

Official Guide

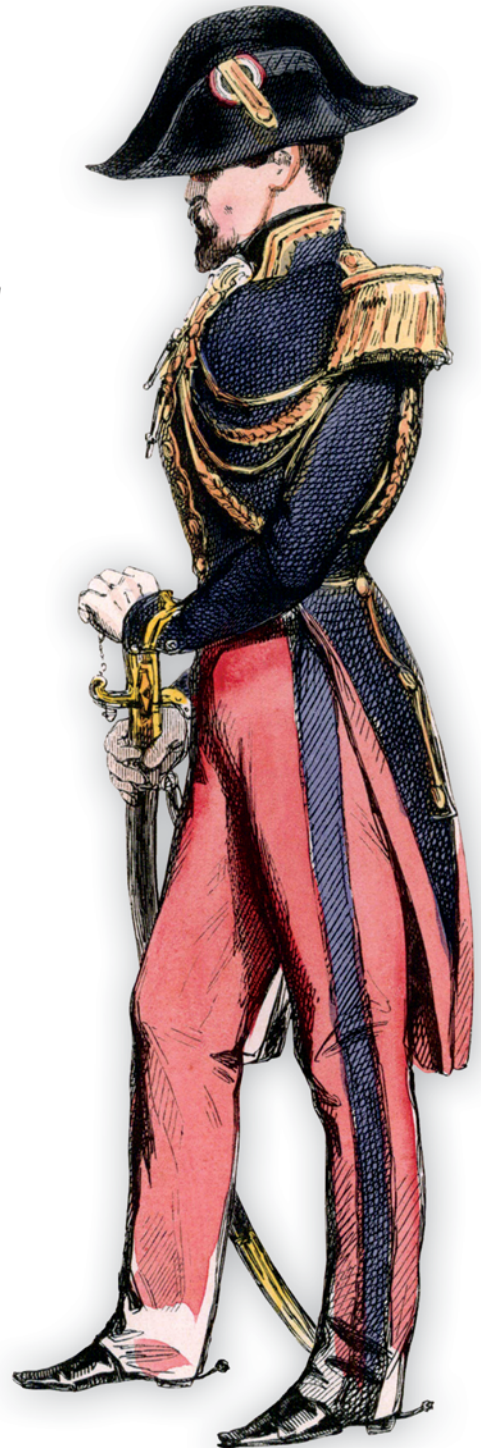
SAMPLE CHAPTER

Liferay

IN ACTION

Richard Sezov, Jr.

FOREWORD BY BRIAN KIM



MANNING



Liferay in Action

by Rich Sezov, Jr.

Chapter 1

Copyright 2012 Manning Publications

brief contents

PART 1 WORKING WITH LIFERAY AND PORTLETS 1

- 1 ■ The Liferay difference 3
- 2 ■ Getting started with the Liferay development platform 30

PART 2 WRITING APPLICATIONS ON LIFERAY'S PLATFORM..... 63

- 3 ■ A data-driven portlet made easy 65
- 4 ■ MVC the Liferay way 91
- 5 ■ Designing your site with themes and layout templates 128
- 6 ■ Making your site social 154
- 7 ■ Enabling user collaboration 176

PART 3 CUSTOMIZING LIFERAY..... 209

- 8 ■ Hooks 211
- 9 ■ Extending Liferay effectively 241
- 10 ■ A tour of Liferay APIs 263

Part 1

Working with Liferay and portlets

The first part of this book gives you an introduction to Liferay, showing you what it is, what it does, and how you can use its powerful features to implement your web site.

Chapter 1 introduces the portal landscape and how Liferay leads in that space. You'll learn how Liferay's users, roles, communities, and organizations work and how to navigate its interface. You'll also see how to run a Liferay development project, using best practices that have been proven to work.

Chapter 2 hits the ground running by introducing you to Liferay's development tools. You'll learn how to use the Plugins SDK to create Liferay projects, and you'll write your first portlet using the industry standard Portlet API.

This part of the book prepares you for the nuances of Liferay's development platform. When you finish reading it, you'll understand what Liferay is for and how you can use it to implement any web site.

The Liferay difference

This chapter covers

- Understanding portals then and now
- Exploring what Liferay is and how to work with it
- Defining basic portal concepts
- Using Liferay to design a portal

Everybody needs a web site these days. Whether you're building one for a company, for a service organization, or for personal reasons, you need one. And when trying to decide how to build it, you've probably found a dizzying array of choices running on a dizzying array of platforms. How do you go about choosing which platform is best?

If you're anything like me, you've got a list of a bunch of products. You created this list by looking at the feature claims of various software platforms that seem to do something close to what you want to do with your site. Now you're going through that list, testing the products, weighing their strengths and weaknesses against each other, and weighing those against how well those products' underlying platforms will fit into your infrastructure.

If Liferay Portal isn't on your list, you should put it at the top right away. Liferay Portal is a Java-based open source portal, containing an unprecedented number of

features that will help you to implement your site in as little time as possible. When you have Liferay on your list, let me respectfully submit that your search can end with Liferay Portal, which is hands down the best platform on which to build a web site.

I feel safe in saying this because I was a Liferay user for some time before I wound up working for the company. Yes, I took the red pill,¹ so to speak, but I've also experienced Liferay from the outside, and so I know what it's like to be doing that search for a platform. I can tell you from experience that you're going to find working with Liferay to be a pleasure, and you'll be happy to know that using the platform that Liferay offers will free you from limitations. It speeds up your development cycle and gives you features that you likely wouldn't have the time or inclination to build yourself. Usually, potential Liferay users focus on Liferay as a product—because it boasts such a huge range of features—but they don't stop to consider the rich development platform it offers. By the end of this chapter, you'll have a good understanding of what Liferay is all about and what it can do for your web site. And I have no doubt that you'll find many reasons to choose Liferay for your next development project.

