
Sensor Widgets Documentation

Versió 1.0

Oscar Fonts

des. 11, 2017

1	Resum de l'estàndard SOS	1
1.1	Conceptes	1
1.2	Peticions	2
1.3	Referències	5
2	Com utilitzar els Sensor Widgets	7
2.1	El Wizard	7
2.2	Emporteu-vos-el: Enllaça i incrusta	7
2.3	Ús des de Javascript	8
2.4	Personalització de l'aspecte gràfic	9
3	Els Widgets un a un	11
3.1	Rumb (compass)	11
3.2	Manòmetre (gauge)	11
3.3	Taula jQuery (jqgrid)	12
3.4	Mapa (map)	12
3.5	Panell (panel)	18
3.6	Barra (progressbar)	18
3.7	Status (status)	19
3.8	Taula (table)	19
3.9	Termòmetre (thermometer)	19
3.10	Sèrie temps (timechart)	20
3.11	Rosa vents (windrose)	20
4	Com contribuir al projecte	23

Resum de l'estàndard SOS

Els Sensor Widgets són una eina de visualització de dades per a serveis que compleixin amb l'estàndard *Sensor Observation Service (SOS)* de l'OGC.

Els widgets inclouen un client SOS que suporta la versió 2.0 de estàndard, i, en aquests moments, necessiten un endpoint en format JSON, que no és part de l'estàndard, sinó un format opcional que proporciona la implementació de *52 north* en la seva versió 4.0.0 o superior.

Nota: De fet, el client SOS podria estendre's per suportar la codificació obligatòria KVP/XML, de manera que els Sensor widgets serien compatibles amb altres implementacions de servidors SOS 2.0.

Un *Sensor Observation Service* ofereix dades procedents d'una col·lecció de sensors. Així és com està organitzat un servei SOS:

1.1 Conceptes

Avís: La següent és una visió simplificada dels principals conceptes de SOS, una referència ràpida per als nous usuaris, que probablement estiguin més interessats en visualitzar algunes dades que en la plena comprensió dels conceptes que s'amaguen als estàndards SWE, SOS i O&M de l'OGC. Si es pretén implementar un servei SOS, es recomana no prendre aquesta guia com a referència, i referir-se a les especificacions OGC oficials (veure referències al final d'aquest capítol).

1.1.1 Offering

Les dades servides per un servei SOS s'agrupen en diferents *Offerings*. Per exemple, un servei SOS "meteo" podria tenir els següents Offerings: imatges de satèl·lit, dades de radar, mesures d'estacions meteorològiques, mapes de predicció, etc. Cada offering exposa dades d'un sensor o una xarxa de sensors, descrits com un *procedure*.

Es poden interpretar els Offerings com a seccions o calaixos que classifiquen les diferents dades segons el seu origen o naturalesa.

1.1.2 Procedure

Una procedure descriu un sensor, un conjunt de sensors, or un procés, que en genera un conjunt d'observacions. Proporciona metadades sobre les entrades i sortides del sensor, dades de calibració i processat, informació de contacte, i la disponibilitat de dades (extensions espacial i temporal), etc.

Normalment ve descrit en el format [SensorML](#).

Es pot considerar una Procedure com un document de metadades sobre el(s) sensor(s) o procés(sos) a càrrec de generar les dades que ofereix el servei.

Un Offering està relacionat amb una sola Procedure, mentre que una Procedure pot ser usada per diferents Offerings. Per exemple, una Procedure podria ser una "Xarxa d'Estacions Meteorològiques", i aquesta mateixa xarxa d'estacions ser usada en diferents Offerings, per exemple per a diferents períodes de temps. L'Offering corresponent podria ser: "Mesures de la Xarxa d'Estacions Meteorològiques per a l'any 2015".

1.1.3 Feature of Interest

Cada observació d'un servei SOS està referida a una *Feature Of Interest* (FOI), que habitualment determina el lloc on el fenomen observat ha tingut lloc. Per exemple, per a imatges satèl·lit, la FOI podria ser el seu *footprint* (polígon que determina l'àrea fotografiada sobre la superfície de la terra), o per a una mesura de temperatura, la FOI podria ser la ubicació del termòmetre (puntual).

Les FOI poden veure's com el conjunt de llocs als quals estan referides les dades.

1.1.4 Observed Property

La propietat que es mesura, tal que: Temperatura, Direcció del vent, Nuvolositat, Nombre de vehicles... pot ser un valor numèric (una quantitat i una unitat de mesura), lògic (pren els valors verdader o fals), categòric (un valor d'entre una llista: assolat, ennuvolat, plujós), o descriptiu (un text).

1.1.5 Observation

Finalment, una *Observation* és el valor que pren una *Observed Property* en un moment (Phenomenon Time) i un lloc (Feature Of Interest) determinats. Per exemple: "La temperatura a Barcelona el 22/09/2015 a les 11:52 és de 23 graus centígrads".

1.2 Peticions

Totes les peticions SOS han de contenir els següents paràmetres mínims:

- Service: SOS.
- Version: 2.0.0 (és la versió suportada pels Sensor Widgets).
- Request: el nom de la petició, per exemple `GetCapabilities`.

A continuació presentem les peticions SOS 2.0 que es fan servir als Sensor Widgets:

1.2.1 GetCapabilities

Com que la resposta d'un GetCapabilities pot ser bastant llarga, la petició GetCapabilities permet especificar un paràmetre `sections` per recuperar només les parts del document que ens interessin.

En concret, la secció `contents` descriu el servei com una col·lecció d'Offerings. Cada Offering conté els següents detalls:

- El nom de l'Offering (per exemple "observacions deuminutals"),

- L'identificador de l'Offering,
- L'identificador del Procedure lligat a aquest offering,
- La col·lecció d'Observable Properties (els seus identificadors),
- L'extensió espacial de les observacions que conté (el rectangle contenidor -bbox- de totes les Features of Interest),
- L'extensió temporal de les observacions que conté (període de temps que delimita totes les Observations).

Exemple de petició GetCapabilities en format JSON:

```
POST http://sensors.fonts.cat/sos/json
Content-Type: application/json
Contingut:
{
  "service": "SOS",
  "version": "2.0.0",
  "request": "GetCapabilities",
  "sections": ["Contents"]
}
```

Aquest document de Capabilities (secció contents) és el punt de partida per descobrir com està estructurat determinat servei SOS, així com les dades que conté. El document conté molts identificadors dels diferents elements (procedures, properties) però no el seu detall, que s'hauran d'obtenir mitjançant altres peticions com DescribeSensor i GetFeatureOfInterest.

1.2.2 DescribeSensor

La petició DescribeSensor accepta com a paràmetre un identificador de procedure, i retorna un document SensorML que conté metadades sobre el() sensor(s) o procés(sos) encarregats de produir les observacions.

Els continguts més rellevants d'aquest document són:

- L'identificador de la Procedure, un nom curt, i un nom més llarg,
- Una col·lecció de paraules clau (útils per als serveis de cerca dels catàlegs de metadades),
- Informació de contacte,
- El període de temps de validesa (redundant amb la resposta de Capabilities),
- El BBOX observat (redundant amb la resposta de Capabilities),
- La col·lecció de Features of Interest (els seus identificadors - nova informació que no es troba en el GetCapabilities),
- La col·lecció de Offerings (els seus identificadors) que es basen en aquesta procedure,
- Una llista de sortides (Optputs): Una col·lecció de ObservableProperties i la seva descripció: IDs, noms, tipus i unitats de mesura.

Aquesta petició es fa servir per ampliar detalls que no s'ofereixen a través del GetCapabilities, especialment la descripció de les Observable Properties (els seus noms i unitats de mesura).

