 169.8
416   },
417   {
418     "date": 15926,
419     "open": 169.58,
420     "high": 170.1,
421     "low": 168.72,
422     "close": 169.31,
423     "volume": 91757700,
424     "adjusted": 169.31
425   },
426   {
427     "date": 15929,
428     "open": 168.46,
429     "high": 169.31,
430     "low": 168.38,
431     "close": 169.11,
432     "volume": 68593300,
433     "adjusted": 169.11
434   },
435   {
436     "date": 15930,
437     "open": 169.41,
438     "high": 169.9,
439     "low": 168.41,
440     "close": 169.61,
441     "volume": 80806000,
442     "adjusted": 169.61
443   },
444   {
445     "date": 15931,
446     "open": 169.53,
447     "high": 169.8,
448     "low": 168.7,
449     "close": 168.74,
450     "volume": 79829200,
451     "adjusted": 168.74
452   },
453   {
454     "date": 15932,
455     "open": 167.41,
456     "high": 167.43,
457     "low": 166.09,
458     "close": 166.38,
459     "volume": 152931800,
460     "adjusted": 166.38
461   },
462   {
463     "date": 15933,
464     "open": 166.06,
465     "high": 166.63,
466     "low": 165.5,
467     "close": 165.83,
468     "volume": 130868200,
```

```
462     "adjusted": 165.83
463 }, {
464     "date": 15936,
465     "open": 165.64,
466     "high": 166.21,
467     "low": 164.76,
468     "close": 164.77,
469     "volume": 96437600,
470     "adjusted": 164.77
471 }, {
472     "date": 15937,
473     "open": 165.04,
474     "high": 166.2,
475     "low": 164.86,
476     "close": 165.58,
477     "volume": 89294400,
478     "adjusted": 165.58
479 }, {
480     "date": 15938,
481     "open": 165.12,
482     "high": 166.03,
483     "low": 164.19,
484     "close": 164.56,
485     "volume": 159530500,
486     "adjusted": 164.56
487 }, {
488     "date": 15939,
489     "open": 164.9,
490     "high": 166.3,
491     "low": 164.89,
492     "close": 166.06,
493     "volume": 101471400,
494     "adjusted": 166.06
495 }, {
496     "date": 15940,
497     "open": 166.55,
498     "high": 166.83,
499     "low": 165.77,
500     "close": 166.62,
501     "volume": 90888900,
502     "adjusted": 166.62
503 }, {
504     "date": 15943,
505     "open": 166.79,
506     "high": 167.3,
507     "low": 165.89,
508     "close": 166,
509     "volume": 89702100,
510     "adjusted": 166
511 }, {
512     "date": 15944,
513     "open": 164.36,
514     "high": 166,
515     "low": 163.21,
516     "close": 163.33,
517     "volume": 158619400,
518     "adjusted": 163.33
519 }, {
```

```
520         "date": 15945,
521         "open": 163.26,
522         "high": 164.49,
523         "low": 163.05,
524         "close": 163.91,
525         "volume": 108113000,
526         "adjusted": 163.91
527     }, {
528         "date": 15946,
529         "open": 163.55,
530         "high": 165.04,
531         "low": 163.4,
532         "close": 164.17,
533         "volume": 119200500,
534         "adjusted": 164.17
535     }, {
536         "date": 15947,
537         "open": 164.51,
538         "high": 164.53,
539         "low": 163.17,
540         "close": 163.65,
541         "volume": 134560800,
542         "adjusted": 163.65
543     }, {
544         "date": 15951,
545         "open": 165.23,
546         "high": 165.58,
547         "low": 163.7,
548         "close": 164.39,
549         "volume": 142322300,
550         "adjusted": 164.39
551     }, {
552         "date": 15952,
553         "open": 164.43,
554         "high": 166.03,
555         "low": 164.13,
556         "close": 165.75,
557         "volume": 97304000,
558         "adjusted": 165.75
559     }, {
560         "date": 15953,
561         "open": 165.85,
562         "high": 166.4,
563         "low": 165.73,
564         "close": 165.96,
565         "volume": 62930500,
566         "adjusted": 165.96
567     }
568   } ],
569 }
570 },
571
572 chart11config: {
573   type: Object,
574   value: function() {
575     return {
576       x: function(d) { return d['date'] },
577       y: function(d) { return d['close'] },

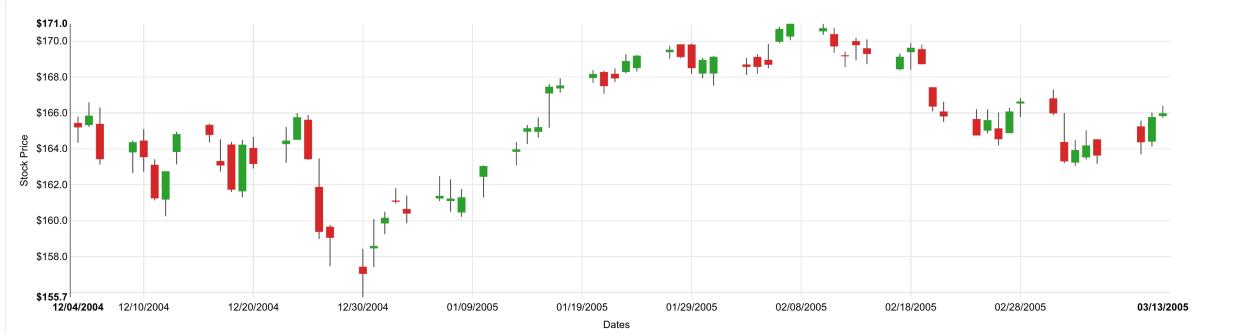
```

```

578     duration: 250,
579     margin: {left: 75, bottom: 50}
580   }
581 }
582 },
583
584 chart11JSConfig: {
585   type: Function,
586   value: function() {
587     return function() {
588       this._chart.xAxis
589         .axisLabel("Dates")
590         .tickFormat(function(d) {
591           // I didn't feel like changing all the above date values
592           // so I hack it to make each value fall on a different date
593           return d3.time.format('%x')(new Date(new Date() - (20000 * 86400000)_
594           ←+ (d * 86400000)));
595         });
596       this._chart.yAxis
597         .axisLabel('Stock Price')
598         .tickFormat(function(d,i){ return '$' + d3.format(',.1f')(d); });
599     }
600   }
601 }
602 );

```

The result will look like this:



1.2.10 Demo 7: Using the theme

Make the theme-demo element

In your cell, run:

```

1 cd src/html/elements
2 ./new-element.js
3 name of the new element: theme-demo
4 creating new element <theme-demo>...
5 removing any .svn folders in theme-demo
6 Finished

```

You now have a working *theme-demo* element. We'll edit it soon.

Register the theme-demo element

Edit src/html/elements/elements.html and add the following line

```
1 <link rel="import" href="theme-demo/theme-demo.html">
```

This will tell AjaXell to load our new element.

Edit the theme-demo element

Edit src/html/elements/theme-demo/theme-demo.html