Choosing Liferay is also safe: you're putting yourself in a group with some of the largest organizations (with the largest web sites) out there that have also chosen Liferay. If I can give you any advice, it would be to end your search with Liferay and begin learning how you can use the platform to build the site of your dreams.

This chapter will go over several important topics. I'll show you why Liferay calls itself a portal, what a portal used to be, and how Liferay pioneered getting past its early limitations. We'll then take a helicopter ride over Liferay's feature set to see what it can do at a high level. After this, we'll delve into how Liferay helps you structure a web site. You'll also get to see what Liferay looks like by default and how you can navigate around it. And finally, using all the information presented, I'll show you how to begin thinking about implementing your site using Liferay Portal.

To get your bearings, let's start by exploring why Liferay calls itself a portal and what that term has come to mean in the industry.

1.1 The Java portal promise: from disappointment to fulfillment

Liferay calls itself a portal. What do you commonly think of when you hear the word portal? As a big fan of sci-fi and fantasy, I tend to think of a doorway to another dimension or time like the portal that Kirk and Spock went through in *Star Trek*, chasing after McCoy to stop him from doing whatever he did to change the timeline. I'll tell you right away: Liferay Portal isn't that elaborate (but you've likely already figured that out). Why do we call it a portal? Let's start with the so-called official definition of a portal.

PORTAL A web-based gateway that allows users to locate and create relevant content and use the applications they commonly need to be productive.

¹ From the 1999 film *The Matrix*.

That definition comes from a bullet on a slide that I've used to teach Liferay to prospective users. I may even have written that bullet, but I'm not sure. Generally, the reaction I get is a narrowing of the eyes, some pursed lips, and then heads nodding up and down. This tells me that people want me to think that what I've just said makes sense, but they're being kind and reserving judgment on my teaching abilities because the definition made no sense at all.

The problem with definitions like that is that they try to say too much in one sentence. Liferay is many, many things, and you can't capture it all in one sentence. But just for fun, let's try it again.

PORTAL A single web-based environment from which all of a user's applications can run. These applications are integrated together in a consistent and systematic way.

That definition is a bit closer when viewed in the context of the web. When we talk about Liferay as a development platform, that's exactly what we mean. At its base, Liferay is a container for integrated applications. Those applications are what make the difference between Liferay and competing products. You're free to use the applications you like, write your own, and disable the rest. And this is what sets Liferay apart. Figure 1.1 shows how you can easily mix and match your applications with Liferay's.

As an analogy, think back to the 1980s and early 90s. If you bought a computer and you needed to use it to write something, you also bought a word processor program. If you then decided you wanted to calculate numbers with the computer, you bought a spreadsheet program. And if you needed to store and retrieve data of some kind (perhaps for a mailing list), you bought a database program. (Nobody created electronic slides back then; they used an overhead projector. And yes, I am dating myself.)

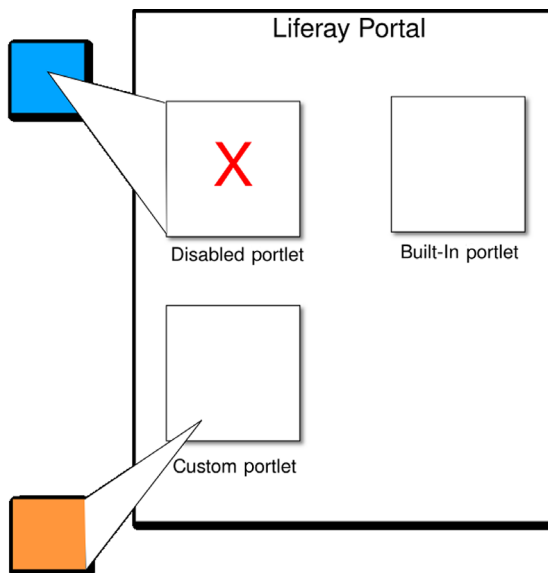


Figure 1.1 Liferay contains many built-in applications called portlets. If there are some you'll never use, you can disable them. You can also write and deploy your own portlets. These custom portlets are indistinguishable from portlets that ship with Liferay Portal.

Most of the time, people chose what was considered the best of whichever program they wanted. One vendor had the best word processor. Another had the best spreadsheet. A third had the best database. If you had to perform all three functions, you probably had three separate programs written by separate entities, but individually each was the best.

Pretty soon, people wanted to create graphs in their spreadsheets that they could insert into a word-processing document and send to a mailing list stored in the database. The problem was that all of these programs were created by different vendors, and they didn't always work together that well. Much effort on users' parts was spent trying to get programs to work together well.

You know the rest of the story. We wound up with office suites, consisting of programs written on the same platform that were designed to work together. Not only did this save us all some money (because buying the separate programs cost a fortune), but it also gave us a level of integration that had so far been unavailable.

The same thing is happening today with software on the web. Liferay is an engine for running web sites. Liferay consists of the base engine as well as the many applications that run on that engine. When you use this platform, your applications can integrate with the rest of Liferay's applications and make your users' experience seamless and smooth. Why? Because the integrated experience is far better than the nonintegrated experience. This is the difference that makes Liferay stand above all the other portals out there. I have to say that because there are some who view the word portal with disdain, sometimes with good reason. Let me explain further.

1.1.1 *The Java portal disappointment*

When Java portals were first announced, they were hailed as the solution to many of the problems facing enterprises and solution architects. The web had grown up. Instead of using proprietary interfaces, everybody finally standardized on TCP/IP networking and open protocols such as HTTP, SOAP, IMAP, SMTP, and the like. Services, applications, and email operated on these open protocols, and products that once relied on proprietary protocols now opened up to the web. Those products that didn't (or whose vendors delayed doing so) were relegated to the dustbin of history. And when we had all of these siloed services speaking the same language, we needed a way to bring everything together for the end user.