Exemple de petició DescribeSensor en format JSON:

```
POST http://sensors.fonts.cat/sos/json
Content-Type: application/json
Contingut:
{
  "service": "SOS",
  "version": "2.0.0",
  "request": "DescribeSensor",
  "procedure": "http://sensors.portdebarcelona.cat/def/weather/procedure",
}
```

```
"procedureDescriptionFormat": "http://www.opengis.net/sensorML/1.0.1"
}
```

1.2.3 GetFeatureOfInterest

L'operació GetFeatureOfInterest accepta una *procedure* com a paràmetre, i retorna totes les Features of Interest relacionades amb aquest *procedure*. De fet, les Features of Interest estan vinculades a cadascuna de les Observations, però aquesta operació ens retorna una mena d'inventari de tots els seus possibles valors.

És útil per obtenir els detalls de les diverses localitzacions, com els seus noms i geometries. Així que generalment s'utilitza aquesta operació per poder dibuixar un mapa o un selector de Features per nom.

Exemple de petició GetFeatureOfInterest en format JSON:

```
POST http://sensors.fonts.cat/sos/json
Content-Type: application/json
Contingut:
{
  "service": "SOS",
  "version": "2.0.0",
  "request": "GetFeatureOfInterest",
  "procedure": "http://sensors.portdebarcelona.cat/def/weather/procedure"
}
```

1.2.4 GetDataAvailability

La petició GetDataAvailability també accepta una *procedure*, i, opcionalment, una col·lecció de FeatureOfInterest i/o ObservedProperty com a paràmetres.

Retorna el rang temporal dins del qual hi ha dades per a cada combinació Procedure-Feature-Property. Així, donat un sensor determinat, sabem per a quines dates disposarem de dades.

Exemple de petició GetDataAvailability en format JSON:

```
POST http://sensors.fonts.cat/sos/json
Content-Type: application/json
Contingut:
{
  "service": "SOS",
  "version": "2.0.0",
  "request": "GetDataAvailability",
  "procedure": "http://sensors.portdebarcelona.cat/def/weather/procedure",
  "featureOfInterest": ["http://sensors.portdebarcelona.cat/def/weather/
↵features#02"],
  "observedProperty": ["http://sensors.portdebarcelona.cat/def/weather/
↵properties#31"]
}
```

1.2.5 GetObservation

I, finalment, les dades mesurades.

Una petició GetObservation accepta els següents paràmetres:

- Un offering,
- Una col·lecció de FeatureOfInterest,
- Una col·lecció de ObservedProperties,

- Filtres espacials i/o temporals.

El filtrat és especialment interessant, ja que permet restringir les cerques de dades a un període de temps o una àrea geogràfica concreta. Els Sensor Widgets existents fins a la data només usen el filtrat temporal per a obtenir, o bé l'última dada disponible ("latest"), o bé una sèrie temporal de dades d'un període concret (per exemple, les darreres 3 hores).

Exemple de petició GetObservation en format JSON:

```
POST http://sensors.fonts.cat/sos/json
Content-Type: application/json
Contingut:
{
  "service": "SOS",
  "version": "2.0.0",
  "request": "GetObservation",
  "offering": "http://sensors.portdebarcelona.cat/def/weather/offerings#10m",
  "featureOfInterest": ["http://sensors.portdebarcelona.cat/def/weather/
↩features#P3"],
  "observedProperty": ["http://sensors.portdebarcelona.cat/def/weather/
↩properties#31"],
  "temporalFilter": [{
    "equals": {
      "ref": "om: resultTime",
      "value": "latest"
    }
  }]
}
```

La resposta és una col·lecció d'observacions, on cada observació consta de:

- L'identificador de l'Offering del qual procedeix,
- L'identificador del Procedure que la va generar,
- La Feature of Interest a què va referida la observació (descripció completa, amb el seu ID, nom i geometria),
- L'identificador de la Property que s'ha observat (però no el seu nom),
- Phenomenon time (quan ha ocorregut el que s'ha mesurat) i result time (quan s'ha generat la dada),
- I, per fi, el resultat en si, que consta d'un **valor** i d'una unitat de mesura.

Així, la resposta completa és tediosament redundant, i pot contenir centenars o milers de repeticions successives d'alguns dels elements descriptius en el mateix document de resposta. Imaginem una sèrie temporal de 5000 observacions del mateix sensor. L'única cosa que canvia és el temps i el valor. La resta de continguts (IDs, Features, unitats de mesura, etc) es repetiran 5000 vegades sense cap necessitat. Això impacta severament en l'agilitat del servei SOS.

Algunes implementacions de SOS (en concret, 52n SOS v.4.0.0+) ofereixen algunes estratègies que extenen l'estàndard per tractar de paliar aquesta situació, com la ja esmentada codificació dels missatges en JSON, i una extensió anomenada MergeObservationsIntoDataArray que "compacta" totes les observacions que procedeixen del mateix procedure, feature of interest i observed property en un SweArrayObservation (sèrie temporal de dades del mateix sensor sevida com un array de valors).

Nota: Els Sensor Widgets no aprofiten encara l'extensió MergeObservationsIntoDataArray. És una possible millora futura.

1.3 Referències

Especificacions oficials de l'Open Geospatial Consortium:

- OGC® Sensor Web Enablement: Overview And High Level Architecture v. 3 (White Paper). Ref. OGC 07-165.
- OpenGIS® SWE Service Model Implementation Standard v. 2.0. Ref. OGC 09-001.
- OGC® SWE Common Data Model Encoding Standard v. 2.0.0. Ref. OGC 08-094r1.
- Sensor Observation Service v. 1.0. Ref. OGC 06-009r6.
- OGC® Sensor Observation Service Interface Standard v. 2.0. Ref. OGC 12-006.
- OpenGIS® Sensor Model Language (SensorML) Implementation Specification v. 1.0.0. Ref. OGC 07-000.
- OGC Abstract Specification - Geographic information - Observations and measurements v.2.0. Ref. OGC 10-004r3.
- Observations and Measurements - XML Implementation v.2.0. Ref. OGC 10-025r1.

Com utilitzar els Sensor Widgets

Cada widget té una col·lecció de paràmetres obligatoris i d'altres opcionals. Configurar un widget és bàsicament escollir els valors adequats d'aquests paràmetres per tal d'obtenir el resultat desitjat.

2.1 El Wizard

La manera més fàcil de configurar un widget és usant el Wizard, que ens assistirà en la selecció dels paràmetres basant-se en una llista de possibles valors. Per exemple, la majoria de widgets tenen com a paràmetres obligatoris un “offering”, un o més “features” i una o més “properties”. El wizard inspeccionarà el servei SOS i ens permetrà triar d'entre una llista d'Offerings, features i properties existents.

Altres paràmetres habituals són el temps de refresc (`refresh_interval`), per als widgets que mostren dades en viu i que cal actualitzar periòdicament, o el rang de temps (`time_start`, `time_end`), per als widgets que mostren una col·lecció de mesures al llarg del temps. En aquest últim cas, el wizard ens assistirà amb un selector de rang de temps restringit al període temporal en què hi ha dades disponibles.

Paràmetres opcionals típics són la nota a peu (“footnote”), que és un text que es mostrarà al costat del widget, i l'adreça URL a un full d'estils CSS propi (“`custom_css_url`”), un mecanisme que permet personalitzar l'aspecte gràfic dels widgets.

Els detalls sobre els paràmetres que accepta de cada widget i el seu ús estan descrits en el proper capítol.