```
1 <link rel="import" href="/extern/bower_components/polymer/polymer.html">
2 <link rel="import" href="/ts/common-elements/reset-css/reset-css.html">
3 <link rel="import" href="/ts/common-elements/iron-flex-layout-attributes/iron-flex-
4   ↴layout-attributes.html">
5
6 <link rel="import" href="/extern/bower_components/paper-button/paper-button.html">
7 <link rel="import" href="/extern/bower_components/iron-icon/iron-icon.html">
8 <link rel="import" href="/extern/bower_components/iron-icons/iron-icons.html">
9 <link rel="import" href="/extern/bower_components/paper-fab/paper-fab.html">
10
11 <link rel="import" href="/extern/bower_components/paper-checkbox/paper-checkbox.html">
12
13 <link rel="import" href="/extern/bower_components/paper-dropdown-menu/paper-dropdown-
14   ↴menu.html">
15 <link rel="import" href="/extern/bower_components/paper-material/paper-material.html">
16 <link rel="import" href="/extern/bower_components/paper-listbox/paper-listbox.html">
17 <link rel="import" href="/extern/bower_components/paper-menu/paper-menu.html">
18 <link rel="import" href="/extern/bower_components/paper-item/paper-item.html">
19 <link rel="import" href="/extern/bower_components/paper-icon-button/paper-icon-button.
20   ↴html">
21
22 <link rel="import" href="/extern/bower_components/paper-input/paper-input.html">
23 <link rel="import" href="/extern/bower_components/paper-input/paper-textarea.html">
24 <link rel="import" href="/extern/bower_components/paper-radio-button/paper-radio-
25   ↴button.html">
26 <link rel="import" href="/extern/bower_components/paper-radio-group/paper-radio-group.
27   ↴html">
28 <link rel="import" href="/extern/bower_components/paper-toggle-button/paper-toggle-
29   ↴button.html">
30
31 <link rel="import" href="/extern/bower_components/paper-progress/paper-progress.html">
32 <link rel="import" href="/extern/bower_components/paper-spinner/paper-spinner.html">
33
34 <link rel="import" href="/extern/bower_components/paper-tabs/paper-tabs.html">
35 <link rel="import" href="/extern/bower_components/paper-tabs/paper-tab.html">
36
37 <link rel="import" href="/extern/bower_components/paper-tooltip/paper-tooltip.html">
38 <dom-module id="theme-demo">
39   <template>
40     <style include="reset-css"></style>
41     <style include="iron-flex-layout-attributes"></style>
42
43     <link rel="stylesheet" type="text/css" href="css/theme-demo-min.css?__inline=true
44   ↴">
```

```

40
41 <h1>Buttons</h1>
42 <div horizontal layout>
43   <paper-button>default</paper-button>
44   <paper-button disabled>disabled</paper-button>
45   <paper-button noink>noink</paper-button>
46   <paper-button><iron-icon icon="polymer"></iron-icon> icon</paper-button>
47   <paper-button primary>primary</paper-button>
48   <paper-button secondary>secondary</paper-button>
49   <paper-button info>info</paper-button>
50   <paper-button warning>warning</paper-button>
51   <paper-button error>error</paper-button>
52 </div>
53 <div horizontal layout>
54   <paper-button raised>default</paper-button>
55   <paper-button raised disabled>disabled</paper-button>
56   <paper-button raised noink>noink</paper-button>
57   <paper-button raised><iron-icon icon="polymer"></iron-icon> icon</paper-button>
58   <paper-button raised primary>primary</paper-button>
59   <paper-button raised secondary>secondary</paper-button>
60   <paper-button raised info>info</paper-button>
61   <paper-button raised warning>warning</paper-button>
62   <paper-button raised error>error</paper-button>
63 </div>
64 <div horizontal layout>
65   <paper-button toggles>default</paper-button>
66   <paper-button toggles disabled>disabled</paper-button>
67   <paper-button toggles noink>noink</paper-button>
68   <paper-button toggles><iron-icon icon="polymer"></iron-icon> icon</paper-button>
69   <paper-button toggles primary>primary</paper-button>
70   <paper-button toggles secondary>secondary</paper-button>
71   <paper-button toggles info>info</paper-button>
72   <paper-button toggles warning>warning</paper-button>
73   <paper-button toggles error>error</paper-button>
74 </div>
75 <div horizontal layout>
76   <paper-fab icon="arrow-forward" title="arrow-forward"></paper-fab>
77   <paper-fab disabled icon="create" title="create"></paper-fab>
78   <paper-fab secondary icon="favorite" title="heart"></paper-fab>
79   <paper-fab info mini icon="reply" title="reply"></paper-fab>
80   <paper-fab warning mini icon="reply" title="reply"></paper-fab>
81   <paper-fab error mini icon="reply" title="reply"></paper-fab>
82 </div>
83
84 <h1>Checkboxes</h1>
85 <p>
86   These demonstrate the ability to customize the theme per-panel.
87 </p>
88 <div horizontal layout>
89   <div vertical layout>
90     <paper-checkbox>Default unchecked</paper-checkbox>
91     <paper-checkbox pink>Unchecked with theme override</paper-checkbox>
92     <paper-checkbox checked>Default checked</paper-checkbox>
93     <paper-checkbox pink checked>Checked with theme override</paper-checkbox>
94   </div>
95   <div vertical layout>
96     <paper-checkbox disabled>Default unchecked</paper-checkbox>
97     <paper-checkbox disabled pink>Unchecked with theme override</paper-checkbox>

```

```
98      <paper-checkbox disabled checked>Default checked</paper-checkbox>
99      <paper-checkbox disabled pink checked>Checked with theme override</paper-
100     ↵checkbox>
101   </div>
102 </div>
103
104 <h1>Menus</h1>
105 <div horizontal layout>
106   <paper-dropdown-menu label="Dinosaurs">
107     <paper-listbox class="dropdown-content" selected="1">
108       <paper-item>allosaurus</paper-item>
109       <paper-item>brontosaurus</paper-item>
110       <paper-item>carcharodontosaurus</paper-item>
111       <paper-item>diploDOCUS</paper-item>
112     </paper-listbox>
113   </paper-dropdown-menu>
114 </div>
115 <div horizontal layout>
116   <paper-material elevation="1">
117     <paper-listbox>
118       <paper-item>Item 1</paper-item>
119       <paper-item>Item 2</paper-item>
120     </paper-listbox>
121   </paper-material>
122   <paper-material elevation="1">
123     <paper-listbox selected="0">
124       <paper-item>Item 1</paper-item>
125       <paper-item>Item 2</paper-item>
126     </paper-listbox>
127   </paper-material>
128   <paper-material elevation="1">
129     <paper-listbox multi>
130       <paper-item>Item 1</paper-item>
131       <paper-item>Item 2</paper-item>
132     </paper-listbox>
133   </paper-material>
134 </div>
135 <div horizontal layout>
136   <paper-material elevation="1">
137     <paper-menu selected="0">
138       <paper-item>Item 1</paper-item>
139       <paper-item>Item 2</paper-item>
140     </paper-menu>
141   </paper-material>
142   <paper-material elevation="1">
143     <paper-menu multi>
144       <paper-item>Item 1</paper-item>
145       <paper-item>Item 2</paper-item>
146     </paper-menu>
147   </paper-material>
148   <paper-menu-button>
149     <paper-icon-button icon="menu" class="dropdown-trigger"></paper-icon-button>
150     <paper-menu class="dropdown-content">
151       <paper-item>Share</paper-item>
152       <paper-item>Settings</paper-item>
153       <paper-item>Help</paper-item>
154     </paper-menu>
155   </paper-menu-button>
```

