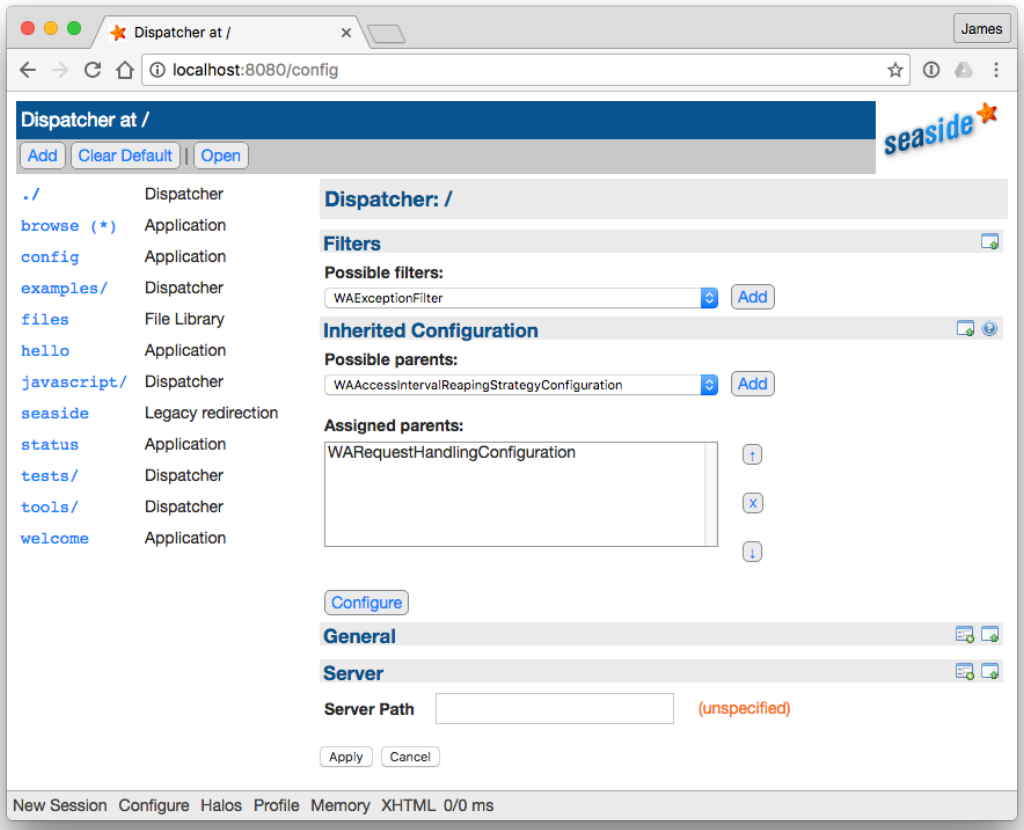
In this chapter we will explore some of the Seaside from the web browser's point of view, including examples available with the default installation of Seaside.

1. Launch the Seaside One-Click Experience (see Chapter 1 for details) and open a web browser on http://localhost:8080/browse.

   This page gives a list of the applications that have been registered with Seaside through the WADispatcher singleton (available by evaluating the Smalltalk expression 'WADispatcher default'). The dispatcher looks at the URL requested, and dispatches the request to one of these registered entry points. This is how the request from the client browser gets to the appropriate Smalltalk code (such as our 'HelloWorld' application).

2. Click on the config link to see the 'Dispatcher Editor.' This tool sets up the information for the Dispatcher Viewer we saw above.