Un cop ajustats els valors en el formulari del wizard, al clicar al botó “Crea Widget”, es visualitzarà el resultat en el requadre “Vista del Widget”, i s'oferiran tres maneres de utilitzar-lo en el requadre “Emporteu-vos-el”: Com una pàgina HTML (“Enllaça”), com un component a incloure en una altra pàgina (“Incrusta”) i com un bloc de codi per utilitzar el widget dins d'un aplicació Javascript més gran (“Codi”).

2.2 Emporteu-vos-el: Enllaça i incrusta

Si fem clic al contingut d'“Enllaça”, obtindrem un URL bastant llarga. Aquesta URL obre una pàgina amb el widget que hem configurat. Si l'únic que ens interessa és fer servir el widget tal qual, ja hem acabat.

Però per a qui estigui interessat en comprendre com està construït aquest enllaç (per exemple, perquè vulgui modificar-lo manualment, sense haver de passar pel wizard), vegem com està format, descomposant un exemple en els seus paràmetres:

```
http://sensors.fonts.cat/widget/  
  name=compass  
  service=http://demo.geomati.co/sos/json  
  offering=http://sensors.portdebarcelona.cat/def/weather/offerings#10m  
  feature=http://sensors.portdebarcelona.cat/def/weather/features#P3  
  property=http://sensors.portdebarcelona.cat/def/weather/properties#31  
  refresh_interval=5  
  lang=en
```

Nota: Una URL vàlida ha de codificar cada paràmetre utilitzant la funció estàndard de javascript `encodeURIComponent` (o el seu equivalent en un altre llenguatge). Per claredat, en l'exemple es mostren els paràmetres ja descodificats.

Com es pot veure, els paràmetres de la URL són bàsicament els paràmetres d'entrada del widget. El formulari del wizard ens presenta els noms de tots els Offerings, features i properties, però els paràmetres del widget usen els seus corresponents identificadors. El wizard interroga el servei SOS per nosaltres i recupera tots els possibles parells de nom-identificador. En cas de voler accedir als identificadors manualment, es pot fer a través de l'operació `GetCapabilities`.

La URL també conté un parell de paràmetres extra que no són estrictament paràmetres de configuració del widget:

- El primer, “name”: És el nom del widget a crear.
- El darrer, “lang”: S'utilitza per indicar l'idioma en què volem veure els textos del widget. És un paràmetre opcional, i el seu valor per defecte és l'anglès (“en”). Altres llengües suportades són l'Espanyol (“és”) i el Català (“ca”).

L'opció “incrusta” simplement embolcalla l'enllaç anterior dins un element HTML `<iframe>`, de manera que pugui ser usat com a component en altres pàgines:

```
<iframe src = "... " width = "570" height = "380" frameBorder = "0"> </iframe>
```

L'amplada i alçada de l'`<iframe>` vénen determinades en primera instància per les mides inicials indicades al formulari del wizard, però poden canviar-se des del mateix wizard simplement redimensionant el requadre de “Vista del Wizard” (vegeu el control per redimensionar a la cantonada inferior esquerra de la vista).

2.3 Ús des de Javascript

La forma més flexible d'usar els widgets és per programació. Simplement s'ha de incloure la llibreria de Sensor widgets a la pàgina, que està disponible a <http://sensors.fonts.cat/js/SensorWidgets.js>, i instanciar el widget usant la factoria `SensorWidget`, que pren 3 paràmetres:

```
SensorWidget (nom, configuració, element);
```

El nom del widget és una cadena de text, la configuració és un objecte on llurs propietats són els paràmetres de configuració, i l'element és el DOM Element on es dibuixarà el widget.

La forma més pràctica de crear un widget programàticament és usant el wizard i copiant i enganxant el tros de codi javascript que ens ofereix. A partir d'aquesta base, s'hi pot afegir dinamisme canviant algun dels paràmetres de configuració.

Vegeu un exemple pràctic d'integració a: <http://bl.ocks.org/oscarfonts/5ad801cf830d421e55eb>

Nota: La funció `SensorWidget` no retorna cap valor, però en alguns casos accepta funcions de callback com a paràmetre. Els widgets es creen de forma assíncrona. En cas d'error, es mostrarà un missatge a l'usuari en l'element on havia dibuixar-se el widget.

2.3.1 Fent crides SOS de baix nivell amb Javascript

Avís: L'accés directe a les operacions SOS de baix nivell és experimental. La API aquí descrita pot canviar en qualsevol moment.

La instància del client SOS s'obté de forma asíncrona:

```
getSOS(function(SOS) {
    // Cal indicar una URL d'un servei 52n SOS 4.x amb encoding JSON
    SOS.setUrl("http://sensorweb.demo.52north.org/sensorwebtestbed/service");
    // A partir d'aquí, es poden invocar els altres mètodes de SOS
});
```

Aquesta és la API:

```
SOS.getCapabilities(callback, error); // Obté la secció "contents" del
↳GetCapabilities.
SOS.describeSensor(procedure, callback, error); // Obté el document SensorML
↳convertit a una estructura JSON.
SOS.getFeatureOfInterest(procedure, callback, error); // Obté totes les
↳FeatureOfInterest del procedure indicat.
SOS.getDataAvailability(procedure, offering, features, properties, callback,
↳error); // Obté rang de dates vàlid per a cada combinació de procedure, feature
↳i property.
SOS.getObservation(offering, features, properties, time, callback, error); // Obté
↳les observacions per a la combinació de paràmetres donada.
```

On els paràmetres són:

- *callback* (funció) recollirà la resposta com un objecte Javascript (JSON parsejat).
- *error* (funció) de callback que es cridarà en cas que el servei SOS retorni un error.
- *procedure* (string) identificador de la procedure.
- *offering* (string) identificador d'offering.
- *features* (array de strings) llista de les Features Of Interest de les que es vol obtenir resposta.
- *properties* (array de strings) llista de les Observable Properties de les que es vol obtenir resposta.
- *time* l'instant (si és string) o rang de temps (si és array de 2 strings) per al que es vol obtenir resposta. Les dates s'indiquen en hora UTC, format "yyyy-mm-ddThh:mm:ssZ". També pot prendre el valor especial "latest" per obtenir la observació més recent disponible.

I la seva obligatorietat és:

- La funció de *callback* és sempre obligatòria, i la funció d'*error* és sempre opcional.
- Per a *describeSensor* i *getFeatureOfInterest*, és obligatori indicar la *procedure*.
- Per a *getDataAvailability* i *getObservation* els filtres (*procedure*, *offering*, *features*, *properties*, *time*) són opcionals. Indiqueu *undefined* en cas de no voler filtrar per un d'aquests conceptes.

2.4 Personalització de l'aspecte gràfic

Tots els widgets admeten un paràmetre opcional `custom_css_url`, que permet indicar la localització d'un full d'estils CSS amb regles que sobreescriguin l'estil per defecte dels widgets.

Tots els widgets estan continguts dins un element `<div>` amb dues classes: la classe `widget`, i una classe amb el nom del widget. Per exemple, la següent regla CSS aplicarà a tots els widgets:

```
.widget {  
  border: 2px solid black;  
}
```

Mentre que la següent s'aplicarà només per a widgets del tipus `compass`:

```
.widget.compass {  
  background-color: grey;  
}
```

Un altre element comú és la nota al peu, que es troba sota un element de la classe `footnote`. Pot canviar l'aspecte de la nota a peu:

```
.widget .footnote {  
  font-color: red;  
}
```

Fins i tot es poden ocultar certs elements del widget mitjançant CSS. Per exemple, el títol principal en un termòmetre:

```
.widget.thermometer h1 {  
  display: none;  
}
```

Per a regles de simbolització més específiques, es recomana inspeccionar el DOM del widget, i aplicar les regles CSS segons els elements observats.

