



Overlays

OpenLayers vous permet de superposer de nombreux types de données différentes par dessus des sources de données diverses. Actuellement, il existe deux manières principales pour afficher des objets vectoriels superposé (overlays) dans OpenLayers, chacune avec des avantages et des inconvénients. Ce document vise à décrire les différences, et les façons dont chacune des méthodes peut être utilisées.

Overlay Basiques

Il existe deux différents types de rendu d'objet dans OpenLayers. Le premier est la gestion du *Overlays vectoriel* d'OpenLayers, qui utilise les capacités de dessin vectoriel du navigateur (SVG, VML, ou Canvas) pour afficher ces données. L'autre type est la gestion des *Marker de superposition* dans OpenLayers. Ce type de couche affiche des objets images HTML dans le DOM.

En général, la couche vecteur fournit plus de possibilité, avec la capacité de dessiner des lignes, des polygones, et plus encore. Les couches basées sur les vecteurs sont mieux maintenues, et ce sont sur celle-là que la plupart des nouveaux développement d'OpenLayers ont lieu. Il y a une meilleure gestion des différentes options de style et de possibilité de configuration sur les comportements et les interactions des couches avec les serveurs distants.

Cependant, la couche markers est maintenu pour une compatibilité arrière parce qu'il y a des choses que vous ne pouvez pas faire avec les vecteurs puisqu'ils ne sont pas encore implémenté et qu'ils fournissent un type différent d'interface pour l'enregistrement des événements.

Overlays vectoriel

Les couches vecteurs forment le cœur des overlays vectoriel. Les overlays vectoriels ont été créés pour ajouter des ensembles d'objet OpenLayers.Vectors à la carte. Cela peut être plusieurs types de géométrie :

- Point / MultiPoint
- Line / MultiLine
- Polygon / MultiPolygon

Ils sont stylés en utilisant les propriétés d'OpenLayers.Style/OpenLayers.StyleMap.

Exemples :

- Exemple StyleMap :
Utilise "Rules" pour déterminer le style des attributs en se basant sur la propriété des objets. C'est utile pour le rendu basé sur les données des attributs comme la population.
- Exemple Context :
Utilise une fonction Javascript personnalisée pour déterminer les propriétés de style de l'objet. Cette exemple montre comment utiliser le cadran du monde dans lequel se trouve l'objet pour déterminer ses couleurs. Des règles similaires peuvent être utilisées pour réaliser des calculs sur une propriété d'un objet pour générer une valeur du style (comme la taille).
- Exemple Rotation :
Les objets vecteurs gèrent le styling avancé, comme la rotation d'objet. Cela peut être utilisé, par exemple, pour afficher la direction des véhicules, la direction du vent, ou d'autres attributs basés sur la direction.
- Exemple de style à valeur unique :
Une étude de cas commune est de prendre une valeur de style spécifique basée sur une paire clé/valeur de cartographie d'objet. Cet exemple montre comment réaliser cela.

Interaction

l'interaction des couches vectorielles sont réalisé via SelectFeatureControl. Ce contrôleur permet la sélection d'objets, en utilisant les événements DOM pour capturer sur quel objet on a cliqué.

Pour prendre en charge des événements d'objet sur une couche vectorielle, vous utilisez

SelectFeatureControl, en combinaison d'un event listener enregistré sur la couche, sur l'événement 'featureselected'.

```
function selected (evt) {
    alert(evt.feature.id + " selected on " + this.name);
}
var layer = new OpenLayers.Layer.Vector("VLayer");
layer.events.register("featureselected", layer, selected);
```

Une fois cela effectué, vous pouvez ajouter un contrôleur de sélection d'objet à votre carte :

```
var control = new OpenLayers.Control.SelectFeature(layer);
map.addControl(control);
control.activate();
```

L'appel activé déplacera la couche vectorielle au premier plan de la carte ; tous les événements auront donc lieu sur cette couche.

Pour la version 2.7 d'OpenLayers, il n'y a pas de gestion pour la sélection d'objets pour plus d'une couche simple à la fois. La couche qui est actuellement utilisée pour la sélection est la dernière couche dont la méthode `.activate()` du contrôleur de sélection d'objet attaché qui a été appelée.

Types de couches

- *Vecteur* (Classe de base)
- *GML* – peut charger plusieurs types de données différents.
- *PointTrack*
- *WFS*

Marker de superposition

Les markers ne gère que les géométries ponctuelles. Ils ne sont stylés qu'avec la classe `OpenLayers.Icon`. Ils ne gèrent pas les lignes, polygones et autres objets complexes. Leur méthode d'interaction diffère significativement des couches vectorielles.

En générale, les markers sont l'ancienne manière d'interagir avec les données géographiques dans le navigateur. La plupart du nouveau code devrait, si possible, utiliser une couche vectorielle

à la place de couches markers.

Interaction

L'interaction sur les couches markers se fait en enregistrant des événements sur la propriété événement des markers individuels :

```
var marker = new OpenLayers.Marker(lonlat);
marker.id = "1";
marker.events.register("onmousedown", marker, function() {
    alert(this.id);
});
```

N'importe quel nombre d'événements peut être enregistré, et différents événements peuvent être enregistrés pour chaque objet.

Types de couche

- Markers (Classs de Base)
- GeoRSS
- Texte
- Boxes (Utilise des makers "Box" spéciaux)

Transition d'une couche Texte ou GeoRSS vers une couche vecteur

Plusieurs applications OpenLayers utilisent des couches Texte Layer ou GeoRSS, chacune interprétant un fichier (valeurs séparées par des tab) et affichant des markers aux coordonnées fournis. Lors d'un clic sur un des markers, une popup s'ouvre et affiche le contenu des champs name et description de la localisation.