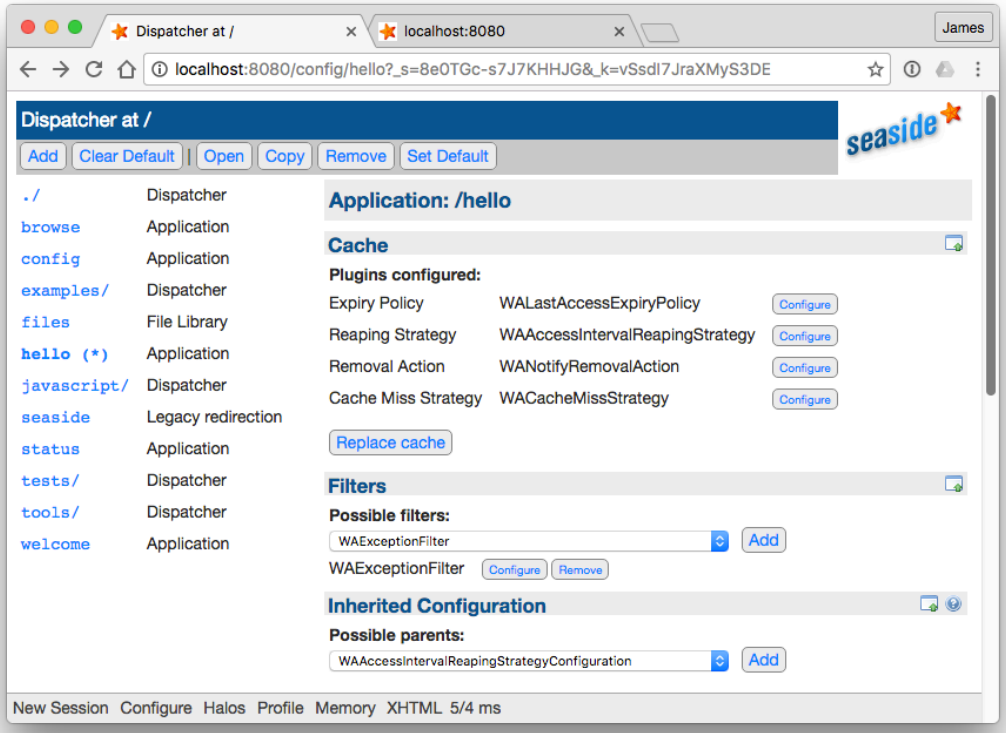


The left column lists each of the entry points and the initial selection is the root, which is just a dispatcher to some other web component. There are various types of components, including other dispatchers (examples/, javascript/, tests/, and tools/), applications (browse, config, hello, status, and welcome), a file library (files), and a legacy redirection (seaside). Selecting any component in the left column gives you some configuration information about that component.

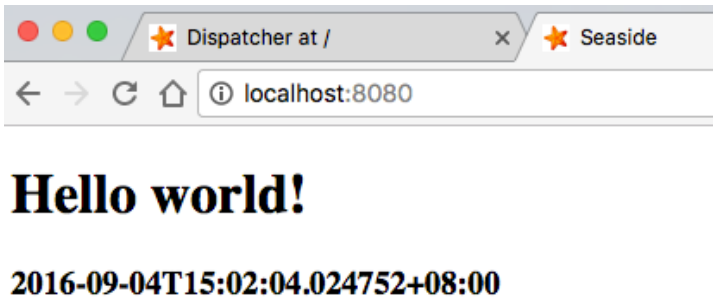
Note that there is a star next to the *default* component. The default component is the one provided by Seaside if you select the root (e.g., http://localhost:8080/). In chapter 2, step 13, we clicked 'Yes' when Seaside offered to change the default application from 'welcome' to 'browse'.

Note that near the top of the config page there is a ‘Clear Default’ button. Click the button then open a new web browser (window or tab) and navigate to the root. You should get an error.

Switch back to the config page (at http://localhost:8080/config if you did not save that page), select ‘hello’ from the links on the left, and then click the ‘Set Default’ button. This will make our Hello World application the default Seaside application.