3.1 Rumb (compass)

El widget “compass” pertany a la categoria dels widgets de “valor instantani únic”. Mostra l’últim valor disponible per a una Property específica, que en aquest cas expressa un azimuth o angle respecte al nord. El widget interrogarà le servidor de forma periòdica per actualitzar el valor mostrat.

Els seus paràmetres obligatoris són:

- “Service”, “offering”, “feature” i “property”: determinen la propietat que es vol mostrar. La propietat ha de prendre valors entre 0 i 360 (a la que se suposa una unitat de mesura de graus centessimals).
- “Refresh_interval” (en segons): és el temps entre dues interrogacions al servidor (el widget llança GetObservations periòdiques cada X segons).

Altres paràmetres opcionals:

- “Title”: Si no s’especifica, per defecte es fa servir com a títol el nom de la Feature.
- “display_utc_times”: Les dates del servei SOS s’obtenen en UTC, i per defecte es converteixen al fus horari local del navegador. Si es volen mostrar en UTC, posar aquest paràmetre a *true*.
- “Footnote”: Text opcional que apareixerà com una petita nota al peu.
- “Custom_css_url”: full d’estils css que s’aplicarà al widget.

3.2 Manòmetre (gauge)

Un altre widget de “valor instantani únic”, en aquesta ocasió per presentar valors de percentatge entre 0 i 100.

Paràmetres obligatoris:

- “Service”, “offering”, “feature” i “property”: determinen la propietat que es vol mostrar. La propietat ha de prendre valors entre 0 i 100 (a la que se suposa una unitat de mesura de tant per cent).
- “Refresh_interval” (en segons): és el temps entre dues interrogacions al servidor (el widget llança GetObservations periòdiques cada X segons).

Paràmetres opcionals:

- “Footnote”: Text opcional que apareixerà com una petita nota al peu.

- “display_utc_times”: Les dades del servei SOS s’obtenen en UTC, i per defecte es converteixen al fus horari local del navegador. Si es volen mostrar en UTC, posar aquest paràmetre a *true*.
- “Custom_css_url”: full d’estils css que s’aplicarà al widget.

3.3 Taula jQuery (jqgrid)

Mostra una **taula jqGrid** amb un conjunt d’observacions per a un període de temps determinat, on cada registre és una observació. La llista de resultats es mostra paginada i pot ser ordenada pels valors de qualsevol de les columnes (Hora, Feature, Propietat, Valor i Unitat de mesura).

Paràmetres obligatoris:

- “Service”, “offering”, un conjunt de “features” i un conjunt de “properties”: selecciona el conjunt de combinacions feature-property a mostrar.
- “Time_start” i “time_end”: Selecciona les observacions que cauen dins d’aquest rang de temps.
- “Title”: el títol del widget.

Paràmetres opcionals:

- “Footnote”: Text opcional que apareixerà com una petita nota al peu.
- “display_utc_times”: Les dades del servei SOS s’obtenen en UTC, i per defecte es converteixen al fus horari local del navegador. Si es volen mostrar en UTC, posar aquest paràmetre a *true*.
- “Custom_css_url”: full d’estils css que s’aplicarà al widget. Tingueu en compte que l’aspecte de jqGrid ve determinat pel tema jQuery-ui subjacent.

Nota: aquest widget depèn de jQuery, jQuery UI i el plugin jqGrid. És un widget bastant pesat i no gaire personalitzable (es va desenvolupar com un exercici d’integració amb una aplicació antiga existent). Es recomana l’ús d’altres widgets com el “table”, que van més en la línia del que pretenen ser els Sensor widgets: ser lleugers, compactes i flexibles.

3.4 Mapa (map)

Aquest widget és especial en diversos sentits. En primer lloc, mostra la resposta a una petició GetFeatureOfInterest, en lloc del cas més habitual en què un widget representa la resposta d’un GetObservation.

En segon lloc, és un widget molt configurable, a través d’alguns paràmetres complexos. Afortunadament la majoria de paràmetres són opcionals, de manera que el seu ús més elemental és de fet molt senzill.

Està basat en la llibreria de mapes [Leaflet](#).

Els únics paràmetres estrictament obligatoris són:

- “Service” i “offering”: Determinen l’offering dels quals mostrarem Features of Interest al mapa.

Això mostrarà un mapa amb les Features. Passant el ratolí per sobre d’una Feature, es mostrarà una petita etiqueta amb el nom de la feature.

Hi ha un altre parell de paràmetres formalment obligatoris (encara que poden deixar-se buits):

- “Features”: Podem seleccionar només algunes features per mostrar al mapa. Si no se n’indica cap, de fet es mostraran *totes* (no s’aplica cap filtrat). Però aquest paràmetre s’ha d’indicar igual, encara que sigui com a una llista buida.
- “Properties”: Si s’indiquen una o més properties, l’etiqueta de cada feature es convertirà de fet en un petit widget de tipus “panel”, mostrant els últims valors de cada property per a cadascuna de les features. Novament, pot indicar-se una llista buida de properties, i en aquest cas, NO es mostrarà cap valor.