Enter the Java portal. The release of the Java portal specification came with the promise of bringing services together in a single unified web desktop. Not only would it unify everything for a corporation's internal applications, but it would also be the hub of all business to business (B2B), business to consumer (B2C), business to employee (B2E), and even government to public (G2P) communication. It would be the presentation layer for the brand-new service-oriented architecture that we had just finished (or were in the process of) implementing. It could also be a platform for new applications. And it could finally bring together our static web sites and our applications, which resided on separate application servers.

Do you think too much was promised? How's that old saying go? "If it seems too good to be true, it probably is." Well, you're right. What happened? At least three issues emerged that prevented Java portals from achieving widespread acceptance.

First, it was difficult to develop solutions using a portal. The initial Portal API turned out to be something like getting to the least common denominator. Instead of providing all the features developers needed to bring all this stuff together, it defined what seemed like the absolute minimum that vendors could standardize on and left everything else up to the individual vendors. This meant developers had to spend time implementing features that should have been part of the platform in the first place. For example, the initial standard didn't include any way for portlets to communicate with each other.

Second, the portal servers were too big and complex (not to mention hideously expensive), often taking days to set up. For the developer trying to get a development environment going, it was sometimes even worse. I can remember trying to work with one of the first portals (sorry, can't tell you which one it was) and finding it impossible to get a development environment properly configured on my laptop. At the time, I was a team lead and was trying to make the install process a repeatable procedure for the rest of the developers on my team. My solution? I went to a conference, grabbed one of the presenters after his talk, and made him help me install the development environment on my machine. When he heard of my plight, he understood completely: he told me everybody was having this problem and that they had to make this process easier.

Third, other things were happening in the industry at the same time. The Web 2.0 concept was beginning to become popular, and the portlet specification left no room whatsoever for enabling a rich, client-side experience for the end user. In order to compete, portal vendors started to implement their own proprietary extensions to the portlet specification. We all know what this leads to: vendor lock-in, which is precisely what defining a standard is supposed to avoid.

At the same time Java portals were getting a bad rap, sites like Facebook and MySpace came out and pretty much implemented what portals were meant to do all along. As they became more and more popular, suddenly other sites like Amazon.com and other software like JIRA began to implement the social collaboration features of Facebook and MySpace, along with their slick, AJAX-enabled user interfaces. What powered all these new and improved web sites? What enabled them to implement such rich features for the end user so quickly? You guessed it: open source.

Open source solves a lot of the problems inherent in the old Java portal paradigm. Open source projects don't wait around for committees to decide on things; they tend to implement what the users want as fast as possible. There are no barriers to entry with open source; the development tools and the software are made available for free. Open source products also tend to be lighter weight: you don't need a large, dedicated server to start building your solution. Development goes faster, because developers don't have to learn the entire architecture to be effective. And you don't need a

huge initial investment to get started using an open source solution—you can start small (free) and then grow your application and hardware as your needs grow. Facebook is the perfect example of this: implemented using PHP (which is an open source web development platform), the site has grown organically as its user base has grown. This is what the market really wanted. And as you're about to see, Liferay Portal provides the same kind of open source platform that has allowed many organizations to do the same.

1.1.2 Liferay keeps the Java portal promises

From the beginning, Liferay Portal has been an open source project. Its whole purpose for existence was to level the playing field so that smaller organizations such as nonprofits, small businesses, and open source projects could take advantage of its platform without having to incur huge expenditures for either software or hardware. Right out of the gate, it did things differently.

An open source project doesn't have the luxury of making it difficult for developers to work on the platform. Instead, developers need to find the platform to be easy to work with, or the project will have major hurdles to community gestation. And if an open source project can't foster the birth and growth of a vibrant community, it's dead. Right away, Liferay was (and continues to be) easy for developers to use, adapting to many different development styles, and not requiring any specific tools to be installed beyond what is already in any Java developer's toolset.

This same philosophy translates to its size. Open source projects also don't have the luxury of being too big or taking up too many system resources; they may be running on new hardware or five-year-old hardware that was donated to a nonprofit that can't afford anything else. Liferay Portal is much smaller and simpler to configure than its competitors. Can you run Liferay on big hardware with a proprietary Java application server? Sure you can. Can you run it on a shared server with a small servlet container like Tomcat? Absolutely. Liferay Portal is provided as a standard .war file—only around 125 MB in size—which can be installed on any application server, or as a bundle, preinstalled in your open source application server of choice. You don't have to go through long installation routines and complex command-line incantations to get it working. If you use a bundle, installing Liferay is as easy as unzipping an archive and editing a text file to point it to your database.

And guess what? Instead of giving you by default an empty portlet container into which portlet applications can be installed, Liferay Portal comes with over 60 portlet applications included. These applications cover about all of the standard functionality you're likely to need in a web site—content management, forums, wikis, blogs, and much more—leaving you to implement only the features specific to your site. And for developers, your setup time will be measured in minutes, not hours. You also don't have to know everything about the architecture to be effective—it's easy to get started.

Open source software also has to be innovative in order to compete with its proprietary competition. Liferay Portal was the first portal to implement that slick, Web 2.0

interface, back in 2006. The first time I saw a portlet being dragged across the browser window and dropped into another spot on the page, I was blown away, because I was used to the old, proprietary solutions that hadn't implemented that yet. Because Liferay Portal was open source, it could respond to market demands faster than the other guys, using the same standards they were using. You'll continue to see that in Liferay Portal, because the open source paradigm works. What users demand gets implemented, without sacrificing adherence to standards.