```

155 </div>
156
157 <h1>Input</h1>
158 <div vertical layout>
159   <paper-input label="text input"></paper-input>
160   <paper-textarea label="autoresizing textarea input"></paper-textarea>
161   <paper-input label="disabled input" disabled></paper-input>
162   <paper-input label="input with at most 10 characters" char-counter maxlength="10"
163     ></paper-input>
164   <paper-input label="this label never floats" no-label-float></paper-input>
165   <paper-input label="this label is always floating" always-float-label_
166     ↪placeholder="placeholder text"></paper-input>
167   <paper-input label="total" type="number">
168     <div prefix>$</div>
169   </paper-input>
170   <paper-slider min="10" max="200" value="110"></paper-slider>
171   <paper-slider min="10" max="200" value="110" pin max-markers="100"></paper-
172     slider>
173 </div>
174 <div horizontal layout>
175   <paper-radio-button>Radio</paper-radio-button>
176   <paper-radio-button checked>Radio</paper-radio-button>
177   <paper-radio-button disabled>Disabled</paper-radio-button>
178   <paper-radio-button noink>No ink</paper-radio-button>
179   <paper-radio-button primary>Primary</paper-radio-button>
180   <paper-radio-button secondary>Secondary</paper-radio-button>
181   <paper-radio-button info>Info</paper-radio-button>
182   <paper-radio-button warning>Warning</paper-radio-button>
183   <paper-radio-button error>Error</paper-radio-button>
184 </div>
185 <paper-radio-group selected="small" vertical layout>
186   <paper-radio-button name="small">Small</paper-radio-button>
187   <paper-radio-button name="medium">Medium</paper-radio-button>
188   <paper-radio-button name="large">Large</paper-radio-button>
189 </paper-radio-group>
190 <div vertical layout>
191   <paper-toggle-button>Toggle</paper-toggle-button>
192   <paper-toggle-button checked>Toggle</paper-toggle-button>
193   <paper-toggle-button disabled>Disabled</paper-toggle-button>
194   <paper-toggle-button noink>No ink</paper-toggle-button>
195 </div>
196
197 <h1>Progress</h1>
198 <paper-progress value="10"></paper-progress>
199 <paper-progress value="10" secondary-progress="30"></paper-progress>
200 <paper-spinner active></paper-spinner>
201
202 <h1>Tabs</h1>
203 <paper-tabs selected="0">
204   <paper-tab>TAB 1</paper-tab>
205   <paper-tab>TAB 2</paper-tab>
206   <paper-tab>TAB 3</paper-tab>
207 </paper-tabs>
208 <paper-tabs selected="0" scrollable>
209   <paper-tab>NUMBER ONE ITEM</paper-tab>
   <paper-tab>ITEM TWO</paper-tab>
   <paper-tab>THE THIRD ITEM</paper-tab>
   <paper-tab>THE ITEM FOUR</paper-tab>

```

```

210   <paper-tab>FIFTH</paper-tab>
211   <paper-tab>THE SIXTH TAB</paper-tab>
212   <paper-tab>NUMBER SEVEN</paper-tab>
213   <paper-tab>EIGHT</paper-tab>
214   <paper-tab>NUMBER NINE</paper-tab>
215   <paper-tab>THE TENTH</paper-tab>
216   <paper-tab>THE ITEM ELEVEN</paper-tab>
217   <paper-tab>TWELFTH ITEM</paper-tab>
218 </paper-tabs>
219 <paper-tabs selected="0" align-bottom autoselect>
220   <paper-tab>ITEM ONE</paper-tab>
221   <paper-tab>ITEM TWO</paper-tab>
222   <paper-tab>ITEM THREE</paper-tab>
223 </paper-tabs>
224
225 <h1>Tooltips</h1>
226 <paper-icon-button id="id_1" icon="favorite" alt="heart"></paper-icon-button>
227 <paper-tooltip for="id_1" offset="0">#10084;#65038;</paper-tooltip>
228
229 <h1>Ripples</h1>
230 <paper-material elevation="1" style="width:200px;height:200px;">
231   <paper-ripple></paper-ripple>
232 </paper-material>
233 <paper-material elevation="1" style="width:400px;height:400px;">
234   <paper-ripple></paper-ripple>
235 </paper-material>
236
237
238 </template>
239 <script src="javascript/theme-demo-min.js?__inline=true"></script>
240 </dom-module>

```

Now edit src/html/elements/theme-demo/css/theme-demo.scss

```

1  :host {
2    display: block;
3    padding: 1em;
4  }
5  [horizontal][layout] {
6    margin-bottom: 1em;
7  }
8
9  paper-fab {
10   margin-right: 1em;
11 }
12
13 paper-radio-button {
14   margin-right: 1em;
15 }
16
17 paper-checkbox[pink] {
18   --paper-checkbox-unchecked-color: var(--paper-pink-500);
19   --paper-checkbox-unchecked-ink-color: var(--paper-pink-500);
20   --paper-checkbox-checked-color: var(--paper-pink-500);
21   --paper-checkbox-checked-ink-color: var(--paper-pink-500);
22   --paper-checkbox-label-color: var(--paper-pink-500);
23 }
24

```

```

25 paper-material {
26   margin-right: 1em;
27 }
28
29 paper-progress {
30   margin-bottom: 1em;
31 }
32
33 paper-tabs {
34   margin-bottom: 1em;
35 }

```

Now edit src/html/elements/theme-demo/javascript/theme-demo.js

```

1 Polymer({
2   is: 'theme-demo'
3 });

```

Now execute Grunt to build our new Polymer element.

```

1 cd src/html
2 grunt

```

Make the theme-demo panel

Make a new c++ file /src/common/panels/ThemeDemo.cc

```

1 #include "subsystem/supervisor/panels/ThemeDemo.h"
2 #include "ajax/PolymerElement.h"
3
4 using namespace subsystempanels;
5 ThemeDemo::ThemeDemo( tsframework::CellAbstractContext* context, log4cplus::Logger& logger)
6 :tsframework::CellPanel(context, logger) {
7   logger_ = log4cplus::Logger::getInstance(logger.getName() + ".ThemeDemo");
8 }
9
10 void ThemeDemo::layout(cgicc::Cgicc& cgi) {
11   remove();
12   add(new ajax::PolymerElement("theme-demo"));
13 }

```

Make the include/subsystem/supervisor/panels/ThemeDemo.h file.

```

1 #ifndef _subsystem_supervisor_panels_ThemeDemo_h_
2 #define _subsystem_supervisor_panels_ThemeDemo_h_
3
4 #include "ts/framework/CellPanel.h"
5 #include "log4cplus/logger.h"
6 #include "cgicc/Cgicc.h"
7
8 namespace subsystempanels {
9   class ThemeDemo: public tsframework::CellPanel {
10     public:
11       ThemeDemo(tsframework::CellAbstractContext* context, log4cplus::Logger& logger);
12       void layout(cgicc::Cgicc& cgi);
13 }

```

```
13    } ;  
14}  
15#endif
```

Register the new class in the Makefile.