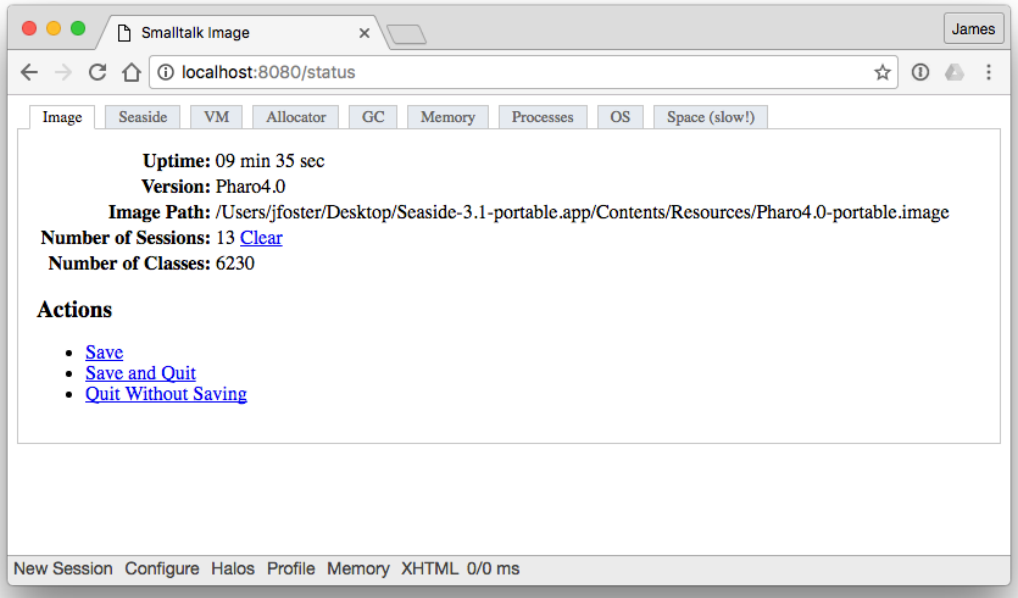


In your web browser, switch back to the tab or window with the error and refresh (or enter http://localhost:8080/). You should now see our Hello World application at the root.

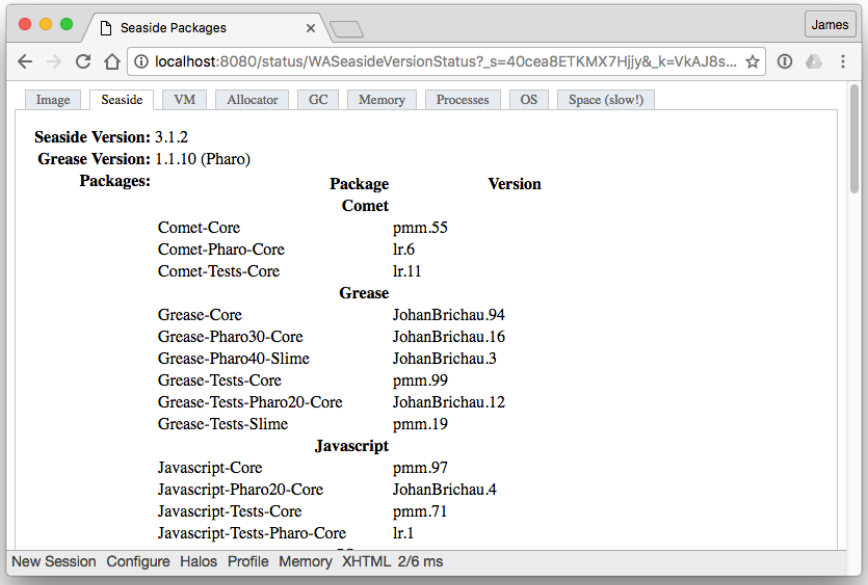


The config page provides a great deal of other configuration capabilities and demonstrates some of the capabilities of a Seaside component. We will not examine this area further at this time.

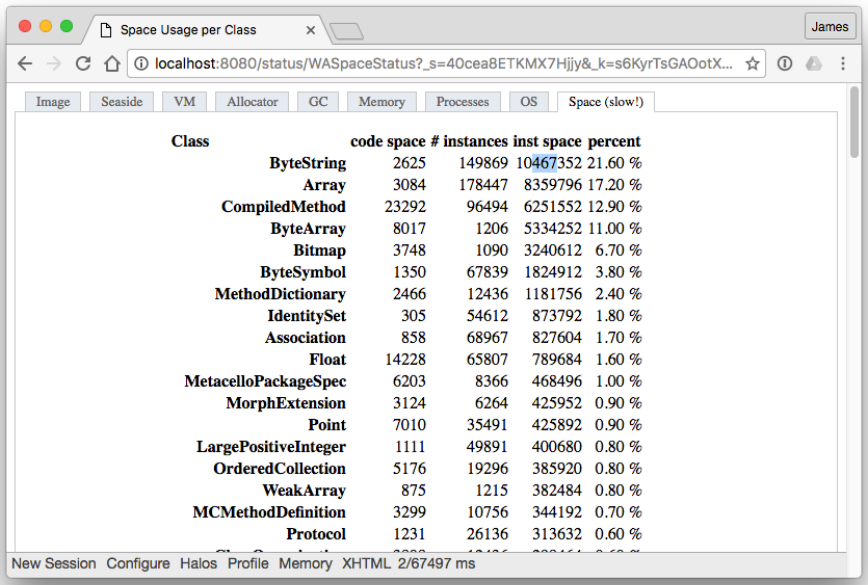
3. Enter http://localhost:8080/browse to return to the list of components and select 'Status' (or enter http://localhost:8080/status in your web browser's address bar). This will show you a list of tabs containing information about your Smalltalk environment. The first tab identifies the location of the file containing a snapshot (image) of the object space when it was last saved to disk. It also gives you the opportunity to make another snapshot of the current object space (by saving the image).



