

```

1      class PageContent
2      {
3          // Asynchronously instantiate a PageContent object, replaces constructor
4          public static async Task<PageContent> AsyncCreate(
5              string url,
6              History singletonHistory,
7              Favourites singletonFavourites)
8
9          // Navigate to a new page, add the new page to history
10         public async void Navigate(string url)
11
12         // Navigate to a page without adding to history
13         public async void NavigateNoHistory(string url)
14
15         // Navigate back through local history
16         public void Back()
17
18         // Navigate forwards through local history
19         public void Forwards()
20     }

```

Figure 1: Useful PageContent public method signatures with descriptions

Figure 2: Some useful EntryRecord public method signatures with descriptions

```

1      {
2          public class Example
3          {
4              public event EventHandler ExampleHandler = delegate { };
5
6              public void ExampleEvent()
7              {
8                  EventHandler handler = ExampleHandler;
9                  // No need to check if ExampleHandler is null because
10                 // it has the empty delegate assigned
11                 handler(null, EventArgs.Empty);
12             }
13
14             public void TriggerEvent()
15             {
16                 ExampleEvent();
17             }
18         }
19
20         [TestClass]
21         class Test
22         {
23             [TestMethod]
24             public void TestEventDelegate()
25             {
26                 bool eventTriggered = false;
27                 Example e = new Example();
28                 // Named delegates are more commonly used here
29                 e.ExampleHandler += delegate { eventTriggered = true; };
30                 Assert.AreEqual(eventTriggered, false);
31                 e.TriggerEvent();
32                 Assert.AreEqual(eventTriggered, true);
33             }
34         }
35     }

```

Figure 3: Using delegates to raise, trigger, and test events