

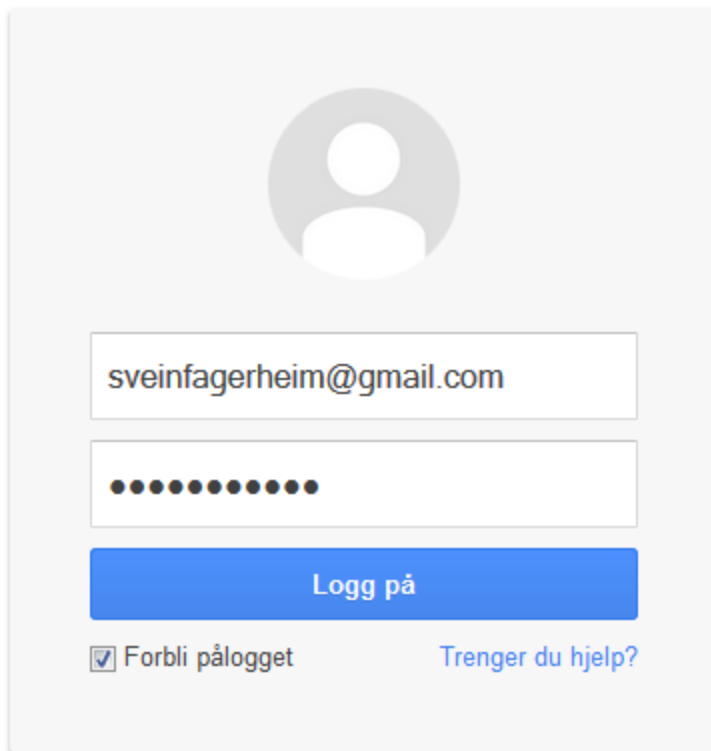
Vedlegg B: Produktdokumentasjon

1.1 Logge inn

Det første som møter brukeren når brukeren kommer til siden for første gang er en innloggingsmeny slik som den nedenfor.



Logg på med Google-kontoen din



The screenshot shows a login form with a grey background. At the top is a circular profile picture placeholder. Below it is a text input field containing the email address "sveinfagerheim@gmail.com". Underneath is a password input field with ten black dots. A blue button labeled "Logg på" is positioned below the password field. At the bottom left, there is a checked checkbox labeled "Forbli pålogget". To its right is a blue link labeled "Trenger du hjelp?".

[Opprett konto](#)

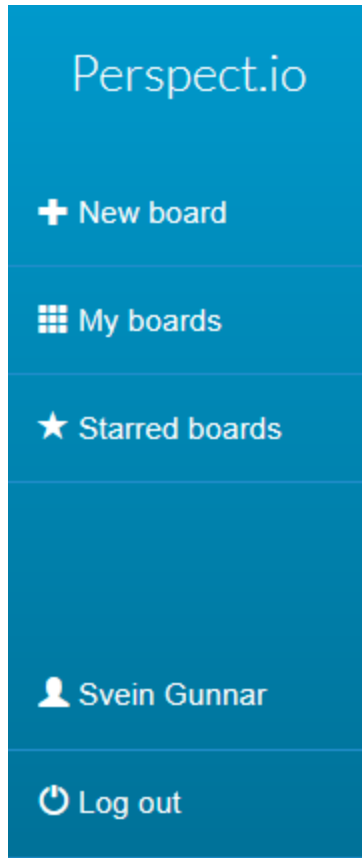
Her vil brukeren bli bedt om å logge inn med en Google konto. Hvis brukeren ikke har dette så må dette opprettes hos Google, ved å trykke på "Opprett konto".

1.2 Logge ut

Trykk først på dette ikonet er oppe i venstre hjørnet.



En ny meny vil da bli synlig på venstre side av skjermen, helt nederst på denne menyen vil det være et valg som heter “Log out”.