```
1 Sources=\  
2   version.cc\  
3   Cell.cc\  
4   CellContext.cc\  
5   Configuration.cc\  
6   ...  
7   panels/ThemeDemo.cc \  
8   ...
```

Now register your new panel in the menu so users can access it.

Edit src/common/Cell.cc

```
1 #include "subsystem/supervisor/panels/ThemeDemo.h"  
2 ...  
3 void subsystemsupervisor::Cell::init() {  
4   ...  
5   tsframework::CellPanelFactory* panelF = getContext()->getPanelFactory();  
6   ...  
7   panelF->add<subsystempanels::ThemeDemo>("Theme demo");
```

Now you can compile your cell and you should see the ThemeDemo panel in the menu under the ‘control-panels’ section.

The screenshot shows a software interface with a green header bar. The header contains the text "MULTICELL SUPERVISOR > Control Panels > Theme demo" and the X-DAG logo.

Buttons

Three rows of buttons are displayed:

- Row 1: DEFAULT, DISABLED, NOINK, ICON, PRIMARY, SECONDARY, INFO, WARNING, ERROR
- Row 2: DEFAULT, DISABLED, NOINK, ICON, PRIMARY, SECONDARY, INFO, WARNING, ERROR
- Row 3: DEFAULT, DISABLED, NOINK, ICON, PRIMARY, SECONDARY, INFO, WARNING, ERROR

Below the buttons are six circular icons with arrows and a heart symbol.

Checkboxes

Text: These demonstrate the ability to customize the theme per-panel.

<input type="checkbox"/> Default unchecked	<input type="checkbox"/> Default unchecked
<input type="checkbox"/> Unchecked with theme override	<input type="checkbox"/> Unchecked with theme override
<input checked="" type="checkbox"/> Default checked	<input checked="" type="checkbox"/> Default checked
<input checked="" type="checkbox"/> Checked with theme override	<input checked="" type="checkbox"/> Checked with theme override

Menus

Dinosaurs dropdown menu showing "brontosaurus". Below it is a horizontal menu bar with items: Item 1 (highlighted), Item 1, Item 1, Item 2, Item 2, Item 2.

1.2.11 Demo 8: Showing notifications

Make the notifications-demo element

In your cell, run:

```

1 cd src/html/elements
2 ./new-element.js
3 name of the new element: notifications-demo
4 creating new element <notifications-demo>...
5 removing any .svn folders in notifications-demo
6 Finished

```

You now have a working *notifications-demo* element. We'll edit it soon.

Register the notifications-demo element

Edit src/html/elements/elements.html and add the following line

```
1 <link rel="import" href="notifications-demo/notifications-demo.html">
```

This will tell AjaXell to load our new element.

Edit the notifications-demo element

Edit src/html/elements/notifications-demo/notifications-demo.html

```
1 <link rel="import" href="/extern/bower_components/polymer/polymer.html">
2 <link rel="import" href="/ts/common-elements/reset-css/reset-css.html">
3 <link rel="import" href="/ts/common-elements/iron-flex-layout-attributes/iron-flex-
4 ↵layout-attributes.html">
5 <link rel="import" href="/extern/bower_components/paper-button/paper-button.html">
6 <link rel="import" href="/ts/ajaxell/html/elements/ag-toaster/throws-toast.html">
7
8 <dom-module id="notifications-demo">
9   <template>
10    <style include="reset-css"></style>
11    <style include="iron-flex-layout-attributes"></style>
12
13    <link rel="stylesheet" type="text/css" href="css/notifications-demo-min.css?_
14 ↵inline=true">
15
16    <div horizontal layout>
17      <paper-button flex info raised on-click="showInfo">show info</paper-button>
18      <paper-button flex info raised on-click="showBigInfo">show info with options</
19 ↵paper-button>
20      <paper-button flex info raised on-click="showModalInfo">show modal info</paper-
21 ↵button>
22      <paper-button flex info raised on-click="showModalInfox5">show 5 big infos</
23 ↵paper-button>
24    </div>
25
26    <div horizontal layout>
27      <paper-button flex warning raised on-click="showWarning">show warning</paper-
28 ↵button>
29      <paper-button flex warning raised on-click="showBigWarning">show warning with_
30 ↵options</paper-button>
31      <paper-button flex warning raised on-click="showModalWarning">show modal warning</
32 ↵paper-button>
33      <paper-button flex warning raised on-click="showModalWarningx5">show 5 big_
34 ↵warnings</paper-button>
35    </div>
36
37    <div horizontal layout>
38      <paper-button flex error raised on-click="showError">show error</paper-button>
39      <paper-button flex error raised on-click="showBigError">show error with options</
40 ↵paper-button>
41      <paper-button flex error raised on-click="showModalError">show modal error</
42 ↵paper-button>
43      <paper-button flex error raised on-click="showModalErrorx5">show 5 big errors</
44 ↵paper-button>
45    </div>
46
47    <paper-button raised on-click="showLog">show log in console</paper-button>
48
49  <p>
```

```

38     Notifications have 3 levels: info, warning, and error.
39     If a notification is triggered while another is still visible, different things ↴
40     can happen:
41         </p>
42         <ul>
43             <li>
44                 <p>
45                     The current notification is a modal window.
46                 </p>
47                 <p>
48                     The new notification will wait for the modal to finish, regardless of the ↴
49                     level.
50                 </p>
51             </li>
52             <li>
53                 <p>
54                     The current notification has a lower level than the new notification.
55                 </p>
56                 <p>
57                     The current notification will be discarded immediately and the new ↴
58                     notification will be shown.
59                 </p>
60             </li>
61             <li>
62                 <p>
63                     The current notification has an equal or higher level than the new ↴
64                     notification.
65                 </p>
66                 <p>
67                     The new notification will be queued until the previous one has finished.
68                     This queue can grow arbitrarily long.
69                 </p>
70             </li>
71         </ul>

72     </template>
73     <script src="javascript/notifications-demo-min.js?__inline=true"></script>
74 </dom-module>
```

Now edit src/html/elements/notifications-demo/css/notifications-demo.scss

```

1  :host {
2      display: block;
3  }
4  [horizontal][layout] {
5      margin-bottom: 10px;
6  }
```

Now edit src/html/elements/notifications-demo/javascript/notifications-demo.js

```

1 Polymer({
2     is: 'notifications-demo',
3
4     behaviors: [throwsToast],
5
6     /**
7      * This produces a toast (https://www.google.com/design/spec/components/snackbars-toasts.html)
8  }
```

```
8  */
9  showInfo: function showInfo() {
10    this.throwToast({
11      'type': 'info',
12      'message': 'this is some info at t=' + new Date().getTime(),
13      'callback': function callback(response) {
14        console.log("callback successfull: ", response);
15      }
16    });
17  },
18
19 /**
20 * This produces a toast, but the theme file makes it have an orange background.
21 * Also, a warning message has a higher priority than an info message.
22 * If an info toast is currently visible and a warning toast is thrown, the info
23 toast will be discarded immediately.
24 * if a warning toast is currently visible and an info toast is thrown, the info
25 toast will be displayed after the warning toast has closed.
26 * if a warning toast is currently visible and a warning toast is thrown, the
27 warning toast will be displayed after the warning toast has closed.
28 */
29 showWarning: function showWarning() {
30   this.throwToast({
31     'type': 'warning',
32     'message': 'this is a warning at t=' + new Date().getTime(),
33     'callback': function callback(response) {
34       console.log("callback successfull: ", response);
35     }
36   });
37 },
38
39 /**
40 * This produces a toast, but the theme file makes it have an red background.
41 * Also, an error message has a higher priority than an info or warning message.
42 */
43 showError: function showError() {
44   this.throwToast({
45     'type': 'error',
46     'message': 'this is an error at t=' + new Date().getTime(),
47     'callback': function callback(response) {
48       console.log("callback successfull: ", response);
49     }
50   });
51 },
52
53 /**
54 * This produces a toast. The user can interact with it and click one of two
55 buttons.
56 * This is still a toast and timeouts after a few seconds and doesn't force the
57 user
58 * to click one of the buttons, the response variable of the callback will be
59 null or
60 * the option (type string) the user selected.
61 */
62 showBigInfo: function showBigInfo() {
63   this.throwToast({
64     'type': 'info',
65     'message': 'this is big info, you can choose something',
```