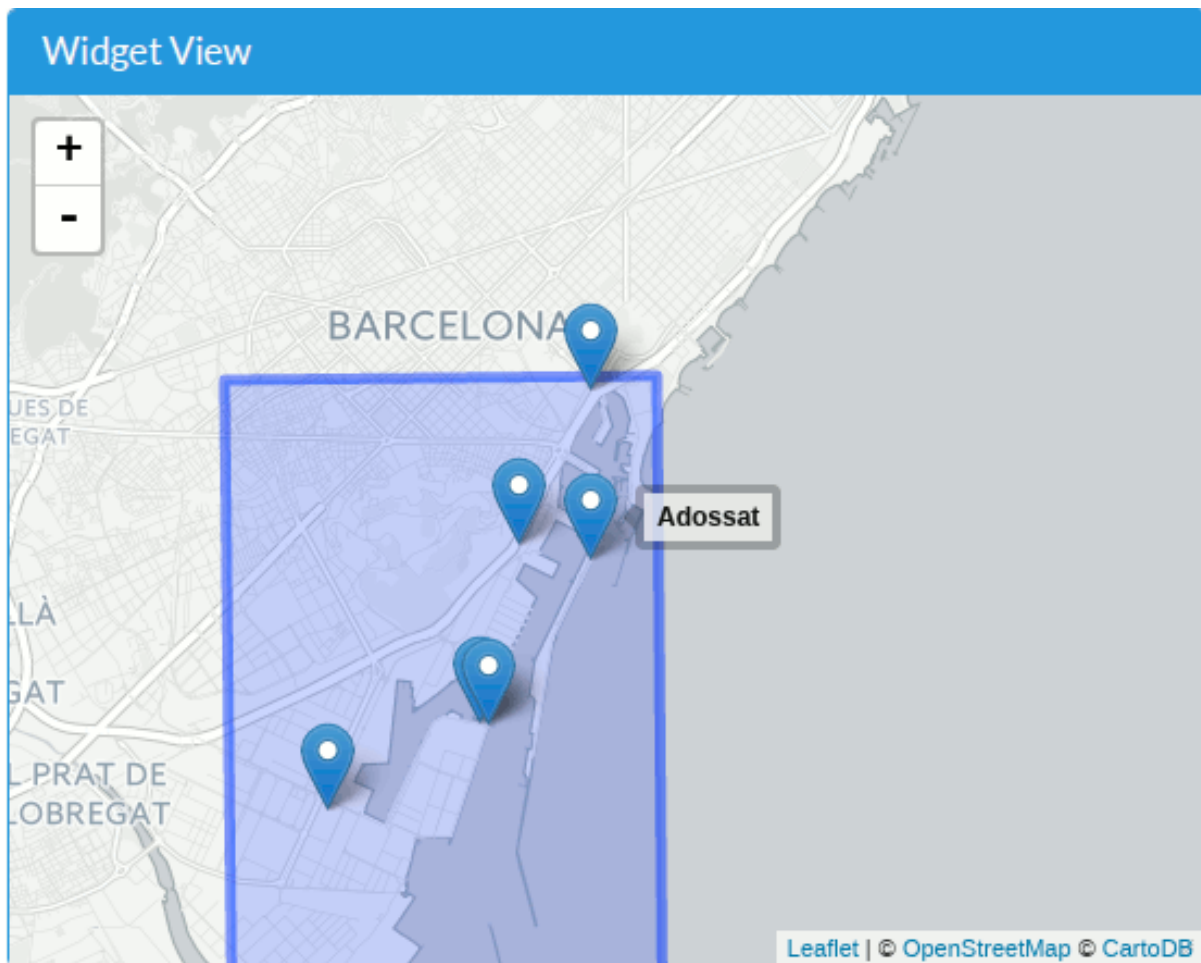


Fig. 3.1: Mapa simple on no s'han indicat ni features ni properties.

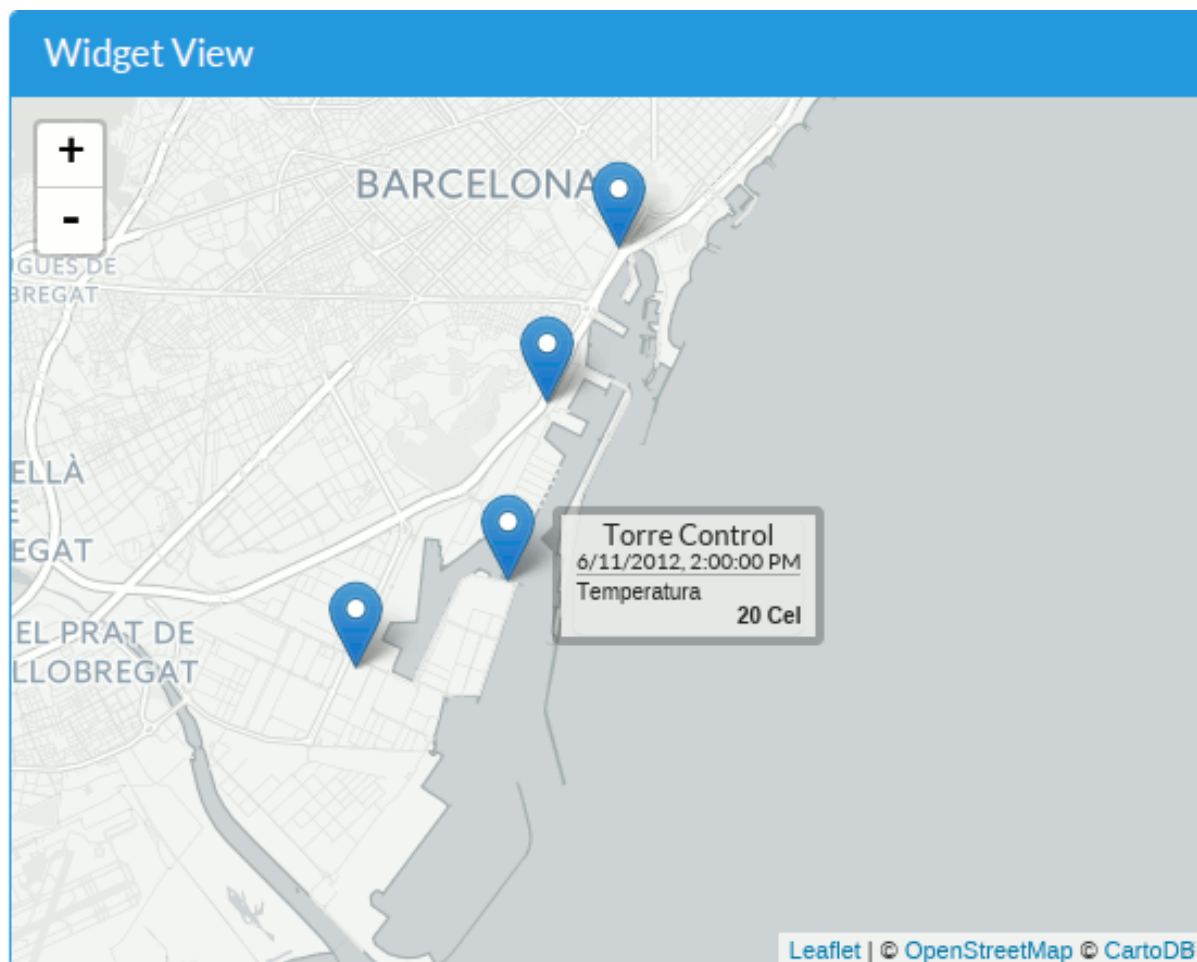


Fig. 3.2: Mapa on s'han seleccionat quatre features i una property, el valor de la qual es mostra a l'etiqueta.

El paràmetre opcional “permanent_tooltips”, si pren el valor “true”, farà que es mostrin totes les etiquetes permanentment, no només quan es passi el ratolí per sobre.