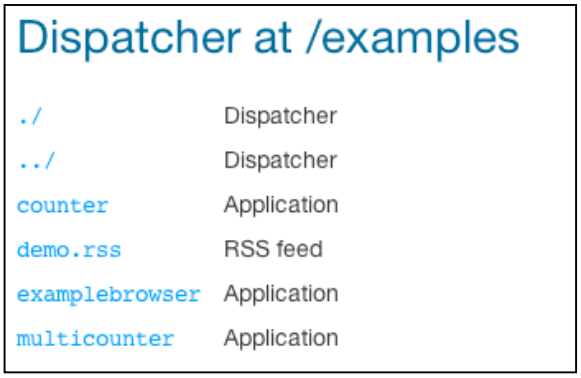
The ‘Seaside’ tab identifies the version of Seaside installed, and lists the version of various installed packages.



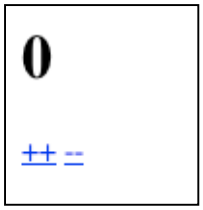
The final tab scans the entire object space and generates a report of objects in your object space. In this example, it took over 60 seconds to determine that there are almost 150 thousand strings in my environment, taking up about 10 MB of space.



4. Return to http://localhost:8080/browse and select 'examples' (a directory of other entry points). You should see the following:

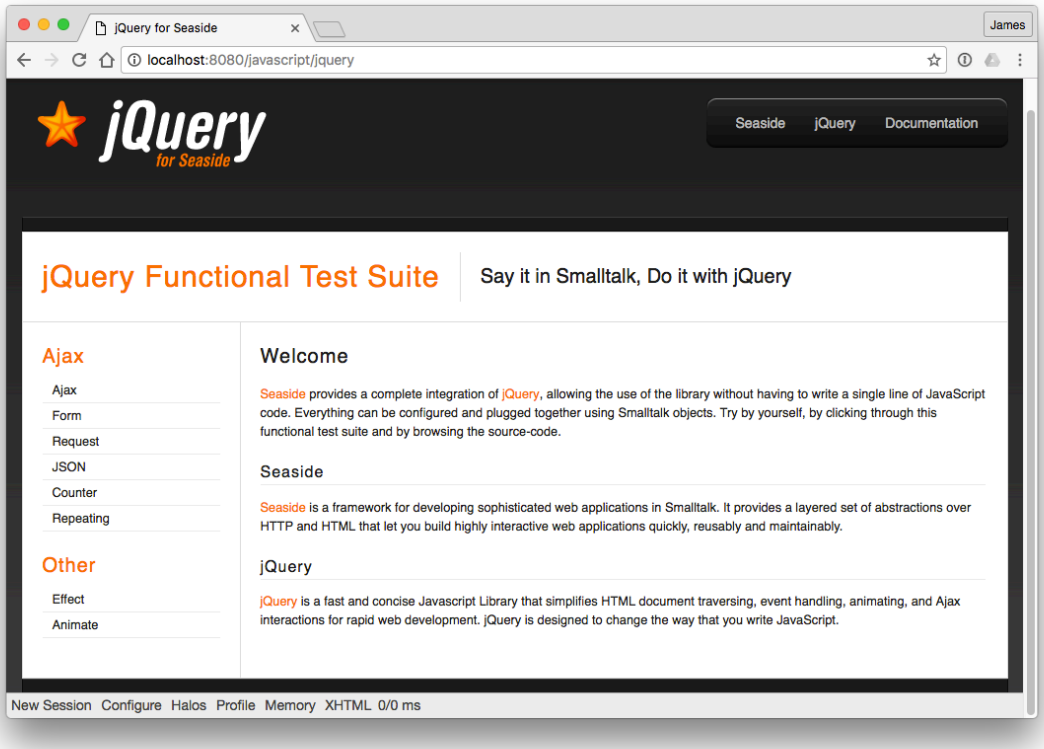


Click on the 'counter' and you will see the traditional Seaside example of an application that keeps state on the server. We will be using a slightly more complex example to explore some Seaside basics.