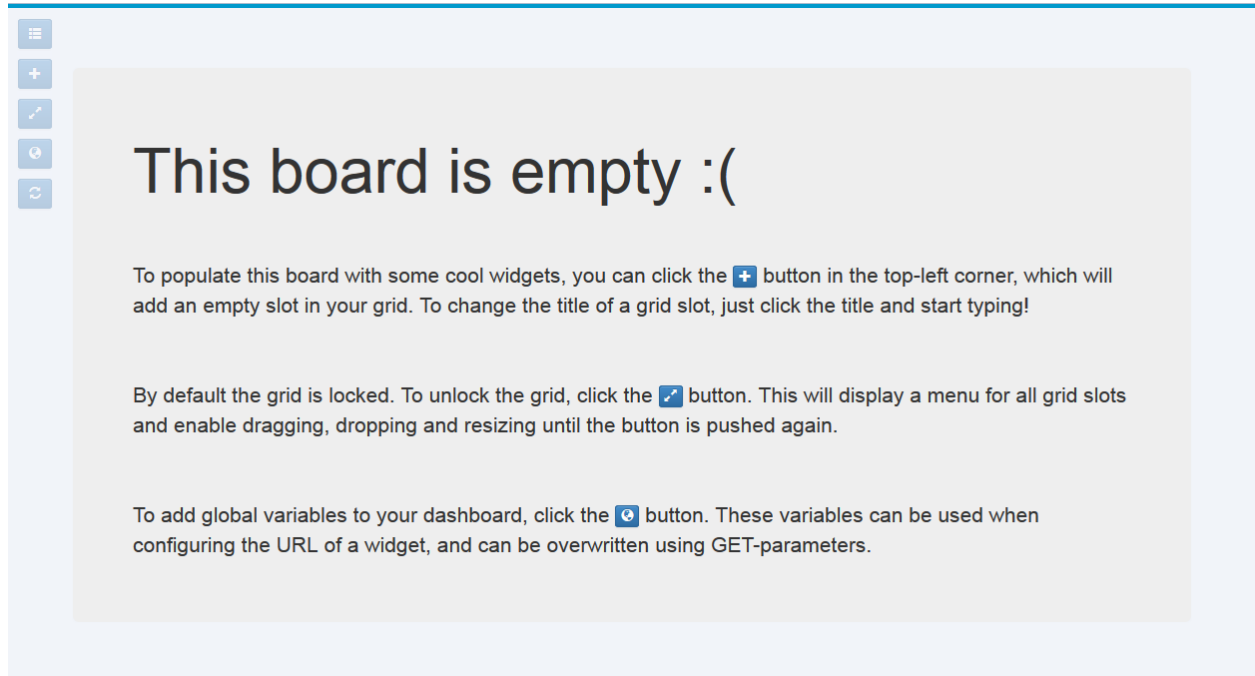
Når brukeren trykker på denne knappen så vil brukeren bli logget ut.

1.3 Legger til første dashboard

Først så åpner man menyen som man har gjort tidligere og deretter velger man alternativet som heter “New board” helt øverst i menyen.

All we need is the name of your new dashboard!

Her trenger man bare å skrive inn navnet man ønsker det nye dashbordet skal ha også trykke på ikonet “Create board” nederst i høyre hjørnet av bildet. Når dette er gjort så får man det bildet man ser under som er et tomt dashbord.