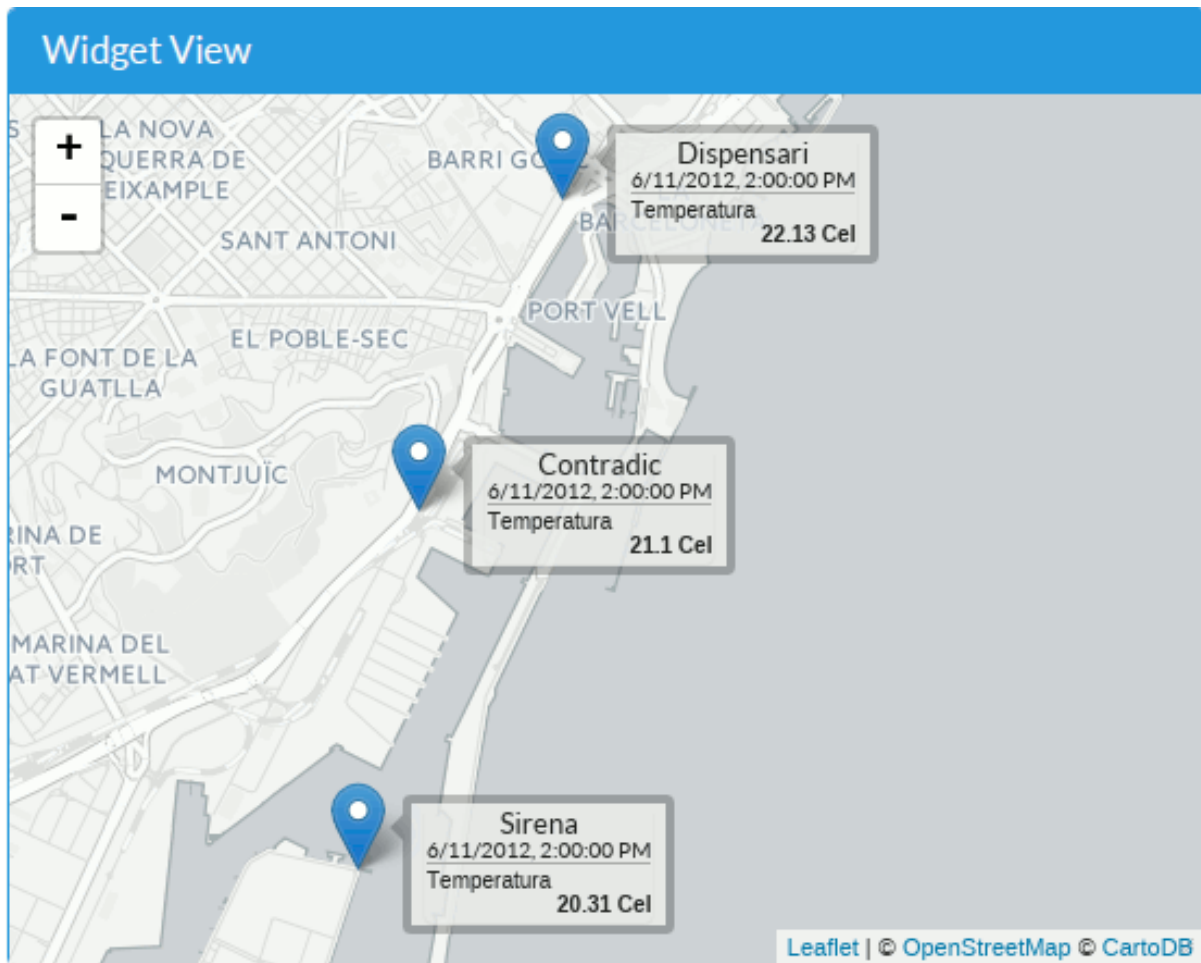


Fig. 3.3: Mapa amb etiquetes permanents.

Si els elements sobre el mapa apareixen a l'altra punta del món, és probable que s'hagi de canviar l'ordre dels eixos de coordenades. Afegint-hi el paràmetre opcional “swap_axis”=true, s'intercanviaran latitud i longitud, y es corregirà aquest efecte.

A més de les etiquetes, també podem vincular un sub-widget a cada feature, que es mostrarà en un globus en fer clic sobre ella. El paràmetre “popup_widget” pren com a valor un JSON amb la configuració del sub-widget. En aquesta configuració, els paràmetres “service”, “offering” i “feature(s)” s'obtenen del widget *pare* (el mapa), així que no s'han d'indicar. La propietat “name” indica quina classe de widget volem incrustar.

Per exemple, si volem que s'obri un globus contenint una gràfica temporal, hem d'indicar:

- “Name”: “timechart”,
- ... Tots els paràmetres del widget timechart, excepte “service” i “offering”.

És a dir:

```
{
  "name": "timechart",
  "title": "temperatures",
  "properties": [
    "http://sensors.portdebarcelona.cat/def/weather/properties#32M",
    "http://sensors.portdebarcelona.cat/def/weather/properties#32",
    "http://sensors.portdebarcelona.cat/def/weather/properties#32N"
  ],
}
```

```

"time_start": "2015-09-03T05:05:40Z",
"time_end": "2015-09-03T08:05:40Z"
}

```

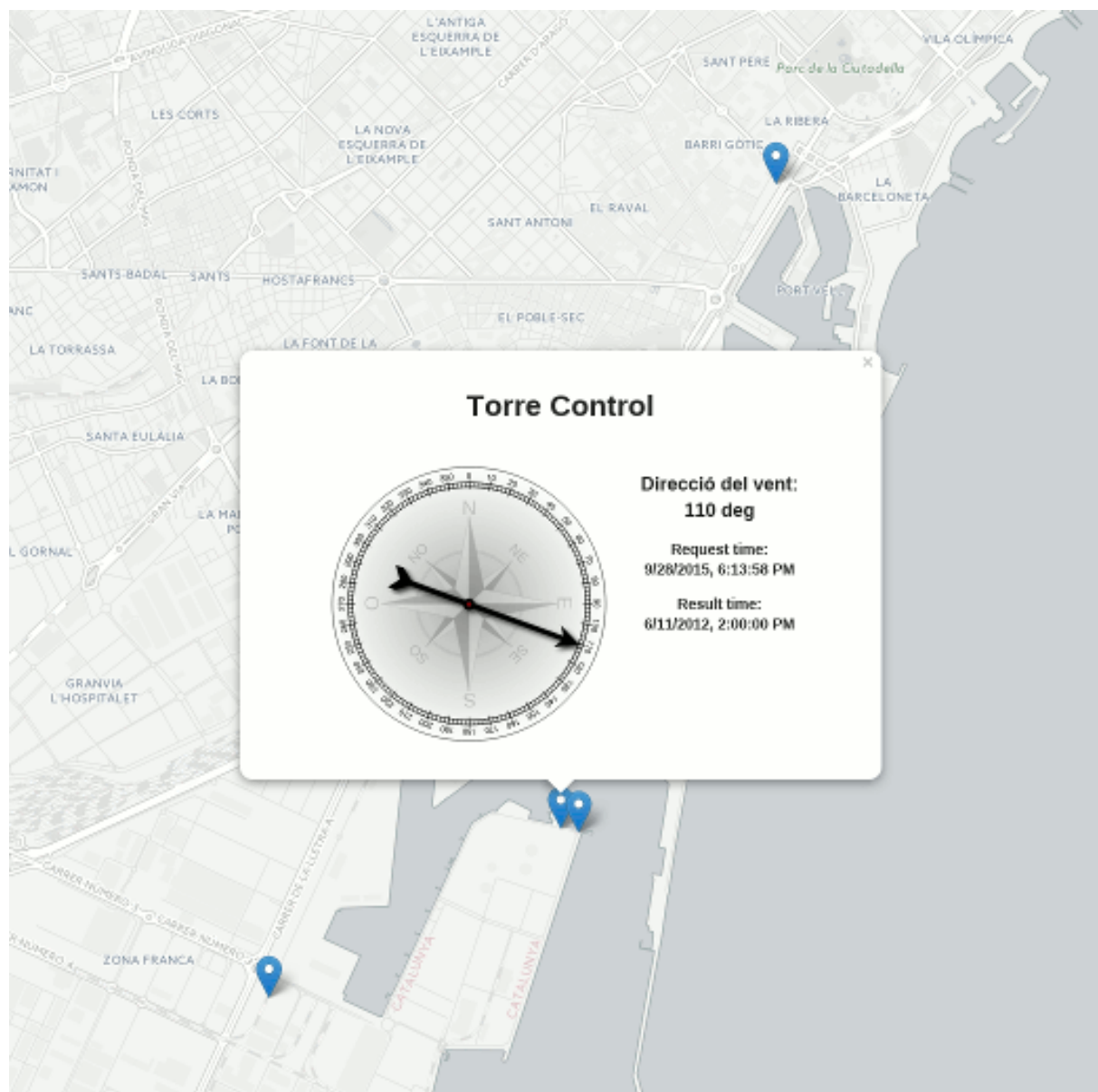


Fig. 3.4: Mapa amb un “popup_widget” de tipus “compass”.

A més de personalitzar les etiquetes i els globus amb detalls sobre cada feature, podem canviar la cartografia de base del mapa amb el paràmetre “base_layer”. Es poden especificar dos tipus de capa base:

- Una capa de tessell·les: Cal indicar una “url” i un conjunt de “options”. Per exemple:

```

{
  "url": "http://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png",
  "options": {
    "maxZoom": 19,
    "attribution": "&copy; <a href='http://www.openstreetmap.org/copyright'>↵  

↵OpenStreetMap contributors </a>"
  }
}

```

Els paràmetres “url” i “options” es corresponen respectivament amb els paràmetres del constructor `TileLayer` de

Leaflet “urlTemplate” i “TileLayer_options”.