As far as standards go, Liferay is also based on widely used, standard ways of doing things. It adheres to the JSR-286 portlet standard. In addition to that, it includes utilities such as Service Builder to automatically generate interfaces to databases (something not covered by the standard). Under the hood, Service Builder is just Spring and Hibernate—widely used by Java developers everywhere. You get the benefit of using the platform to get your site done more quickly while taking advantage of standards that keep your code free.

Now that I've spent so much time extolling the virtues of this magical, mystical thing known as Liferay Portal, you're probably anxious to see what this wonderful specimen I've described looks like.

1.2 Getting to know Liferay

Liferay Portal is an open source project that uses the Lesser General Public License (LGPL). This is the GPL license you know and love, with one important exception: Liferay can be linked to software that isn't open source. As long as you use Liferay's extension points for your custom code, you don't have to release your code as open source if you don't wish to. You can keep it, sell it, or do whatever you want with it; it's yours. But if you make a change to Liferay itself by modifying Liferay's source code and want to redistribute the product thereafter, then you need to contribute that change back to Liferay. You get an important exception with the LGPL: you can use Liferay as a base for your own product and either open-source the result or sell it commercially if you wish. Or, if you want to change Liferay directly, you can contribute to the open source project. It's entirely up to you.

You can download the open source version of Liferay Portal for free from Liferay's web site.

Alternatively, Liferay sells an Enterprise Edition of Liferay Portal. This is a commercially available version of the product that comes with support and a hot-patching system for bug fixes and performance improvements. There are web sites running on both versions of Liferay Portal, and both are perfectly appropriate for serving up your site.

In this section, we'll take a quick tour of some of the things you can do with Liferay to begin building a web site. You're going to play around with the interface a bit so you can get to know it better. Figure 1.2 shows the default Liferay Portal 6 user interface.

Okay, I agree; it doesn't look like much, does it? But there's an awful lot of power hidden in the humble interface that Liferay shows you by default. If you're ahead of the game and already have Liferay running, you can follow along. If not, sit back and

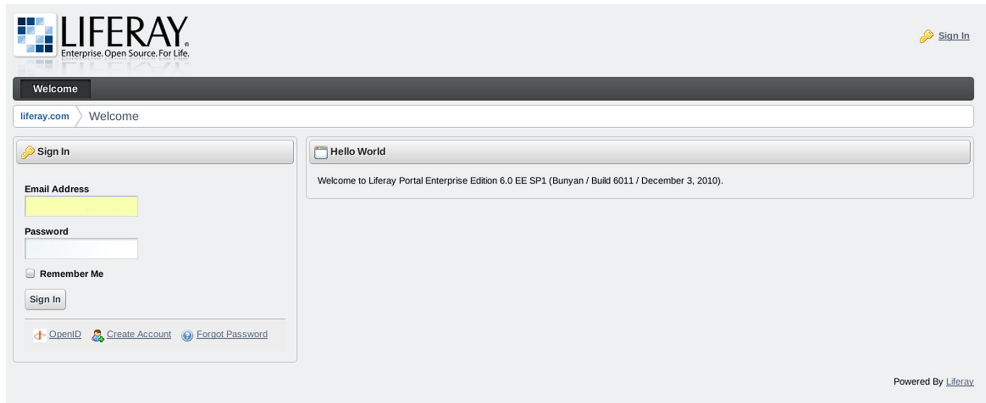


Figure 1.2 Liferay Portal 6, as it looks the first time you start it. It presents a basic interface at first, but as you'll see, you can easily jazz it up.

enjoy the ride: we'll go over how to get Liferay installed and running on your system in Chapter 2.

1.2.1 *Liferay is an application aggregator*

We've been saying that Liferay Portal isn't just a product; it's a platform. This platform runs applications, and these applications are integrated in ways that separate applications can't be, by virtue of their shared platform.

This means you can take that default Liferay page and load it with integrated applications. Liferay makes doing something like that easy. First, you have to log in as the default administrative user, whose user name is *test@liferay.com* and whose password is *test*. Doing so displays the *Dockbar* (see figure 1.3) at the top of the page, which gives you access to several other functions.

We'll come to all the things you can do with the Dockbar in a moment. For now, all we want to look at is applications, which you can access from the Add menu. Commonly used applications appear directly in the menu, but if you want to see the whole list, choose More. Doing so pops up a fully searchable, categorized view of all the

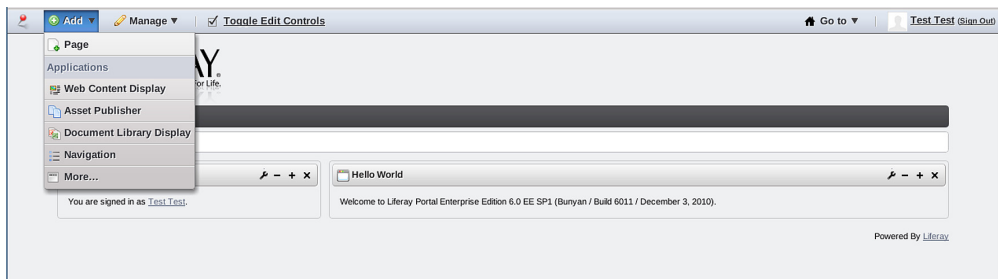


Figure 1.3 The Dockbar appears when you log in. Hovering the mouse over the Add menu in the Dockbar opens a drop-down menu. To see a full list of available applications, choose More.

applications that have been installed in your Liferay Portal by default. As an aside, by the time you're finished with this book, one of the things you'll be able to do is write your own applications, which can appear in this list.

You're going to fill this page with applications so you can see how Liferay aggregates them. You can browse the applications by opening the categories to which they're assigned. Or if you know the name of the application you're looking for, you can search for it by using the search bar at the top of the Applications window. You can add an application to a specific column by dragging the application off the Applications window and dropping it into the appropriate column, as shown in figure 1.4. Let's pick some cool applications to add to your page. To the left column, add Navigation, Activities, Dictionary, and Translator. To the right column, add Message Boards, Wiki, and Calendar. Note that in a real-world web site, you'd likely never put all of these on one page—you're doing an experiment here to see the concept.