```

60         'options': ['save the world', 'make dinosaurs'],
61         'callback': function callback(response) {
62             console.log("you have chosen to", response);
63         }
64     });
65 },
66
67 showBigWarning: function showBigWarning() {
68     this.throwToast({
69         'type': 'warning',
70         'message': 'this is a big warning, you will need to choose something',
71         'options': ['save the world', 'make dinosaurs'],
72         'callback': function callback(response) {
73             console.log("you have chosen to", response);
74         }
75     });
76 },
77
78 showBigError: function showBigError() {
79     this.throwToast({
80         'type': 'error',
81         'message': 'this is a big error, you will need to choose something',
82         'options': ['save the world', 'make dinosaurs'],
83         'callback': function callback(response) {
84             console.log("you have chosen to", response);
85         }
86     });
87 },
88
89 /**
90 * This produces a modal window. The user must choose one of the provided
91 * options and will not be able to do anything else.
92 */
93 showModalInfo: function showModalInfo() {
94     this.throwToast({
95         'type': 'info',
96         'message': 'this is big info, you will need to choose something',
97         'options': ['save the world', 'make dinosaurs'],
98         'blocking': true,
99         'callback': function callback(response) {
100            console.log("you have chosen to", response);
101        }
102    });
103 },
104
105 showModalWarning: function showModalWarning() {
106     this.throwToast({
107         'type': 'warning',
108         'message': 'this is a big warning, you will need to choose something',
109         'options': ['save the world', 'make dinosaurs'],
110         'blocking': true,
111         'callback': function callback(response) {
112             console.log("you have chosen to", response);
113         }
114     });
115 },
116
117 showModalError: function showModalError() {

```

```
118     this.throwToast({
119         'type': 'error',
120         'message': 'this is a big error, you will need to choose something',
121         'options': ['save the world', 'make dinosaurs'],
122         'blocking': true,
123         'callback': function callback(response) {
124             console.log("you have chosen to", response);
125         }
126     });
127 },
128
129 showModalInfox5: function showModalInfox5() {
130     for (var i = 1; i <= 5; i++) {
131         this.throwToast({
132             'type': 'info',
133             'message': 'this is big info ' + i + ', you will need to choose',
134             'options': ['save the world', 'make dinosaurs'],
135             'blocking': true,
136             'callback': function callback(response) {
137                 console.log("you have chosen to", response);
138             }
139         });
140     }
141 },
142
143 showModalWarningx5: function showModalWarningx5() {
144     for (var i = 1; i <= 5; i++) {
145         this.throwToast({
146             'type': 'warning',
147             'message': 'this is big warning ' + i + ', you will need to choose',
148             'options': ['save the world', 'make dinosaurs'],
149             'blocking': true,
150             'callback': function callback(response) {
151                 console.log("you have chosen to", response);
152             }
153         });
154     }
155 },
156
157 showModalWarningx5: function showModalWarningx5() {
158     for (var i = 1; i <= 5; i++) {
159         this.throwToast({
160             'type': 'warning',
161             'message': 'this is big error ' + i + ', you will need to choose',
162             'options': ['save the world', 'make dinosaurs'],
163             'blocking': true,
164             'callback': function callback(response) {
165                 console.log("you have chosen to", response);
166             }
167         });
168     }
169 },
170
171 /**
172 * Ajaxell keeps logs of every thrown toast. This function dumps this log
```

```

173     * into the console window of the browser.
174     */
175     showLog: function showLog() {
176         toaster.showLog();
177     }
178 }) ;

```

Now execute Grunt to build our new Polymer element.

```

1 cd src/html
2 grunt

```

Make the notifications-demo panel

Make a new c++ file /src/common/panels/NotificationsDemo.cc

```

1 #include "subsystem/supervisor/panels/NotificationsDemo.h"
2 #include "ajax/PolymerElement.h"
3
4 using namespace subsystempanels;
5 NotificationsDemo::NotificationsDemo( tsframework::CellAbstractContext* context,
6                                     log4cplus::Logger& logger)
7 :tsframework::CellPanel(context, logger) {
8     logger_ = log4cplus::Logger::getInstance(logger.getName() +".NotificationsDemo");
9 }
10 void NotificationsDemo::layout(cgicc::Cgicc& cgi) {
11     remove();
12     add(new ajax::PolymerElement("notifications-demo"));
13 }

```

Make the include/subsystem/supervisor/panels/NotificationsDemo.h file.

```

1 #ifndef _subsystem_supervisor_panels_NotificationsDemo_h_
2 #define _subsystem_supervisor_panels_NotificationsDemo_h_
3
4 #include "ts/framework/CellPanel.h"
5 #include "log4cplus/logger.h"
6 #include "cgicc/Cgicc.h"
7
8 namespace subsystempanels {
9     class NotificationsDemo: public tsframework::CellPanel {
10     public:
11         NotificationsDemo(tsframework::CellAbstractContext* context, log4cplus::Logger&
12                           logger);
12         void layout(cgicc::Cgicc& cgi);
13     };
14 }
15 #endif

```

Register the new class in the Makefile.

```

1 Sources=\
2     version.cc\
3     Cell.cc\
4     CellContext.cc\

```

```
5 Configuration.cc \
6 ...
7 panels/NotificationsDemo.cc \
8 ...
```

Now register your new panel in the menu so users can access it.

Edit src/common/Cell.cc

```
1 #include "subsystem/supervisor/panels/NotificationsDemo.h"
2 ...
3 void subsystemsupervisor::Cell::init() {
4 ...
5     tsframework::CellPanelFactory* panelF = getContext()->getPanelFactory();
6 ...
7     panelF->add<subsystempanels::NotificationsDemo>("Notifications demo");
```

Now you can compile your cell and you should see the NotificationsDemo panel in the menu under the ‘control-panels’ section.

The screenshot shows a user interface for the 'Notifications demo' panel. At the top, there's a navigation bar with the path: MULTICELL SUPERVISOR > Control Panels > Notifications demo. To the right of the path is a user icon. Below the navigation bar is a grid of buttons. The buttons are arranged in four rows: Row 1: SHOW INFO (blue), SHOW INFO WITH OPTIONS (blue), SHOW MODAL INFO (blue), SHOW 5 BIG INFOS (blue). Row 2: SHOW WARNING (orange), SHOW WARNING WITH OPTIONS (orange), SHOW MODAL WARNING (orange), SHOW 5 BIG WARNINGS (orange). Row 3: SHOW ERROR (red), SHOW ERROR WITH OPTIONS (red), SHOW MODAL ERROR (red), SHOW 5 BIG ERRORS (red). Row 4: SHOW LOG IN CONSOLE (gray). Below the buttons is a text area containing the message: 'Notifications have 3 levels: info, warning, and error. If a notification is triggered while another is still visible, different things can happen:'. Underneath this text is a bulleted list of four items. At the bottom of the panel is a yellow status bar with the text: 'this is a warning at t=1464081654623'.

1.3 Advanced section

1.3.1 Writing tests for your elements

Writing tests is important. It will allow you to keep track of your elements as browsers develop. And allows you to detect problems and fix them before they become a big problem.

Some people believe it is a good practice to write your tests before writing actual code. This practice is called Test-Driven Development (TDD). Whether you want to endorse this practice is up to you. But since you made it this far, and are capable of writing complex interfaces, it might be something to consider.

The grunt build tool supports a special command *grunt test*. When executed, it will traverse all your elements and look for a /<element-name>/test/index.html file and run tests defined there.

This chapter will get you started writing tests for this system.

Defining tests in your element

Your elements are defined in /src/html/elements. Every element resides in a subfolder there.

To define tests for your element (in this example called *my-element*), create a file /src/html/elements/my-element/test/index.html with the following content:

/src/html/elements/my-element/test/index.html