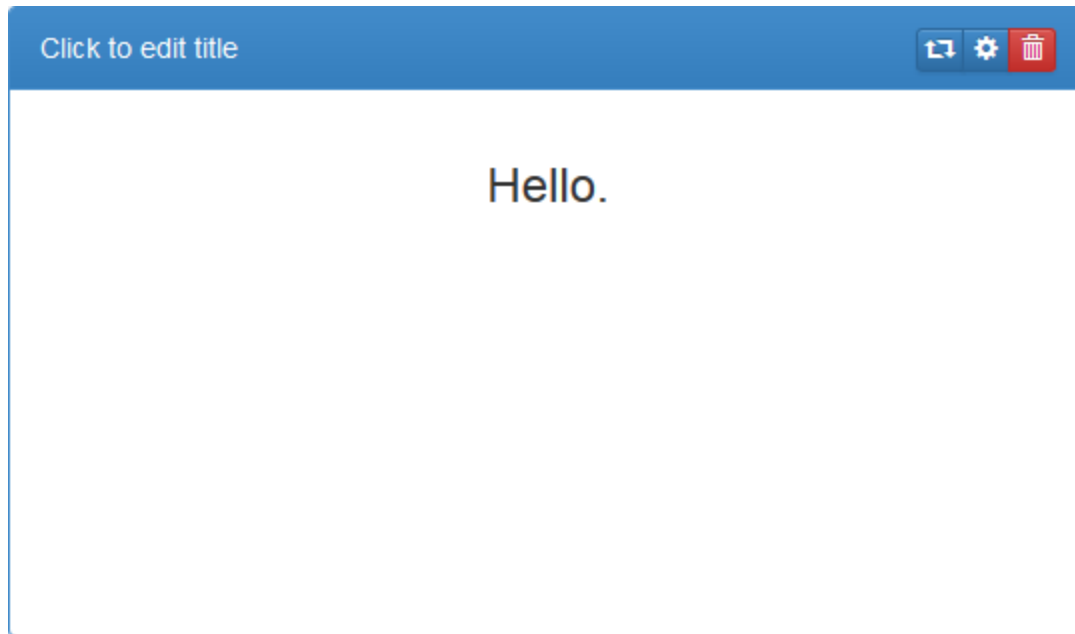


1.4 Legge til widget

Nå når man har laget et nytt dashbord så kan man legge til widgeter på dette dashbordet. Dette gjøres ved først å på ikonet som har en pluss i seg, vist nedenfor.

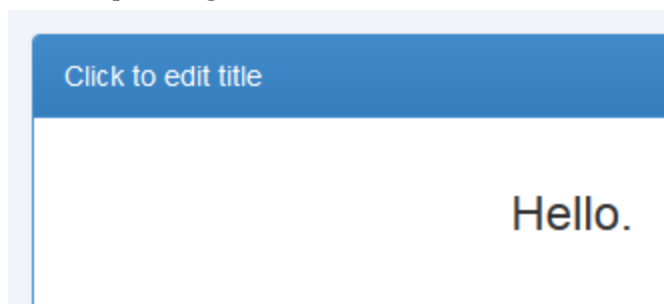


Når man har trykket på denne så vil det bli laget en tom widget som legger seg på dashbordet til brukeren. Denne widgeten er brukeren selv nødt til å fylle med innhold. Figuren under er av en tom widget.



1.5 Gir nytt navn til widget

For å gi nytt navn på widgeten man har laget så er det bare å trykke på teksten hvor det står "Click to edit title". Nå er det bare å viske ut den eksisterende teksten og skrive inn navnen på widgeten.

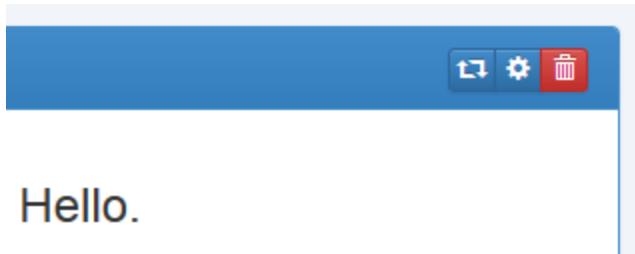


1.6 Legge til innhold i widgeten.

Først må man slå på muligheten til å endre innholdet i den tomme widgeten. Dette gjøres ved å trykke på dette ikonet.



Fargen på ikonet vil gå fra blått til gult og det er da mulig å endre innholdet. Det er nå tre nye ikoner oppe til høyre på widgeten. For å endre på innholdet så må man trykke på den midterste ikonet som har et tannhjul i seg.



Når man trykker på dette ikonet får man opp en ny meny slik som den vist på figuren under.

Widget-type: Bar chart

OpenTSDB URL: Insert URL here!

Buttons: Close, Resolve variables, Save widget

Her velger man hvilken widget type man ønsker også limer man inn en openTSDB url og trykker på "Save widget".

1.7 Flytter widget

Trykker på "Grid editing" ikonet og deretter holder musepekeren over widgeten og hvis musepekeren ser slik ut.



Så er det bare å trykke og holde inn venste museknapp også trekke widgeten dit brukeren ønsker å ha den. Når brukeren er ferdig med å plassere widgetene så må brukeren huske å skru av "Grid editing" ved å trykke på ikonet slik at posisjonene blir lagret.