Now you have a single page with a bunch of applications on it. These applications can perform a lot of different functions.

The Message Boards application is a complete implementation of web-based forum software. If you're planning to have discussion forums on your web site, Liferay already has them built in. And the cool thing about it is you don't have to integrate anything. They already work with Liferay's user-management and security features, as do all of Liferay's applications.

You've also added a Wiki application to the page. Again, this is a full-fledged wiki that you can use for whatever purpose suits you. As with the Message Boards application, Wiki is integrated with Liferay's user management and security. But (and this is

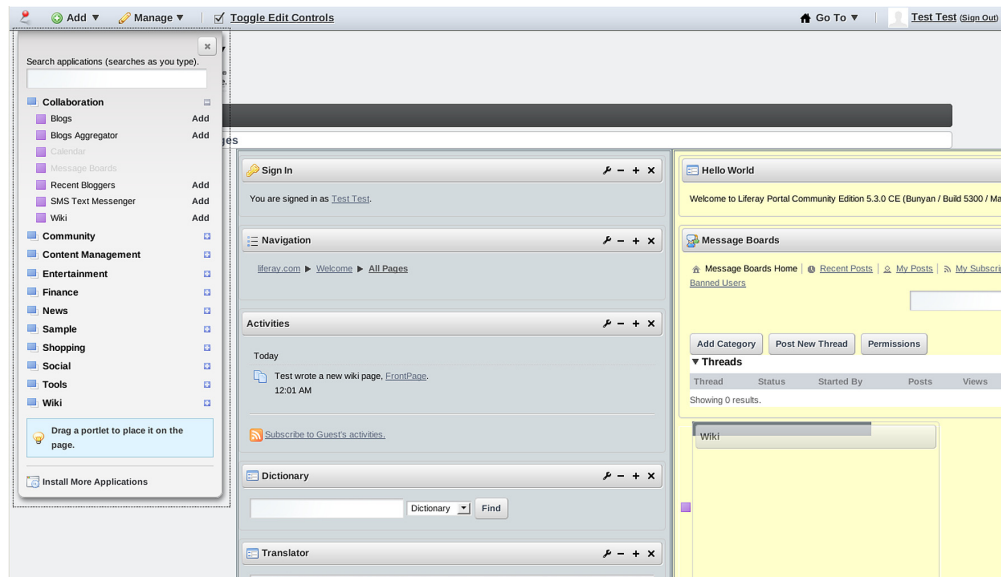


Figure 1.4 Most of the applications have been added. This screen shot was taken while dragging the Wiki application into the column on the right.

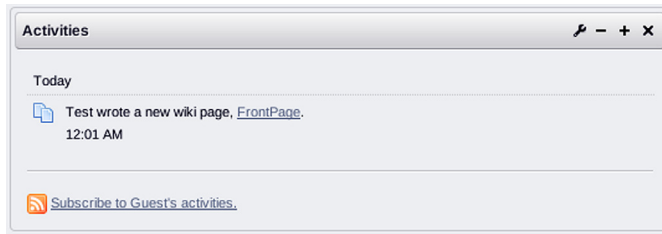


Figure 1.5 Every application in Liferay that uses the Social API can capture activities unique to that application. The Activities portlet displays those activities. Did you really create a new wiki page?

the cool part) Wiki also is integrated with Liferay’s Message Boards application, because it borrows functionality from that application to provide comment threads at the bottom of Wiki articles. Those threads use your users’ profile information (including pictures) to uniquely identify them in a consistent way throughout your site, which is yet another level of integration.

What about the Calendar application? Again, it’s totally integrated, complete with email notifications and more. And it’s a full-fledged calendar application that supports export and import of calendar data from other applications.

The other applications are smaller, and I don’t want to gloss over them, but you’re probably getting the picture at this point. Let me point out one other thing, though: the Activities application. Notice in figure 1.5 what it says.

When you added the Wiki application to your page, you created a top-level wiki page, which is by default called *FrontPage*. Because the Wiki application uses Liferay’s Social API to capture its unique activities, the Activities application can report on what you did (and even provides an RSS feed of activities). If you also use Liferay’s Social API, this opens up all kinds of possibilities for your own applications, doesn’t it? (We’ll cover this API in detail in chapter 6.)

NOTE Because Liferay is a portal, its applications are called portlets. I have been careful so far to refer to them only as applications, but for the remainder of this book, we’ll use the terms portlet and application interchangeably.

Naturally, you’d never in the real world create a page like this. Your users would throw conniption fits if they had to navigate such a thing. I just wanted to illustrate how integrated Liferay’s applications are.

In addition to providing a development platform and a slew of applications out of the box, Liferay is also a powerful content management system (CMS).

1.2.2 Liferay is a content manager

If you have lots of web content and wish to publish that content using a workflow or on a schedule, statically or dynamically, to staging or production, with templates or without, then you may want to check out Liferay’s CMS.