```

1 <!doctype html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <script src="/bower_components/webcomponentsjs/webcomponents-lite.js"></script>
6     <script src="/bower_components/web-component-tester/browser.js"></script>
7   </head>
8   <body>
9     <script>
10    WCT.loadSuites([
11      'my-element.html',
12      'my-element.html?dom=shadow'
13    ]);
14    </script>
15  </body>
16</html>
```

This will instruct the testing system to execute tests in *my-element.html* in both shadow DOM mode (=Google Chrome) and shady DOM mode (=Firefox, Safari).

Now for the actual test file:

/src/html/elements/my-element/test/my-element.html

```

1 <!doctype html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <meta name="viewport" content="width=device-width, minimum-scale=1.0, initial-
6       scale=1.0, user-scalable=yes">
7
8     <script src="/bower_components/webcomponentsjs/webcomponents-lite.js"></script>
9     <script src="/bower_components/web-component-tester/browser.js"></script>
10
11    <link rel="import" href="../my-element.html">
12  </head>
13  <body>
14
15    <!-- You can use the document as a place to set up your fixtures. -->
16    <test-fixture id="my-element-fixture">
17      <template>
18        <my-element>
19          <h2>my-element</h2>
20        </my-element>
21      </template>
22    </test-fixture>
23
24    <script>
25      suite('<my-element>', function() {
26        var element;
```

```
27     element = fixture('my-element-fixture');
28   });
29
30   test('defines the "author" property', function() {
31     assert.equal(element.someObject.name, 'deinonychus');
32   });
33   test('distributed children', function() {
34     var els = element.getContentChildren();
35     assert.equal(els.length, 1, 'one distributed node');
36     assert.equal(els[0], element.querySelector('h2'), 'content_
→distributed correctly');
37   });
38 });
39 </script>
40
41 </body>
42 </html>
```

The first test in this file simply tests if an object called `someObject` defined in your element has a property `name` with value `deinonychus`. It uses the `equal()` function, the one you'll probably use most as well. There are many more functions available to you to write your tests. The library providing these functions is the Chai Assertion Library. Some examples:

```
1 expect(element).to.contain.all.keys('browserName', 'platform', 'url', 'version');
2 expect(element).to.not.have.keys('browsers', 'disabled');
3 expect(element.someObject).to.have.property('name', 'deinonychus');
4 expect(object1).to.deep.equal(object2);
5 expect(element.someArray).to.have.length(4);
6 element.someObject.name.should.be.a('string');
7 element.someObject.should.have.property('arr').with.length(3);
8 assert.typeOf(element.someObject.name, 'string');
```

Check <http://chaijs.com/> for a full list and documentation.

The second test is more advanced. It tests if your element contains a `<content></content>` tag. Notice the `h2` tag in the template, this code checks if it is actually inserted.

Interacting with your element

You can use the Polymer DOM API to execute JavaScript on elements inside your element (e.g. push buttons):

```
1 // shorthand function for selection by id
2 var mybutton = element.$buttonid;
3 // or full access to anything with querySelector
4 // this selects the first paper-button element inside your element
5 var mybutton = Polymer.dom(element.root).querySelector("paper-button");
6 mybutton.click();
```

Testing Events

Use `addEventListener` to respond to events. Do remember to trigger them.

```
1 test('fires lasers', function(done) {
2   element.addEventListener('seed-element-lasers', function(event) {
3     assert.equal(event.detail.sound, 'Pew pew!');
```

```

4     done();
5   });
6   element.fireLasers();
7 });

```

Testing AJAX

Web Component Tester includes Sinon, which enables you to mock XHR requests and create fake servers.

`/src/html/elements/my-element/test/my-element.html`

```

1 <!doctype html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <meta name="viewport" content="width=device-width, minimum-scale=1.0, initial-
6       scale=1.0, user-scalable=yes">
7
8     <script src="/bower_components/webcomponentsjs/webcomponents-lite.js"></script>
9     <script src="/bower_components/web-component-tester/browser.js"></script>
10
11   <link rel="import" href="../my-element.html">
12 </head>
13 <body>
14
15   <!-- You can use the document as a place to set up your fixtures. -->
16   <test-fixture id="my-element-fixture">
17     <template>
18       <my-element>
19         <h2>my-element</h2>
20       </my-element>
21     </template>
22   </test-fixture>
23
24   <script>
25     suite('<my-element>', function() {
26       var element;
27       var server = sinon.fakeServer.create();
28       var responseHeaders = {
29         json: { 'Content-Type': 'application/json' }
30       };
31       setup(function() {
32         element = fixture('my-element-fixture');
33         server.respondWith(
34           'GET',
35           '/\responds_to_get_with_json.*/',
36           [
37             200,
38             responseHeaders.json,
39             '{"success":true}'
40           ]
41         );
42       });
43       teardown(function() {
44         server.restore();
45       });
46
47       test('correctly handles the AJAX request', function() {

```

```

46     var request = element.functionThatTriggersAnAJAXRequest();
47     // catch the response and return fake data
48     server.respond();
49     expect(request.response).to.be.ok;
50     expect(request.response).to.be.an('object');
51     expect(request.response.success).to.be.equal(true);
52   });
53   test('has the correct xhr method', function() {
54     var request = ajax.generateRequest();
55     expect(request.xhr.method).to.be.equal('GET');
56   });
57 });
58 </script>
59
60 </body>
61 </html>

```

1.3.2 Theming your cell

The cell follows a theme file. Your elements also follow this by importing the ‘reset-css’ style element.

By default this theme file styles your buttons, dropdowns, etc.

The screenshot shows a 'Theme demo' panel with a green header bar. The sidebar on the left lists categories such as 'Commands', 'Operations', and 'Control Panels'. Under 'Control Panels', there are links to 'About', 'ButtonExample', 'Chart examples', 'Code editor example', 'Collapsible Demo', 'DialogExample', 'EmbeddingExample', 'Flexbox layout example', 'Form example', 'GDbPanel', 'HelloWorld', 'Inputfile', 'MeasurementExample', 'Notifications demo', 'Refresh example', 'RootExample', 'TabContainerExample', 'Table example', 'Theme demo', 'TreeNodeExample', and 'ts-ajax and data-binding'. The main content area displays several sections: 'Buttons' showing various button states (Default, Disabled, NoInk, Icon, Primary, Secondary, Info, Warning, Error) and icons; 'Checkboxes' showing checked and unchecked states with theme overrides; 'Menus' showing dropdowns and item lists; and 'Input' showing form fields. The interface uses a dark theme with green and blue highlights for primary and secondary colors.

You will also notice the existence of a *primary* and *secondary* color. By default these colors are green (#00671a) and blue (#448aff) respectively. You can change these things.

Overriding the theme in an element

You can use the CSS styles in your element to override the theme for that particular element.

This example will show how you can make pink checkboxes.

If your html code looks like this (notice the *pink* attribute):