1.8 Endrer størrelse på widget

Må bare trykke på "Grid editing" ikonet også plassere musepekeren nede i høyre hjørne av widgeten slik at musepekeren endrer seg til musepekeren på figuren under.



Nå er det bare å holde inn venstre museknapp og dra musepekeren i en retning for å gjøre størrelsen større eller mindre.

Når man er ferdig så er det bare å trykke på “Grid editing” ikonet igjen.

1.9 Fjerner widget

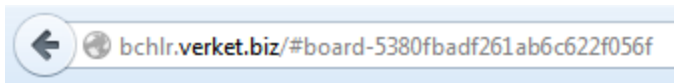
Når man skal slette en widget så må man trykke på “Grid editing” ikonet igjen og deretter trykke på “Delete widget” ikonet oppe i høyre hjørnet på widgeten, figuren under viser hvordan ikonet ser ut.



Når ikonet er trykket på så vil widgeten bli slettet og man er bare nødt til å trykke på “Grid editing” for å lagre endringene.

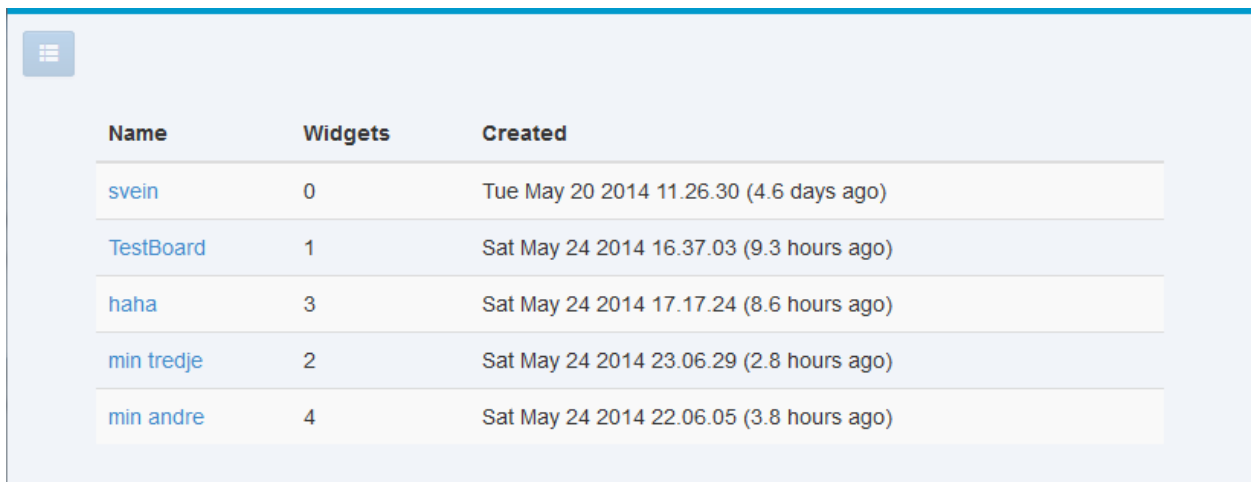
1.10 Deler dashboard

Hvis man er på dashboardet man ønsker å dele med andre så kan man gjøre dette ved å dele linken med andre.



1.11 Bytte dashboard

Når man skal bytte dashboard så åpner man menyen til venstre og trykker på “My boards” alternativet. Da vil man få opp alle dashboardene man har laget og det er bare å trykke på det dashboardet man ønsker å bruke. Nedenfor ser man hvordan listen med dashboard ser ut.



Name	Widgets	Created
svein	0	Tue May 20 2014 11.26.30 (4.6 days ago)
TestBoard	1	Sat May 24 2014 16.37.03 (9.3 hours ago)
haha	3	Sat May 24 2014 17.17.24 (8.6 hours ago)
min tredje	2	Sat May 24 2014 23.06.29 (2.8 hours ago)
min andre	4	Sat May 24 2014 22.06.05 (3.8 hours ago)