You can access the web-content functions from the same Applications window you’ve already seen; they’re in their own category. But the quickest way to do it is to select Web Content Display from the Add menu. It’s on the menu for convenience—if

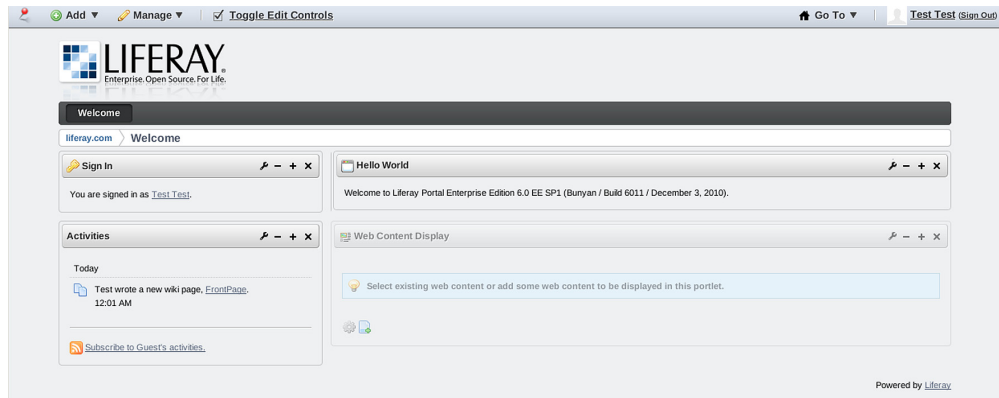


Figure 1.6 The Web Content Display portlet is added, but it has no content (yet). You'll remedy that quickly.

you're building a content-rich web site, you'll use it a lot. After it's added, you can drag it to whatever position on the page you want. Figure 1.6 shows this portlet added to the right column on the page.

The Web Content Display portlet does what its name implies: it displays web content. In order for it to do its job, you'll have to create some web content. You can do that quickly by clicking the Add Web Content icon at lower right. You're then brought to a form where you can add content (see figure 1.7).

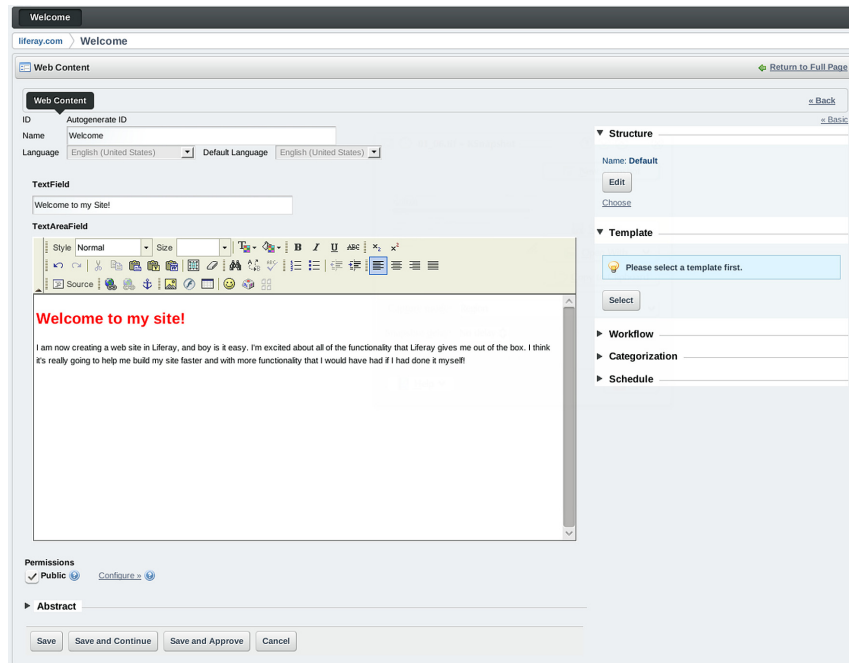


Figure 1.7 Entering content in Liferay's CMS

For now, don't worry about all the options on the right side of the screen (Structure, Template, Workflow, and so on). For basic content management, all you have to do is begin adding content. Give your piece of content a name and a description, and type some content into the editor. Notice in figure 1.7 that you can apply all sorts of formatting in the editor: fonts, tables, bullets, colors, and images.

When you've finished adding your content, notice the buttons at the bottom of the page. Although there is a workflow process you can go through, you're logged in as a portal administrator. This means you can short-circuit the workflow process by clicking the Save and Approve button. Do that, and you'll be brought back to your original portal page; the Web Content Display portlet will contain the content you just added.

You could go further with Liferay's web CMS, but suffice it to say that it's sufficiently powerful for whatever content needs you may have. For example, you can create your own structures with custom fields for your content, as well as templates to go along with your structures to display content exactly the way you want it displayed. You can stage content on a staging server and have it published on a schedule of your choosing. And you can write powerful, scripted templates in XSL, Velocity, or FreeMarker.

You've seen so far that Liferay can be a platform or a UI for your applications, and it can also manage your site's content. The last ingredient that you need, Liferay provides in spades, and that's a way for your users to find and collaborate with each other.

1.2.3 Liferay is a collaboration tool

Liferay Portal is ideal for setting up collaboration environments among workgroups. Whether you call these environments communities or virtual team rooms, Liferay can be used to help your team get their work done. It does this by providing applications that are geared specifically toward document sharing and communicating with one another.

One of the portlets you can add to a page in Liferay is Document Library. This application provides a facility for sharing documents with your entire team. It keeps a complete version history of all your documents and is integrated with Liferay's permissions system. This integration allows you to grant access to shared documents or prevent some of your users from accessing sensitive documents. And if your users need an easier way to access the documents than the web interface provides, Document Library supports WebDAV, allowing documents to be uploaded and downloaded through their operating system's native interface. Figure 1.8 shows both of these interfaces.

Documents are one thing, but what about communication? Liferay's portlets allow for communication in context, so your users can keep all the relevant information in the right place. Document Library allows your users to create discussion threads next to the documents they need to talk about. Wiki does the same thing. And applications are provided for both chat and email, so that currently logged-on users can communicate in real time no matter what their physical distance is from one another.