```
1 <paper-checkbox>Default unchecked</paper-checkbox>
2 <paper-checkbox pink>Unchecked with theme override</paper-checkbox>
```

The theme override can be implemented like this in the CSS:

```
1 paper-checkbox[pink] {
2   --paper-checkbox-unchecked-color: var(--paper-pink-500);
3   --paper-checkbox-unchecked-ink-color: var(--paper-pink-500);
4   --paper-checkbox-checked-color: var(--paper-pink-500);
5   --paper-checkbox-checked-ink-color: var(--paper-pink-500);
6   --paper-checkbox-label-color: var(--paper-pink-500);
7 }
```

Checkboxes

These demonstrate the ability to customize the theme per-panel.

- | | |
|---|---|
| <input type="checkbox"/> Default unchecked | <input type="checkbox"/> Default unchecked |
| <input type="checkbox"/> Unchecked with theme override | <input type="checkbox"/> Unchecked with theme override |
| <input checked="" type="checkbox"/> Default checked | <input checked="" type="checkbox"/> Default checked |
| <input checked="" type="checkbox"/> Checked with theme override | <input checked="" type="checkbox"/> Checked with theme override |

Notice that you always have access to the material design (*paper*) colors. They always use the `--paper-<colorname>-<intensity>` naming convention. A full list can be found at <https://www.google.com/design/spec/style/color.html>

Overriding colors for the entire cell

In the C++ code of your cell you can specify the primary and secondary colors to use in your cell.

In the cell.cc file, add the following lines:

```
1 // makes primary color red
2 getContext() -> setPrimaryColor(ajax::RGB(255, 0, 0));
3 // makes secondary color blue
4 getContext() -> setSecondaryColor(ajax::RGB(0, 0, 255));
```

The result will look like this:

The screenshot shows a navigation bar at the top with 'MULTICELL SUPERVISOR > Control Panels > Theme demo'. On the left, there's a sidebar with sections for Commands (Candela, Default, Monitor, Supervisor, TStore, XDAQ), Operations (Peers, DB, MON, WORKER 1, WORKER 2, XDAQ), and Control Panels (Monitoring). The main content area is titled 'Buttons' and shows examples of primary, secondary, info, warning, and error buttons. Below that is a section for 'Checkboxes' with a legend for checked and unchecked states with theme overrides. The 'Menus' section shows dropdowns with items like 'Item 1' and 'Item 2'. Finally, there's an 'Input' section.

Overriding the theme across multiple elements

You can also create your own theme that complements the default theme. A theme file is also a Polymer element, but it lacks JavaScript code.

When creating your own theme file, the HTML should look like this.

```

1 <link rel="import" href="/extern/bower_components/polymer/polymer.html">
2
3 <dom-module id="my-theme">
4   <template>
5     <link rel="stylesheet" type="text/css" href="css/my-theme-min.css?__inline=true">
6   </template>
7 </dom-module>
```

```

1 paper-checkbox[pink] {
2   --paper-checkbox-unchecked-color: var(--paper-pink-500);
3   --paper-checkbox-unchecked-ink-color: var(--paper-pink-500);
4   --paper-checkbox-checked-color: var(--paper-pink-500);
5   --paper-checkbox-checked-ink-color: var(--paper-pink-500);
6   --paper-checkbox-label-color: var(--paper-pink-500);
7 }
```

The `reset-css` element can be found at <https://svnweb.cern.ch/trac/cactus/browser/trunk/cactuscore/ts/common-elements/src/reset-css>, the scss file can be consulted to see the full capabilities of theme files.

Available colors

You always have access to the material design (*paper*) colors. They always use the `--paper-<colortype>-<intensity>` naming convention. A full list can be found at <https://www.google.com/design/spec/style/color.html>

The framework also defines the following colors:

- `-dark-primary-color: #e50000;`
- `-default-primary-color: #ff0000;`
- `-primary-color: #ff0000;`
- `-light-primary-color: #ff0000;`
- `-secondary-color: #0000ff;`
- `-primary-color-light-10: #ff0000;`
- `-primary-color-dark-10: #e50000;`
- `-secondary-color-light-10: #0000ff;`
- `-secondary-color-dark-10: #0000e5;`
- `-primary-color-light-20: #ff0000;`
- `-primary-color-dark-20: #cc0000;`
- `-secondary-color-light-20: #0000ff;`
- `-secondary-color-dark-20: #0000cc;`
- `-primary-color-light-30: #ff0000;`
- `-primary-color-dark-30: #b20000;`
- `-secondary-color-light-30: #0000ff;`
- `-secondary-color-dark-30: #0000b2;`
- `-primary-color-light-40: #ff0000;`
- `-primary-color-dark-40: #990000;`
- `-secondary-color-light-40: #0000ff;`
- `-secondary-color-dark-40: #000099;`
- `-primary-color-light-50: #ff0000;`
- `-primary-color-dark-50: #7f0000;`
- `-secondary-color-light-50: #0000ff;`
- `-secondary-color-dark-50: #00007f;`
- `-primary-color-light-60: #ff0000;`
- `-primary-color-dark-60: #660000;`
- `-secondary-color-light-60: #0000ff;`
- `-secondary-color-dark-60: #000066;`
- `-text-primary-color: #ffffff;`
- `-accent-color: #ff4081;`
- `-primary-background-color: #ffffff;`
- `-primary-text-color: #212121;`
- `-secondary-text-color: #757575;`
- `-disabled-text-color: #bdbdbd;`

- --divider-color: #e0e0e0;
- --error-color: red;

variables and mixins

The colors described earlier are variables. You can use them in your CSS code like this:

```
1 :host {  
2   // declare variables  
3   --my-own-color: green;  
4   --my-own-padding: 5px;  
5 }  
6 P {  
7   // use variables  
8   color: var(--my-own-color);  
9   padding: var(--my-own-padding);  
10  // a second argument is a value to use if the variable isn't defined  
11  background-color: var(--primary-color-light-50, 'blue');  
12 }
```

Mixins are a way to not just store one variable, but an entire block of CSS code.