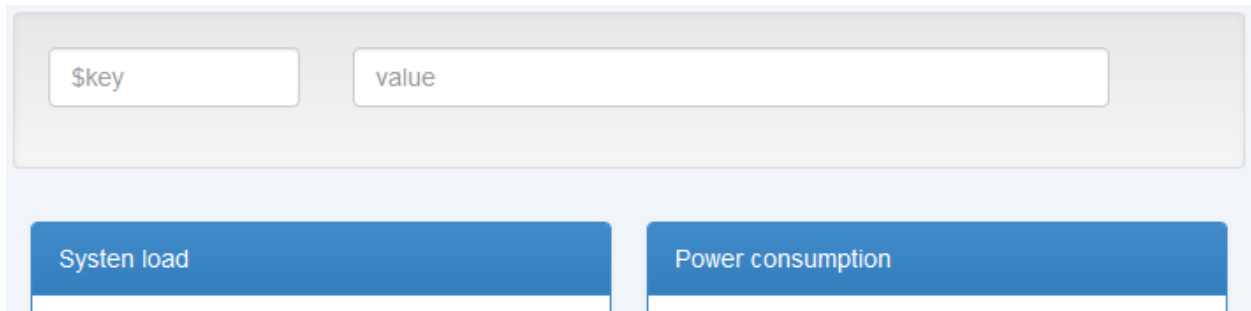
1.13 Endre globale variabler

Globale variabler går ut på at man kan endre den samme variabelen i alle widgetene ved å skrive navnet på variabelen og hva den skal endres til.

For å få tilgang til globale variabler så trykker man på ikonet som heter “Global variables”



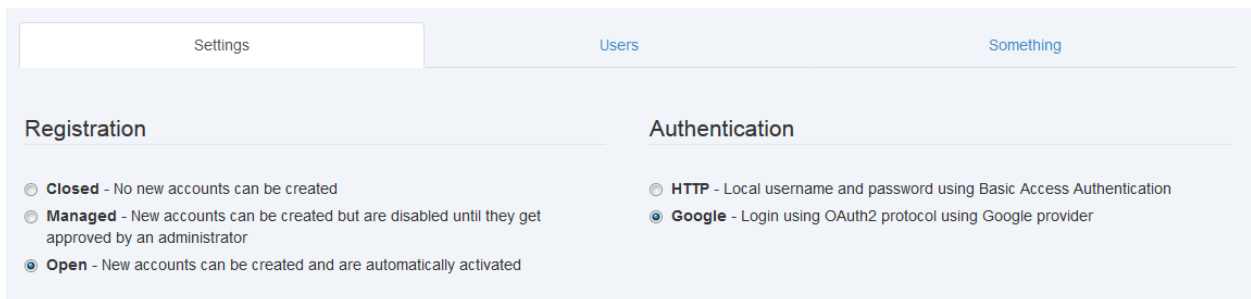
Man vil da få opp en meny som er over alle widget slik som den vist under.



Her setter man inn navnet på variabelen i venstre og hva variabelen skal være til høyre. Når man har gjort det så er det bare å trykke enter så har man lagt inn variabelen.

2.1 Endrer registreringspolicy

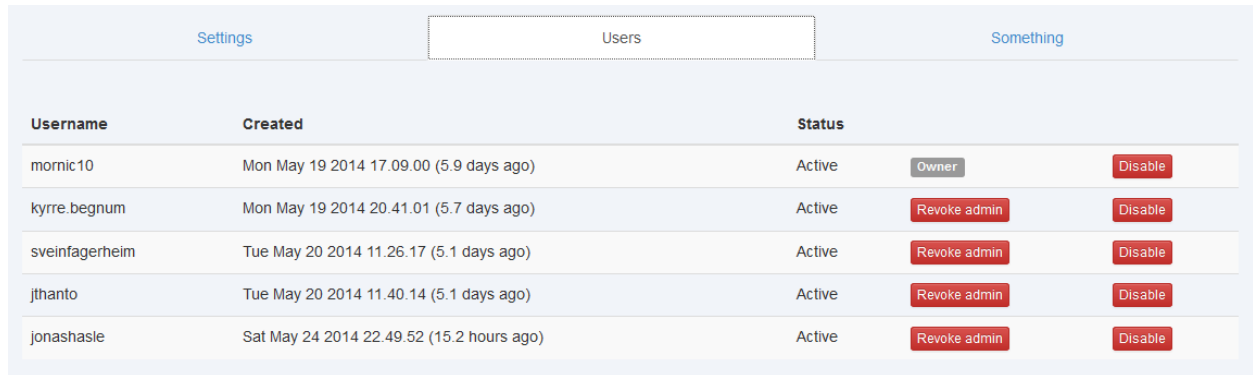
Man åpner menyen til venstre og velger alternativet som heter “Administration”. Da får man opp en meny som den vist nedenfor.