Es pot escollir entre una bona col·lecció de capes de tesselles aquí: <http://leaflet-extras.github.io/leaflet-providers/preview/>

- Una capa WMS: Cal especificar “type”=”wms”, una “url” i un conjunt d’“options”. Per exemple:

```
{
  "type": "wms",
  "url": "http://geoserveis.icc.cat/icc_mapesbase/wms/service",
  "options": {
    "layers": "orto5m",
    "format": "image/jpeg",
    "attribution": "Ortofoto 1:5.000: CC-by <a href='http://www.icc.cat' ↵
↳target='_blank'>Institut Cartogràfic de Catalunya</a>"
  }
}
```

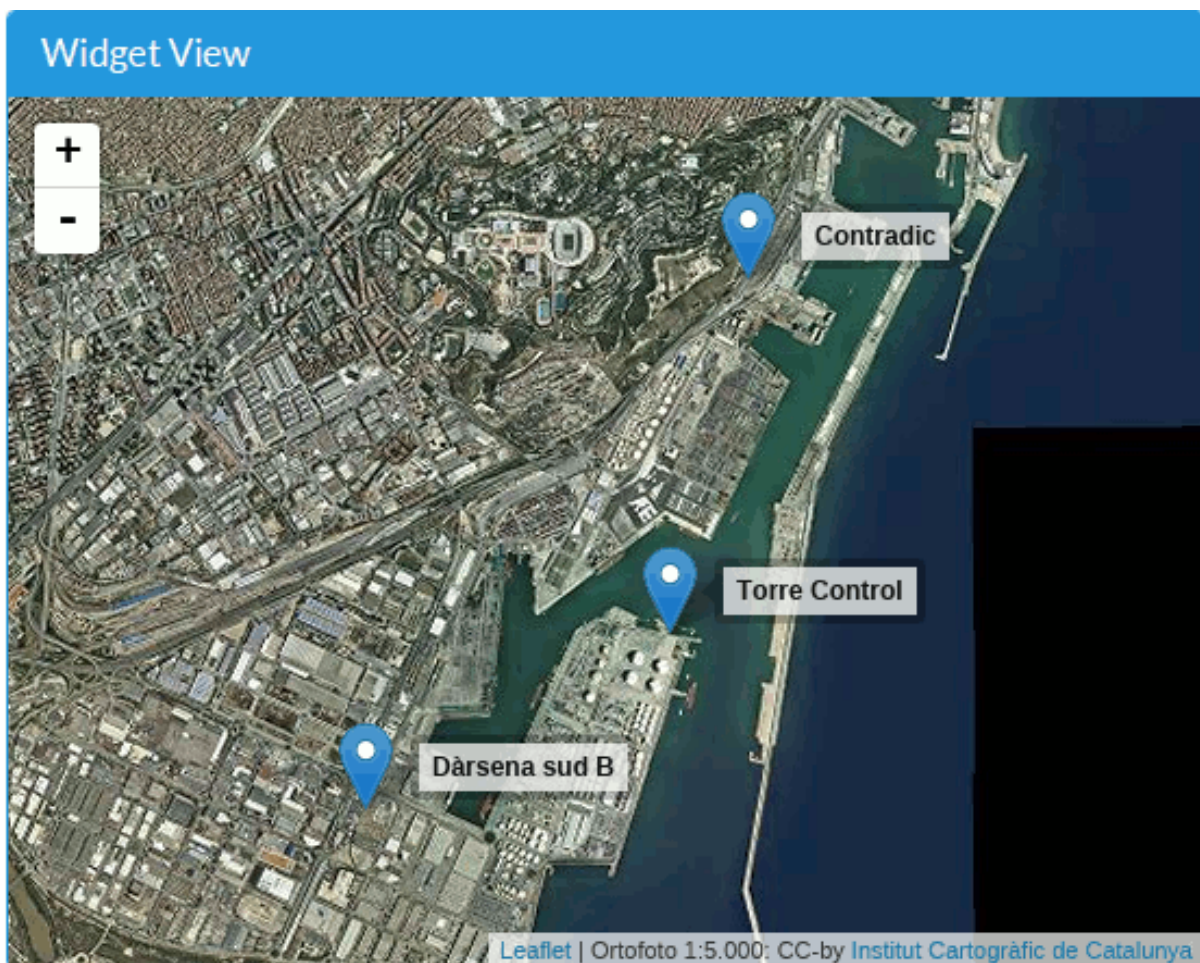


Fig. 3.5: Mapa amb cartografia WMS.

Els paràmetres “url” i “options” es corresponen amb els paràmetres del constructor `TileLayer.WMS` de Leaflet “baseUrl” i “TileLayer.WMS_options” respectivament.

Un altre paràmetre opcional és “max_initial_zoom”: Indica el nivell de zoom màxim a utilitzar en la vista inicial del mapa. Això evita acostar-se massa i perdre context cartogràfic, especialment útil quan es mostra una única feature puntual.

Quan hi ha molts marcadors sobre el mapa, s’aplica automàticament una funció d’agrupació dels mateixos (clustering). Si no es vol aplicar aquest clustering de forma automàtica, cal posar a `true` el paràmetre opcional “no_clustering”.

Si s'està utilitzant el widget amb Javascript, és possible capturar el "click" sobre els marcadors i obtenir-ne els detalls:

```
"on_click": function(marker) {
    console.log(marker.feature);
}
```

Finalment, els paràmetres opcionals habituals "display_utc_times", "footnote" i "custom_css_url" també estan disponibles.

Vegeu un **exemple funcional complet** aquí: <http://bl.ocks.org/oscarfonts/265d734349396cf4372c>

3.5 Panell (panel)

El widget "panel" s'usa per mostrar els últims valors d'un conjunt de propietats d'una Feature donada. Està construït com una Llista de Definicions (<dl>) d'HTML, compatible amb les classes CSS de Bootstrap. El contingut del widget s'actualitzarà automàticament de forma periòdica.

Els seus paràmetres obligatoris són:

- Els habituals "service", "offering" i "feature".
- Una llista de "properties" a mostrar.
- El "refresh_interval", en segons.

I els paràmetres opcionals: "title", "display_utc_times", "footnote" i "custom_css_url".

El panell també mostrarà la data de les observacions com a subtítol. En el cas que algun dels valors sigui d'una data anterior a la data comú, es mostrarà el valor en color vermell i es mostrarà la data per a aquesta observació en particular.

03 - Adossat		
Minutals	10 minutals	30 minutals
29/9/2015 11:31:00	29/9/2015 11:20:00	29/9/2015 11:00:00
Maximum wind speed 8.82 m/s	Maximum wind speed 10.19 m/s	Maximum wind speed 9.7 m/s
Wind direction 64.87 deg	Solar Radiation 0 W/m2*	Solar Radiation 0 W/m2*
Wind Direction Max 67.93 deg	*10/4/2015 10:00:00	*10/4/2015 10:00:00
Wind direction MIN 43.43 deg	Solar Radiation Max 0 W/m2*	Solar Radiation Max 0 W/m2*
Wind Speed 7.16 m/s	*10/4/2015 10:00:00	*10/4/2015 10:00:00
Wind Speed MIN 5.78 m/s	Solar Radiation MIN 0 W/m2*	Solar Radiation MIN 0 W/m2*
Wind Speed STD 0.87 m/s	*10/4/2015 10:00:00	*10/4/2015 10:00:00
	Wind direction 62.41 deg	Wind direction 70.16 deg
	Wind Direction Max 66.57 deg	Wind Direction Max 68.46 deg
	Wind direction MIN 28.99 deg	Wind direction MIN 25.27 deg
	Wind Speed 7.52 m/s	Wind Speed 7.16 m/s
	Wind Speed MIN 4.8 m/s	Wind Speed MIN 1.96 m/s
	Wind Speed STD 1.12 m/s	Wind Speed STD 1.02 m/s

Fig. 3.6: Tres widgets de tipus Panell, alguns d'ells mostrant valors amb un temps distint.

