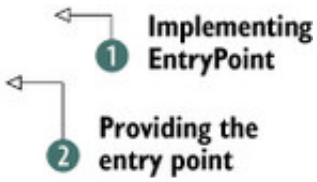

3.3. Enhancing the code

Here's `BasicProject`'s `EntryPoint` class (the class that's identified to have the `onLoad` method that's called when the application loads in the user's web browser):

```
public class BasicProject implements EntryPoint {

  public void onModuleLoad() {
    setUpGui();
    startHistoryHandling();
    setUpEventHandling();
  }
}
```

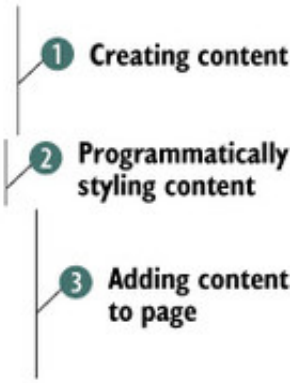


It implements the `EntryPoint` interface (1) and the `onModuleLoad` method (2) as you'd expect. Unlike `HelloWorld`'s entry point, where you added a `Label` to the `RootPanel`, `BasicProject`'s method calls two helper methods defined in the same class to set up the user interface and start GWT's history handling. You can put any valid Java code in the `onModuleLoad` method; our preference is to keep it clean and simple and use helper methods.

Setting up the GUI is achieved by the helper method shown in the following listing.

Listing 3.2. Setting up `BasicProject`'s GUI

```
private void setUpGui() {
  buildTabContent();
  wrapExistingSearchButton();
  insertLogo();
  createFeedbackTab();
  styleTabPanelUsingUIObject();
  styleButtonUsingDOM();
  RootPanel.get().add(feedback);
  RootPanel logoSlot = RootPanel.get("logo");
  if (logoSlot!=null)logoSlot.add(logo);
  RootPanel contentSlot = RootPanel.get("content");
  if (contentSlot!=null) contentSlot.add(content);
}
```



In listing 3.2's method you call several methods (1) that help you create content either by inserting the logo image or wrapping the Search button in the HTML page, as you'll see in section 3.4, or scraping out the content for the tabs that we'll look at in section 3.5.

A lot of the content is styled via plain CSS or a GWT theme, but you can also choose to apply styles using methods in the widget's underlying `UIObject` class or through GWT's `DOM` class (2). You'll see how all this is done in section 3.8.

The rest of the method is concerned with manipulating the browser DOM (web page), inserting GUI components onto the page (3). For now you'll build the user interface directly in your code because it's the simplest way to get across the points in this chapter. But from chapter 6 you'll use the UiBinder approach—GWT's approach to declarative user interface definition. There's no right or wrong way here, but after using UiBinder in chapter 6, we hope you'll find it the best way.

Let’s take a look at how to create user interfaces with widgets, panels, and events.