

## Pinning Views to the Map and Map Positions

Previously in this chapter, you saw how to add the Zoom View to a Map View by pinning it to a specific screen location. You can pin any View-derived object to a Map View (including layouts and other View Groups), attaching it to either a screen position or a geographical map location.

In the latter case, the View will move to follow its pinned position on the map, effectively acting as an interactive map marker. As a more resource-intensive solution, this is usually reserved for supplying the detail “balloons” often displayed on mashups to provide further detail when a marker is clicked.

Both pinning mechanisms are implemented by calling `addView` on the `MapView`, usually from the `onCreate` or `onRestore` methods within the `MapActivity` containing it. Pass in the View you want to pin and the layout parameters to use.

The `MapView.LayoutParams` parameters you pass in to `addView` determine how, and where, the View is added to the map.

To add a new View to the map relative to the screen, specify a new `MapView.LayoutParams` including arguments that set the height and width of the View, the x/y screen coordinates to pin to, and the alignment to use for positioning, as shown below:

```
int y = 10;
int x = 10;
MapView.LayoutParams screenLP;
screenLP = new MapView.LayoutParams(MapView.LayoutParams.WRAP_CONTENT,
MapView.LayoutParams.WRAP_CONTENT,
x, y,
MapView.LayoutParams.TOP_LEFT);
EditText editText1 = new EditText(getApplicationContext());
editText1.setText("Screen Pinned");
mapView.addView(editText1, screenLP);
```

To pin a View relative to a physical map location, pass four parameters when constructing the new `MapView.LayoutParams`, representing the height, width, `GeoPoint` to pin to, and the layout alignment.

```
Double lat = 37.422134*1E6;
Double lng = -122.084069*1E6;
GeoPoint geoPoint = new GeoPoint(lat.intValue(), lng.intValue());
MapView.LayoutParams geoLP;
geoLP = new MapView.LayoutParams(MapView.LayoutParams.WRAP_CONTENT,
MapView.LayoutParams.WRAP_CONTENT,
geoPoint,
MapView.LayoutParams.TOP_LEFT);
EditText editText2 = new EditText(getApplicationContext());
editText2.setText("Location Pinned");
mapView.addView(editText2, geoLP);
```

Panning the map will leave the first `TextView` stationary in the upper left corner, while the second `TextView` will move to remain pinned to a particular position on the map.

To remove a View from a `MapView`, call `removeView`, passing in the View instance you wish to remove, as shown below:

```
mapView.removeView(editText2);
```