3.6 Barra (progressbar)

Un altre widget que mostra un valor instantani, aquest cop mostrat com una barra proporcional entre dos valors. És útil per mostrar gràficament on cau un valor respecte els seus valors límit. Es pot usar per a mostrar un percentatge si s'ajusten els valors mínim i màxim a 0 i 100 respectivament, en aquest cas seria molt similar a un widget de tipus "gauge" però mostrant el valor linealment. "ProgressBar" també pot prendre altres valors límit diferents, amb el que és més flexible que "gauge". A més el contingut és HTML, l'aspecte és més fàcil de personalitzar mitjançant CSS.

Paràmetres obligatoris:

- Els habituals “service”, “offering”, “feature” i “property”.
- “min_value” i “max_value”, que determinen els valors extrems.
- “refresh_interval” en segons.

I els paràmetres opcionals habituals: “display_utc_times”, “footnote” i “custom_css_url”.

3.7 Status (status)

El widget “status” mostra l’estat global de tot un offering d’un cop d’ull. Donat un offering, construeix una taula on cada cel·la representa una de les possibles combinacions de feature-property. Per a cada combinació, es mostra el darrer valor observat i la seva antiguitat. És una bona manera d’inspeccionar l’estat de salut d’un offering: Es veu ràpid si estan arribant noves observacions, i per a quins sensors.

Aquest widget està pensat com una eina de gestió (una espècie d’hiper-taula), i és més pràctica si es mostra a pantalla completa.

Els seus únics paràmetres obligatoris són “service” i “offering”.

I els paràmetres opcionals habituals: “display_utc_times”, “footnote” i “custom_css_url”.

3.8 Taula (table)

Donats un feature i un període de temps, un widget “table” mostra les observacions d’un conjunt de propietats al llarg del temps. És similar a “jqgrid” però proporciona una vista més compacta. El widget és una simple taula HTML amb classes CSS compatibles amb Bootstrap.

Paràmetres:

- Els habituals “service”, “offering” i “feature”.
- Una llista de “properties” a mostrar.
- “time_start” i “time_end”: Període de temps del que volem obtenir observacions.
- I el “title”.

A més dels paràmetres opcionals comuns: “display_utc_times”, “footnote” i “custom_css_url”.

3.9 Termòmetre (thermometer)

Un altre widget de tipus “valor instantani únic”, tal com Compass i Gauge, però per mostrar una temperatura ambiental en graus Celsius.

Mostra el dibuix d’un termòmetre que pot prendre valors dels -24°C als 56°C. També es mostra el valor numèric. Com altres widgets de la seva categoria, incorpora un mecanisme d’actualització periòdica.

Paràmetres obligatoris:

- “service”, “offering”, “feature” i “property”: Determinen la propietat de la qual volen mostrar-se mesures. Se li suposa graus centígrads com a unitat de mesura.
- “Refresh_interval” (en segons): el temps entre actualitzacions del valor.

Altres paràmetres opcionals:

- “footnote”: Text opcional que apareixerà com una petita nota al peu.
- “display_utc_times”, per mostrar l’hora en temps universal i no en el fus horari local.
- “custom_css_url”: full d’estils css que s’aplicarà al widget.

3.10 Sèrie temps (timechart)

Donats una feature i un rang de temps, mostra els valors que van prenent certes propietats al llarg del temps. La seva interfície és la mateixa que el widget “table”, però els resultats es mostren sobre una gràfica.

Les gràfiques estan basades en la llibreria [Flot Charts](#), que al seu torn depèn de jQuery.

Paràmetres:

- Els habituals “service”, “offering” i “feature”.
- La llista de “properties” a mostrar.
- “time_start” i “time_end”: Període de temps del qual volem obtenir observacions.
- I el “title”.

Paràmetres opcionals:

- “colors”: Array de colors en format `#rrggbb`, que s’aplicarà en dibuixar les línies de cadascuna de les propietats.
- “callback”: Funció que es crida després d’instanciar el widget. Recull la instància del Flot chart com a paràmetre.

A més dels paràmetres opcionals comuns: “display_utc_times”, “footnote” i “custom_css_url”.

3.11 Rosa vents (windrose)

Aquest és un widget per a un cas d’ús molt específic: mostra estadístiques del règim de vents, on es pot apreciar d’una ullada la direcció i velocitat predominants del vent, així com la seva variabilitat al llarg d’un període de temps.

Nota: La gràfica polar resultant està basada en la llibreria [Highcharts](#). Aquesta llibreria és gratuïta per a usos no comercials, però **se n’ha d’adquirir una llicència per al seu ús comercial**.

Paràmetres obligatoris:

- “service”, “offering”, “feature”: determinen una localització, de la qual ha d’haver dades de direcció i velocitat del vent.
- “properties”: admet un array de dos (i només dos) propietats. Una serà la velocitat del vent en m/s , i l’altra la seva direcció en deg . Les observacions per a ambdues propietats s’han de produir a intervals regulars i de forma síncrona.
- “time_start” i “time_end”: el període de temps sobre el qual es descarregaran dades i s’extrauran les estadístiques.
- “refresh_interval” (en segons): temps entre actualitzacions del widget. Es recomanen valors de diversos minuts per no saturar el servidor, ja que la quantitat de dades a descarregar és gran, i les estadístiques sobre un període de temps més o menys llarg tampoc no canviaran bruscament.
- “title” el títol del widget.

Paràmetres opcionals:

- “subtitle”.
- “display_utc_times”, “footnote” i “custom_css_url”.

Així és com s’agrupen les dades per construir la gràfica de la rosa dels vents:

1. Els valors de direcció del vent es classifiquen en 16 sectors: N, NNE, NE, ENE, E, ESE, SE, SSE, S, SSW, SW, WSW, W, WNW, NW, NNW and N.

2. Per a cada sector, les velocitats del vent corresponents es classifiquen en rangs: 0-2 m/s, 2-4 m/s, 4-6 m/s, 6-8 m/s, 8-10 m/s i > 10 m/s.

Es dibuixa llavors una gràfica polar amb 16 columnes, en cadascuna de les quals s'hi apilen els diferents segments acolorits segons la seva velocitat, d'alçada proporcionas al recompte d'observacions d'aquest rang.

Nota: A diferència d'altres widgets, més lleugers i flexibles, aquest requereix que el servei SOS de què s'alimenta exposi les dades d'una manera molt concreta. A més, depèn d'una llibreria de gràfics no estrictament lliure. Però els resultats per al cas d'ús que cobreix són excel·lents. Així doncs, preneu aquest widget no com un cas de widget genèric i reusable, sinó com un exemple de l'*especialització* a la qual es pot arribar programant widgets propis. Per a desenvolupar els vostres propis widgets que us ajudin a expressar millor les vostres pròpies dades, consulteu el capítol sobre com contribuir al projecte (en anglès).

CAPÍTOL 4

Com contribuir al projecte

La documentació per a desenvolupadors està disponible únicament [en anglès](#).