

How to Create and Register a Custom Observer

In the [previous page](#), we demonstrated the use of the ReportPublisher, and its Report object in order to send log and report messages through the publisher. In this section, we will demonstrate how to create a custom observer and register it with the ReportPublisher.

Creating a custom observer is a very simple one. Here is a custom observer example:

```
1 public class SampleObserver extends BaseObserver {
2
3     @Getter
4     private final List<String> entries;
5
6
7     public SampleObserver() {
8         super();
9         entries = new ArrayList<>();
10    }
11
12
13    @Override
14    public void update(ReportEntry reportEntry) {
15        entries.add("SampleObserver data is " + reportEntry.toString());
16    }
17 }
```

All you need to do in order to create an observer is to extend the BaseObserver class and override the update method. This very basic observer simply puts the data in a list of strings.

The next thing we will need to do is **register the observer with the publisher**.

You can do this operation at any time. It is also a very simple operation:

```
1 ReportPublisher.getInstance().register(sampleObserver);
```

That's it. From this line forward, our SampleObserver object will be notified of any log or report entry.

The structure of the ReportEntryObject

Here is the implementation of the ReportEntry object:

```
1 @Data
2 @Builder
3 public class ReportEntry {
4     ReportLevel reportLevel;
5     ReportSource reportSource;
6     @Nullable
7     Object additionalObject;
8     String msg;
9 }
```

So each ReportEntry holds the level, source, message, and additional (optional) object.

If, for example, we would like to filter out all log messages which are lower than a given level, this is one way to go about it:

```
1 @Getter
2 private final ReportLevel minReportLevel;
```

```
3
4 public SampleObserver(ReportLevel minReportLevel) {
5     this.minReportLevel = minReportLevel;
6 }
7
8 @Override
9 @Synchronized
10 public void update(ReportEntry reportEntry) {
11
12     if (reportEntry.getReportLevel().compareTo(minReportLevel) < 0)
13         return;
14
15     // The message is for us - do some stuff with it.
16     ...
17 }
```