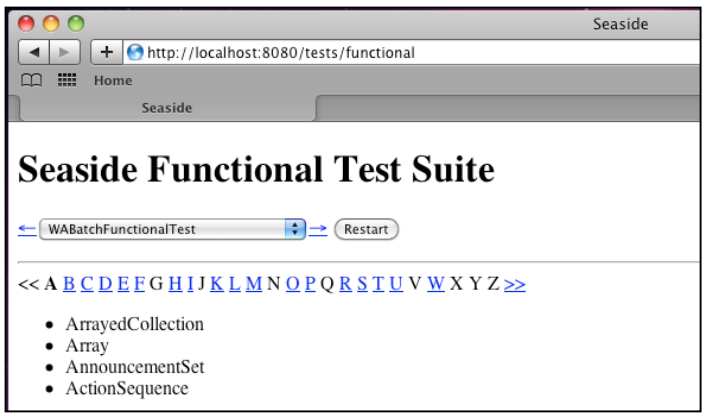
5. Use the web browser's back button to return to the examples page and then select the 'javascript/' link. Here you can find demos (with code) showing usage of jQuery and Scriptaculous.

Select the 'jquery' link to see a page describing Seaside support for JavaScript.

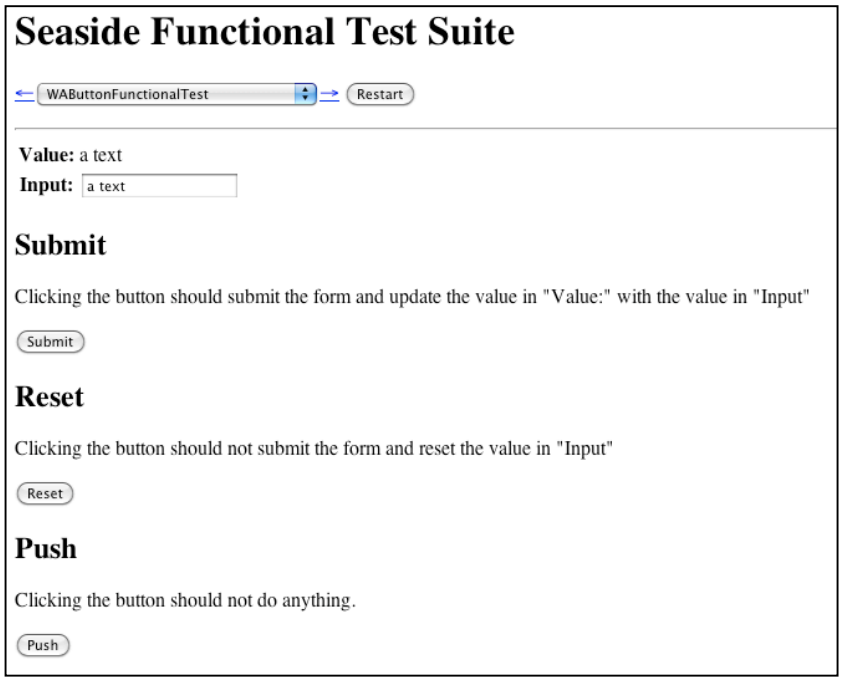


Select the 'Ajax' link in the left column. This gives a demonstration of how clicking on a button sends an Ajax request to Seaside to execute Smalltalk code on the server and send back a response (the current date and time).