For å endre registreringspolicy så er det bare å velge en radioknapp under “Registration” til venstre i bildet. “Closed” betyr at ingen nye brukere kan registrere seg. “Open” betyr at alle kan registrere seg og begynne å bruke dashbordet med en gang. “Managed” betyr at man kan registrere seg, men man kan ikke gjøre noe før administratoren har aktivert den nye brukeren.

2.2 Brukerhåndtering

Man går på administrasjons siden og trykker på fanen som heter "Users". Da får man opp en liste lik den som er vist nedenfor.

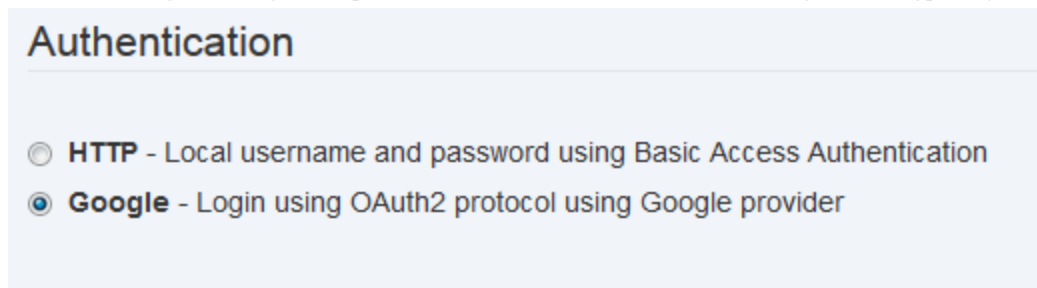


Username	Created	Status		
mornic10	Mon May 19 2014 17.09.00 (5.9 days ago)	Active	Owner	Disable
kyrre.begnum	Mon May 19 2014 20.41.01 (5.7 days ago)	Active	Revoke admin	Disable
sveinfagerheim	Tue May 20 2014 11.26.17 (5.1 days ago)	Active	Revoke admin	Disable
jthanto	Tue May 20 2014 11.40.14 (5.1 days ago)	Active	Revoke admin	Disable
jonashasle	Sat May 24 2014 22.49.52 (15.2 hours ago)	Active	Revoke admin	Disable

Her kan man aktivere og deaktivere brukere ved å trykke på knappen lengst til høyre på bildet. Man kan også gi og ta ifra andre brukere admin rettigheter ved å trykke på knappen som er til venstre for activate/disable knappen.

2.3 Endre Autentiseringmåte

Man kan velge om man skal bruke google sitt autentisering verkøy for brukerne eller en HTTP løsning. Dette gjøres på samme sted hvor man endrer registreringpolicy.



2.4 Installasjon

Se <<Installation.pdf>>

3.1 Dummymodul Autentisering

1. /*
2. * Perspect.io
3. *


```
4.  * Authentication route example.
5.  */
6.
7.  var passport = require('passport');
8.  var LocalStrategy = require('passport-local').Strategy;
9.  var User = require('../././model/user').UserModel;
10. var userCore = require('.././././core/user');
11.
12. /*
13. * Initialize Passport strategy
14. */
15. passport.use(new LocalStrategy(
16.   function(username, password, done) {
17.
18.     // Try to find existing user
19.     UserModel.findOne({username: username.valueOf()}, function(err, user) {
20.       if (err) return done(err);
21.
22.       // User exists
23.       if (user) {
24.
25.         // Match passwords
26.         return user.comparePassword(password.valueOf(), function(err, match) {
27.           if (err) return done(err);
28.
29.           // Authentication successful
30.           if (match) return done(null, user);
31.
32.           done('Username and password mismatch');
33.         });
34.       }
35.
36.       // Try to create new user
37.       var userObj = {
38.         username: username.valueOf(),
39.         password: password.valueOf()
40.       };
41.
42.       userCore.register(userObj, done);
43.     });
44.   }
45. ));
46.
```

```
47.  
48. /*  
49. * - Router -  
50. * Export module routes  
51. */  
52. module.exports = function(router) {  
53.  
54. /*  
55. * HTTP POST to /auth/local will authenticate the user  
56. */  
57. router.post('auth/local', passport.authenticate('local'));  
58.  
59. };
```