Ce comportement est relativement facile à réaliser en utilisant les couches vecteurs, et offre plus de possibilités de configuration des comportements lors du clic sur un objet. Au lieu d'être forcé d'utiliser des popups, vous pouvez envoyer une nouvelle url au navigateur, ou changer le comportement d'une autre manière.

Charger des données

Pour copier le comportement de chargement des couches *Texte Layer* ou *GeoRSS*, il y a deux options :

- Utiliser une couche *GML* – expliqué dans ce document.
- Utiliser une couche *Vecteur*, avec un protocole et une stratégie.

Dans tous les cas, la manière de contrôler le comportement d'une sélection d'objet est le même.

Charger des données avec une couche GML

La couche *GML* est une couche simple de donnée “qui charge les données d'une URL en une fois”. Vous lui fournissez une URL et un format à utiliser, et elle chargera les données de cette URL et l'interprétera en fonction du format.

```
var layer = new OpenLayers.Layer.GML("Layer Name",
    "http://example.com/url/of/data.txt",
    { format: OpenLayers.Format.Text });
map.addLayer(layer);
map.zoomToMaxExtent();
```

Cela chargera vos données et les affichera comme point sur la carte.

Sémiologie des données

Certains formats de données n'incluent pas d'information sur la sémiologie, comme GeoRSS. Afin de permettre une correspondance entre le style d'OpenLayers par défaut et le marker par défaut dans OpenLayers, vous devez créer un StyleMap qui correspond au style par défaut d'OpenLayers :

```
var style = new OpenLayers.Style({
    'externalGraphic': OpenLayers.Util.getImagesLocation() + "marker.png",
    'graphicHeight': 25,
    'graphicWidth': 21,
    'graphicXOffset': -10.5,
    'graphicYOffset': -12.5
});
```

```
var styleMap = new OpenLayers.StyleMap({'default':style});

var layer = new OpenLayers.Layer.GML("Layer Name",
  "http://example.com/url/of/data.txt",
  {
    format: OpenLayers.Format.GeoRSS,
    styleMap: styleMap
  }
);
```

L'utilisation d'une "carte de style" comme celui-ci n'entraînera aucune différence lorsque votre objet est sélectionné. Pour créer un style différent pour la sélection – par exemple, avec un marker de couleur différente – vous pouvez créer un nouvel objet style et créer votre StyleMap comme ceci :

```
var styleMap = new OpenLayers.StyleMap({
  'default': style,
  'select': selectStyle
});
```

Pour plus d'information sur la sémiologie des objets, lisez la documentation [Styles](#) ou [StyleMap](#).

Afficher des Popups

Les couches [Texte](#) et [GeoRSS](#) ouvrent des popups contenant le titre et le texte de la description pour l'objet cliqué. Il est aisé de répliquer ce comportement dans votre application.

Définissez d'abord un ensemble de fonction pour gérer votre popup.

```
function onPopupClose(evt) {
  // 'this' est la popup.
  selectControl.unselect(this.feature);
}

function onFeatureSelect(evt) {
  feature = evt.feature;
  popup = new OpenLayers.Popup.FramedCloud("featurePopup",
    feature.geometry.getBounds().getCenterLonLat(),
    new OpenLayers.Size(100,100),
    "<h2>" + feature.attributes.title + "</h2>" +
```

```
        feature.attributes.description,  
        null, true, onPopupClose);  
  
    feature.popup = popup;  
    popup.feature = feature;  
    map.addPopup(popup);  
}  
  
function onFeatureUnselect(evt) {  
    feature = evt.feature;  
    if (feature.popup) {  
        popup.feature = null;  
        map.removePopup(feature.popup);  
        feature.popup.destroy();  
        feature.popup = null;  
    }  
}
```

Nous définissons ensuite deux handlers d'événements sur la couche pour appeler ces fonctions appropriées. Nous utilisons la définition de la couche ci-dessus, et supposons que la couche a été ajouté à la carte.

```
layer.events.on({  
    'featureselected': onFeatureSelect,  
    'featureunselected': onFeatureUnselect  
});
```

La combinaison de ces deux sections de code va provoquer l'ouverture d'un popup par la carte lorsque l'objet est sélectionné, puis fermera la popup lorsque l'objet est désélectionné ou le bouton "fermer" est pressé.

Le code HTML dans le quatrième argument du constructeur `FramedCloud` est basé sur le type de données que vous analysez. Cet exemple est basé sur la couche `Text`, mais vous pouvez faire de même avec une couche `KML` en remplaçant 'titre' par 'nom'. La couche `GeoRSS` pourrait utiliser la propriété `feature.attributes.link` en plus, pour créer un lien vers l'objet.

Il est à noter que ce contenu – passé au constructeur `FramedPopup` – est définie en utilisant `innerHTML`, et comme tel, est soumis à des attaques XSS si le contenu en question n'est pas fiable. Si vous ne pouvez pas faire confiance au contenu de vos fichiers sources, vous devez utiliser un certain type de décapage pour supprimer tout contenu malveillant avant de définir le contenu contextuel pour protéger votre site contre les attaques XSS.

Une fois que vous avez fait ceci, vous pouvez personnaliser le comportement de votre couche au cœur du contenu. Changer le design de votre popup HTML, modifier le type de popup, ou modifier le comportement du clic pour ouvrir une nouvelle fenêtre à la place – tout est possible, et simple, avec les fonctionnalités fournies par les couches vectorielles et `SelectFeatureControl`.