```
1 :host {  
2   // declare mixin  
3   --my-mixin-name: #{'  
4     background-color: green;  
5     border-radius: 4px;  
6     border: 1px solid gray;  
7   '};  
8 }  
9 P {  
10  // use mixin  
11  @apply(--my-mixin-name);  
12 }
```

This can be used for sharing style across elements. You can also put an @apply(); or var(); rule in your CSS without declaring the variable or mixin. This will allow other developers to influence the style of your element in a controlled manner without needing to change the element's source code.

1.4 Available resources

1.4.1 The bower-components package

The bower-components package contains any front-end package that is pulled from the internet (i.e. not made by us).

It currently contains:

- The Polymer library
- iron-elements
- paper-elements
- gold-elements
- neon-elements

- platinum-elements
- juicy-jsoneditor
- juicy-ace-editor
- vaadin-grid
- paper-datable
- jQuery 2.2.2
- moment.js
- page.js
- cytoscape.js
- file-saver.js
- saveSvgAsPng
- KaTeX

iron-elements

The iron-elements are a set of web components aiming to provide a basic set of tools and enhancements to standard elements, for example to provide them with data-binding capabilities.

These elements do not make assumptions about the used layout or styling, and are expected to maintain a spartan view, if they render a view at all.

Iron-elements aim to extend basic html elements (e.g. <iron-input> to extend <input>), provide façade elements for javascript functionality (e.g. <iron-ajax> to easily make AJAX requests), or provide new functionality that would be considered basic functionality (e.g. <iron-icon> to display an icon).

paper-elements

Paper-elements is a set of elements that focus on bringing Material Designcite{materialdesign} to web components.

Paper-elements aims to extend iron-elements with material design (e.g. <iron-input> becomes <paper-input>), and introduce new elements that are unique to material design (e.g. <paper-toast>)

gold-elements

Gold elements are input elements for specific use cases (e.g. email, phone numbers, credit card numbers, ldots).

They all extend the *paper-input* element and provide specific validation and formatting functionality.

platinum-elements

Platinum-elements are a set of Web Components focused on providing a façade for web-app capabilities like Service Workers, server push, and bluetooth connectivity.

neon-elements

neon-elements are a set of Web Components designed to be façades for the JavaScript animation API to make them available by purely writing HTML.

These elements do not use CSS Transitions, CSS Animations, or SVG, rather they use the new Web Animations API (url{<https://www.w3.org/TR/web-animations/>}).

These are among the most advanced Web Components in the packages available to panel developers. More info about their usage is provided here: url{<https://youtu.be/-tX0e29GQa4>}.

juicy-jsoneditor

juicy-json-editor is a web-based tool to view, edit, format, and validate JSON. It has various modes such as a tree editor, a code editor, and a plain text editor.

juicy-ace-editor

juicy-ace-editor is a web component that provides easy access to the Ace library. Ace is an embeddable code editor written in JavaScript. It matches the features and performance of native editors such as Sublime, Vim and TextMate. It can be easily embedded in any web page and JavaScript application. Ace is maintained as the primary editor for Cloud9 IDE and is the successor of the Mozilla Skywriter (Bespin) project.

vaadin-grid

Vaadin Grid is a fully featured datagrid for showing table data. It performs great even with huge data sets, fully supporting paging and lazy loading from any data source like a REST API. Grid allows you sort and filter data and customize how each cell gets rendered.

paper-datatatable

A material design implementation of a data table. Currently none of the panels use paper-datatatable, and use vaadin-grid instead. This because the development on this element seems dead.

moment.js

Moment.js is a JavaScript library that makes parsing, validating, manipulating, and displaying dates in JavaScript easy.

page.js

Tiny Express-inspired client-side router. It is used by AjaXell to manage the loading of panels.

cytoscape.js

A JavaScript library designed to paint network graphs.

file-saver.js

FileSaver.js implements the HTML5 W3C saveAs() FileSaver interface in browsers that do not natively support it. There is a FileSaver.js demo that demonstrates saving various media types.

FileSaver.js is the solution to saving files on the client-side, and is perfect for webapps that need to generate files, or for saving sensitive information that shouldn't be sent to an external server.

saveSvgAsPng

A JavaScript library that can save an SVG element to a PNG file.

KaTeX

KaTeX is a fast, easy-to-use JavaScript library for TeX math rendering on the web.

1.4.2 The common-elements package

The common-elements package contains web components that are made in-house and are useful across multiple projects (e.g. chart elements).

For more information visit [the common-elements page](#)

1.4.3 other packages

Every package can contain its own set of custom-made elements, yours can too.

AjaXell

AjaXell contains a set of elements to make its page-flow work, such as the *page-handler* and the *ts-session* element.

A notable element is *ag-toaster*, which allows anyone to show notifications to the user from their panels. For more information visit [the AjaXell page](#)

TS Framework

The TS Framework contains a set of panels that are always included in a cell (i.e. the about panel).

For more information visit [the TS Framework page](#)

SPHINX SYNTAX EXAMPLES

2.1 This is a Title

2.1.1 subtitle

subsubtitle

and so on

2.1.2 basic markup

italic text.

bold text.

a link <<http://xkcd.com/>>.

verbatim code

TODO: a reference

2.1.3 lists

- This is a bulleted list.
 - It has two items, the second item uses two lines. (note the indentation)
1. This is a numbered list.
 2. It has two items too.
 - sub 1
 - sub 2
3. This is another list
 4. Two lists separated only by whitespace are concatenated

2.1.4 terms

term (up to a line of text) Definition of the term, which must be indented

and can even consist of multiple paragraphs

next term Description.

2.1.5 line blocks

These lines are
broken exactly like in
the source file.

2.1.6 tables

Header row, column 1 (header rows optional)	Header 2	Header 3	Header 4
body row 1, column 1	column 2	column 3	column 4
body row 2	

A	B	A and B
False	False	False
True	False	False
False	True	False
True	True	True

2.1.7 links

This is a paragraph that contains a link.

Lorem ipsum ¹ dolor sit amet ... ²

Lorem ipsum [Ref] (page ??) dolor sit amet.

2.1.8 Source code

basic

This is a normal text paragraph. The next paragraph is a code sample:

```
It is not processed in any way, except  
that the indentation is removed.
```

```
It can span multiple lines.
```

This is a normal text paragraph again.

¹ Text of the first footnote.

² Text of the second footnote.

fancy

```

1 <h1>code block example</h1>
2 <auto-update data="{{my_variable}}" callback="cpp_callback" handle-as="text"></auto-
3   update>
<span>{{my_variable}}</span>
```

from external file

```

1 <!doctype html>
2 <html>
3   <head>
4     <title>command-input demo</title>
5
6     <meta name="viewport" content="width=device-width, minimum-scale=1.0, initial-
7       scale=1, user-scalable=yes">
8     <meta name="mobile-web-app-capable" content="yes">
9     <meta name="apple-mobile-web-app-capable" content="yes">
10
11    <script>window.Polymer = window.Polymer || {} ; window.Polymer.dom = 'shadow' ;</
12      <script>
13        <script src="/extern/bower_components/webcomponentsjs/webcomponents-lite.js"></
14          <script>
15            <link rel="import" href="../command-input.html">
16            <style>
17              command-input {
18                margin: 10px;
19              }
20            </style>
21          </head>
22          <body unresolved>
23            <template is="dom-bind">
24              <command-input name="dinosaurs are great" value="true" type="bool"></command-
25                input><br />
26                <command-input name="your_favorite_dinosaur" value="deinonychus" type="string">
27                  </command-input><br />
28                  <command-input name="positive number" value="0" type="unsigned int"></command-
29                    input>
30                    <command-input name="a number" value="0" type="int"></command-input>
31                    <command-input name="fraction" type="float"></command-input>
32                    <command-input name="fraction" value="0.1e-4" type="double"></command-input><br />
33                  </>
34            </template>
35          </body>
36        </html>
```

2.1.9 custom HTML

2.1.10 Fancy Math

Since Pythagoras, we know that $a^2 + b^2 = c^2$.

$$(a + b)^2 = a^2 + 2ab + b^2$$

$$(a - b)^2 = a^2 - 2ab + b^2$$

$$\begin{aligned}(a + b)^2 &= (a + b)(a + b) \\ &= a^2 + 2ab + b^2\end{aligned}$$

$$y = ax^2 + bx + c \tag{2.1}$$

$$f(x) = x^2 + 2xy + y^2 \tag{2.2}$$

Euler's identity, equation (2.3), was elected one of the most beautiful mathematical formulas.

$$e^{i\pi} + 1 = 0 \tag{2.3}$$