3.2 Dummymodul widgets

```
1. /*  
2. * Perspect.io  
3. *  
4. * Widget module example.  
5. */  
6.  
7. var Chart = Chart || {};  
8.  
9. (function($chart) {  
10. function MyChart(model, el, board) {  
11. // Widget model  
12. this.model = model;  
13.  
14. // DOM element  
15. this.el = el;  
16.  
17. // Board model  
18. this.board = board;  
19.  
20. // Extra options  
21. var extra = model.get('metadata').extra || {};  
22.
```

```

23. // Default chart options
24. this.chartOpts = {
25.   interpolateNulls: true,
26.   hAxis: {
27.     textPosition: (extra.xaxis && (extra.xaxis == 'true')) ? '' : 'none'
28.   },
29.   vAxis: {
30.     textPosition: (extra.yaxis && (extra.yaxis == 'true')) ? '' : 'none'
31.   },
32.   legend: {
33.     position: extra.legend || 'none'
34.   }
35. };
36.
37. // Init Google LineChart
38. this.chart = new google.visualization.SomeChart(this.el[0]);
39. this.data = [];
40. }
41.
42.
43. /*
44.  * Poll OpenTSDB for new data
45.  */
46. MyChart.prototype.poll = function() {
47.   var self = this;
48.
49.   // Resolve global variables and get data
50.   self.model.stats(resolveGlobalVariables(self.board), function(err, data) {
51.     if (err)
52.       return toastr.error(err);
53.
54.     var labels = ['Time'];
55.     var dataEntries = [];
56.
57.     $.each(data, function(timestamp, entries) {
58.       // Convert epoch timestamp to Date object
59.       var newEntry = [new Date(parseInt(timestamp) * 1000)];
60.
61.       entries.forEach(function(entry) {

```

```
62. // Gather all labels in a separate array
63. if (labels.indexOf(entry.tags.host) === -1)
64.     labels.push(entry.tags.host);
65.
66. // Insert data into correct index for this label
67.     newEntry[labels.indexOf(entry.tags.host)] = parseFloat(entry.value);
68. });
69.
70.     dataEntries.push(newEntry);
71. });
72.
73. // Fill empty slots with null-values or they will be missing
74. for (var i in dataEntries) {
75.     for (var x = 0; x < labels.length; x++) {
76.         if (typeof dataEntries[i][x] === 'undefined')
77.             dataEntries[i][x] = null;
78.     }
79. }
80.
81. // Join labels and data
82. var ad = [labels].concat(dataEntries);
83. self.data = google.visualization.arrayToDataTable(ad);
84.
85. // Redraw
86. self.redraw();
87. });
88. };
89.
90.
91. /*
92.  * Redraw the chart
93.  */
94. MyChart.prototype.redraw = function() {
95.     this.chart.draw(this.data, this.chartOpts);
96. };
97.
98.
99. /*
100.  * Chart export
```

```
101.  */
102.  $chart.mychart = {
103.    name: 'My chart',
104.    class: MyChart,
105.    extraOpts: {
106.      legend: $('<select/>').attr({
107.        'data-legend': 'Legend',
108.        'class': 'form-control'
109.      })
110.      .append($('<option/>').val('none').text('Disabled'))
111.      .append($('<option/>').val('top').text('Top'))
112.      .append($('<option/>').val('right').text('Right'))
113.      .append($('<option/>').val('bottom').text('Bottom'))
114.      .append($('<option/>').val('left').text('Left')),
115.    xaxis: $('<input/>').attr({
116.      'type': 'checkbox',
117.      'class': 'checkbox',
118.      'data-legend': 'X-Axis'
119.    }),
120.    yaxis: $('<input/>').attr({
121.      'type': 'checkbox',
122.      'class': 'checkbox',
123.      'data-legend': 'Y-Axis'
124.    })
125.  }
126. };
127. })(Chart);
```