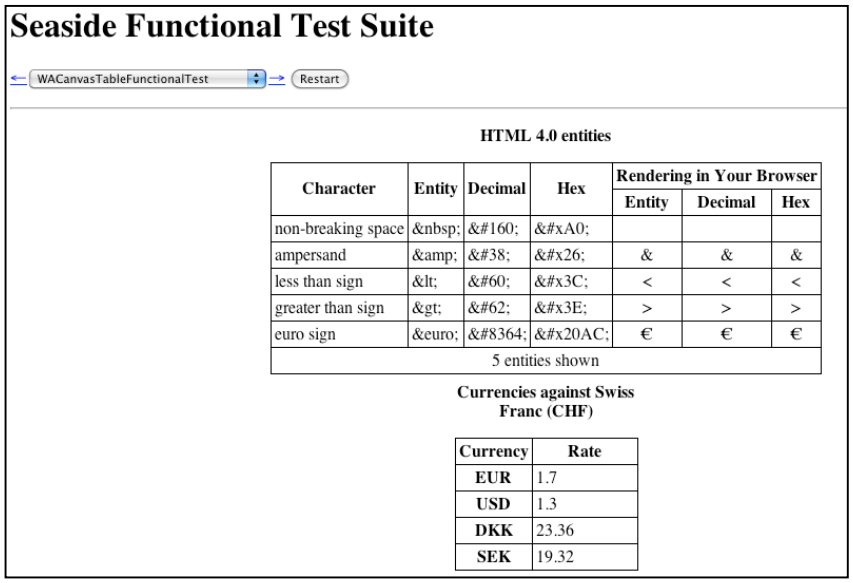
6. Return to http://localhost:8080/browse and select 'tests' and then 'functional.' The Functional Seaside Test Suite shows a drop-down list of various tests, with the test selected. The WABatchFunctionalTest encapsulates a list of class names and shows a horizontal list of the letters of the alphabet that can be used to jump to a particular place in the list. You can also use the previous ('<-') and next ('->') links to move through the list.



7. To reach the second test, click on the drop-down list showing 'WABatchFunctionalTest' and select the second item ('WAButtonFunctionalTest'). This will demonstrate the difference between various HTML button types ('submit', 'reset', and 'button'). If you enter text in the input field and click the Submit button, the new value will be returned to the server. If you enter text in the input field and click the reset button, the old value will be restored.

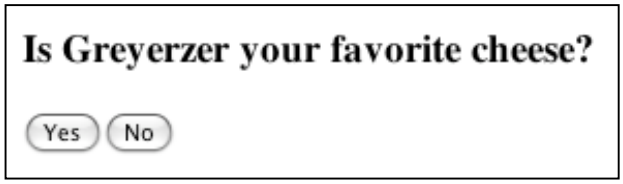


8. The WACanvasTableFunctionalTest demonstrates Seaside's ability to generate various table-related HTML elements, including <table>, <caption>, <colgroup>, <thead>, <tfoot>, <tbody>, <tr>, <th>, and <td>. This test gives you an example to use if you want to build a table in Seaside.

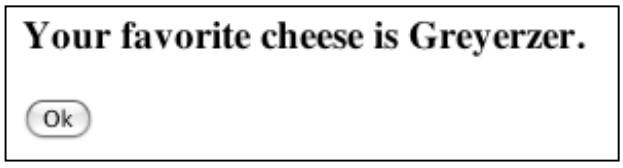


9. The WAFlowConvenienceFunctionalTest demonstrates three simple built-in components. The first component is the 'WAChoiceDialog' in which Seaside presents a list of options, add a prompt message, and then will return one of the items in the list of options or 'nil' if the user clicks 'Cancel'. (The object 'nil' is the single instance of UndefinedObject, an object that is the something that represents nothing in Smalltalk.) Any subclass of WAComponent (like our 'HelloWorld' class) can send 'chooseFrom:caption:' to self providing a list and a string and get back an answer.

If you click 'Ok' on the previous component, Seaside immediately presents the 'WAYesOrNoDialog' in which a message is presented with two buttons, 'Yes' and 'No' that will be returned as 'true' or 'false' Booleans. Any subclass of WAComponent can send 'confirm:' to self with a string and get back a Boolean.



If you clicking on 'Yes' in the previous component, Seaside immediately presents 'WAFormDialog' in which a message is presented with a simple 'Ok' button. Any subclass of WAComponent can send 'inform:' to self with a string to create this page.



10. The WAInputGetFunctionalTest demonstrates a variety of HTML input tags. By reviewing these examples you can get ideas of what can be done and then go to the test class to look at sample code. (As you progress through this tutorial you will learn more about finding classes in Pharo. In this example you would right-click on the first column of a System Browser, select the 'Find Class' menu item, enter WAInputTest' when prompted for a class name fragment, and then select 'WAInputTest' from the list.)