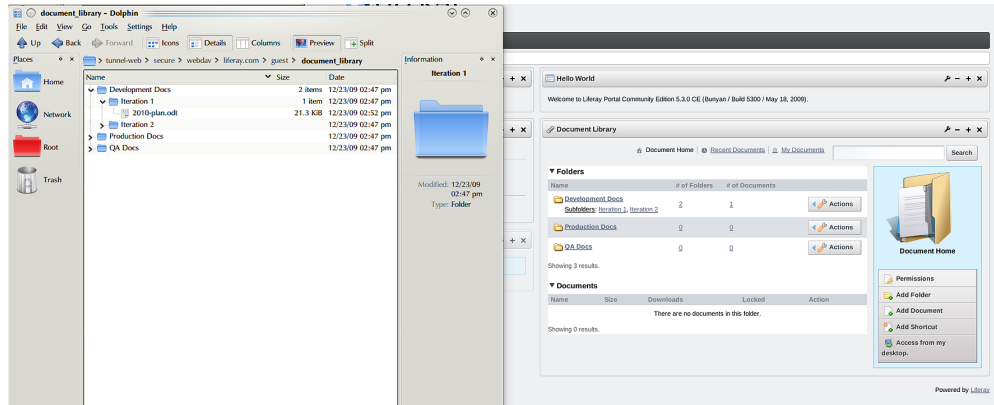


Figure 1.8 Accessing the same folders in Document Library in the operating system via WebDAV or using the browser interface

Need a group calendar? The Calendar portlet can be used for either individuals or for groups. Additionally, users can have their own individual blogs on their own pages, which are then aggregated to the community page using the Blogs Aggregator. This enables you to display a “blog of blogs,” allowing your team to stay updated on what everyone is doing. Combining this with Activities makes for a consistent, rolling list of what the team is up to.

All the functionality I’ve mentioned so far is what is built in to Liferay (and there’s more we haven’t touched on). But Liferay is extensible, too.

1.2.4 Liferay is anything you want it to be and any way you want it to look

Liferay offers a level of customization that is unparalleled, because you can modify *anything* in Liferay, from simple functionality changes all the way to making your own product out of it.

This book will systematically show you how to write your own portlets so that your applications can be added seamlessly to your Liferay-powered web site’s pages in a way that is indistinguishable from the built-in portlets. You’ll also learn how to customize Liferay’s layout templates so that your page layouts can be what you want them to be. You’ll learn about hooks, which let you customize Liferay by substituting your own classes and JSPs in the place of Liferay’s. And finally, you’ll learn about Ext plugins, which let you override anything in Liferay with functionality of your own.

No discussion of customizing Liferay would be complete without covering themes. Using themes, you can transform Liferay’s look and feel to make it look any way you want it to (see figure 1.9). In short, Liferay can be anything you want it to be, and it can look any way you want it to look. This gives you the power and flexibility you need to build your own custom site, with the functionality you need to get it done in a timely fashion.

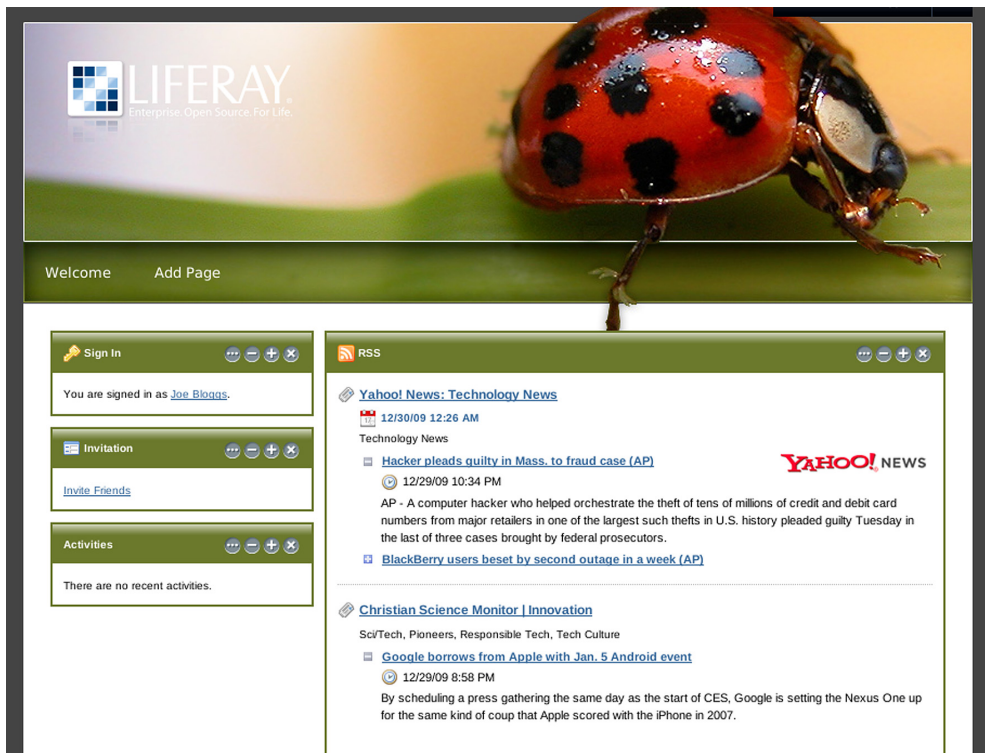


Figure 1.9 Liferay with a custom theme applied. This is just one of many themes in Liferay's community repository.

You can modify Liferay's look and feel using skills you already have. Themes are nothing more than custom HTML and CSS applied to the page. You can design your site just as you would if you were writing the whole thing from scratch—except that you have less work to do, because of Liferay's built-in functionality and rich development platform.

Liferay also comes connected to two repositories of ready-made plugins that extend Liferay's functionality. One of these is a Liferay-provided repository, and the other is a community repository. Figure 1.9 is an example of a theme provided by Liferay's community through the community repository. Liferay's repositories make it easy to both distribute and to install new software that runs on Liferay's platform, as you can see in figure 1.10.

As you can see, a lot of functionality is built in to Liferay Portal, and it's also extremely easy to add functionality to Liferay Portal. You can rest assured that the software you create on Liferay's platform will be easily installed by your users. Let's take a step back now so you can see what you have accomplished with just a few clicks.

The screenshot shows the Liferay Plugin Installer interface. At the top, there's a 'Plugin Installer' header with tabs for 'Browse Repository', 'Upload File', 'Download File', and 'Configuration'. Below this, there are tabs for 'Portlet Plugins', 'Theme Plugins', 'Layout Template Plugins', 'Hook Plugins', and 'Web Plugins'. A search bar is present with fields for 'Keywords', 'Tag' (set to 'All'), 'Repository' (set to 'All'), and 'Install Status' (set to 'Not Installed or Out of Date'). A 'Search' button is below the search fields.

Below the search area, it says 'Showing 21 - 40 of 111 results.' and 'Items per Page 20' and 'Page 2 of 6'. Navigation links for 'First', 'Previous', 'Next', and 'Last' are provided.

Portlet Plugin	Trusted	Tags	Installed Version	Available Version	Modified Date
Add New User Wizard 5.2.2.1 ID: robisoft/03/5.2.2.1/war A plugin portlet using Spring Wizard	No	sample, spring, wizard	-	5.2.2.1	3/22/09 1:24 AM
WOL 5.2.2.1 ID: liferay/world-of-liferay/5.2.2.1/war This is the World of Liferay portlet that integrates social networking features.	Yes	wol	-	5.2.2.1	2/27/09 5:49 PM
Web Form 5.2.2.1 ID: liferay/web-form/5.2.2.1/war This is the Web Form Portlet.	Yes	web form	-	5.2.2.1	2/27/09 5:49 PM
Mail 5.2.2.1 ID: liferay/mail-portlet/5.2.2.1/war This is the Liferay Mail portlet.	Yes		-	5.2.2.1	2/27/09 5:46 PM
Google Maps 5.2.2.1 ID: liferay/google-maps-portlet/5.2.2.1/war This portlet allows easy integration with Google Maps.	Yes	google	-	5.2.2.1	2/27/09 5:45 PM

Figure 1.10 Browsing Liferay's plugin repository from within the control panel. Installing any plugin is a simple matter of clicking the plugin and then clicking the Install button that appears with the full description of the plugin.

1.2.5 What has this little exercise accomplished?

I hope you see the power that Liferay gives you. In about 10 minutes—and without any additional software—you've created a web site that contains web content, forums, and a wiki; displays users' activities; shares documents; and has a custom look and feel. You didn't have to use separate applications—instead, all that functionality (and more) is included in Liferay. And because you didn't have to use separate applications to implement what you wanted, you didn't have to spend any time integrating those applications. Users get the experience of being able to sign in to your site once and then navigate to the content they have access to, and you don't have to do *anything* to make that work.

Pretty awesome, isn't it?

Obviously, this only scratches the surface. You need ways of organizing and granting permissions to all those users you're going to have. To do this, you'll need to understand the reinforcement beams, foundation blocks, and structures Liferay gives you to support that portal full of users.

1.3 **How Liferay structures a portal**

Every portal is different in the way it handles users, security, and pages. Because these aspects of a portal aren't covered by the JSR-286 standard, every portal vendor has implemented these concepts differently. If you're going to start developing on Liferay's platform, you'll need to understand how a Liferay portal is configured and organized. Don't worry: it's not all that complicated, although it may look that way at first. When you start using the system, you'll get the hang of it quickly.

Liferay 6.0 and Liferay 6.1

Liferay 6.0 has organizations and communities; Liferay 6.1 has organizations and sites. In the following discussion, I mention organizations and communities quite a bit. If you're on Liferay 6.1, everything I say about communities applies to sites. Organizations don't have pages in 6.1, but sites can be linked to organizations. Sites linked like this in 6.1 are essentially the same as organizations in 6.0. For this reason, the terminology is interchangeable. All 6.1 does is give you a little more flexibility.

In this section, you'll see how to collect users into various categories and what those categories can do for you. You'll also see how Liferay makes it easy to create web pages in your site and how to place content on them.

1.3.1 **The high-level view**

At its most basic level, a Liferay server consists of one or more portals. Portals have users, and these users can be categorized into various collections. Some of these collections can also have web pages that compose a portion of your site.

You can define many portals per portal server, and each portal has its own set of users and user collections. Figure 1.11 displays this graphically. As shown in the figure, each portal has users, and those users can be organized into several different types of

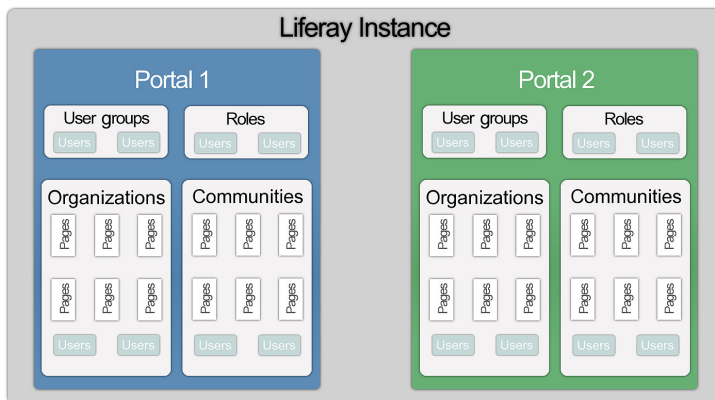


Figure 1.11 A single Liferay Portal installation can host many different portals, all with separate users and content.

collections: roles, organizations, communities, user groups, and any combination of those collections within that portal. Table 1.1 lists the collection types Liferay offers.

Table 1.1 Liferay collection types

Collection type	Description
Role	Collects users by their function. Permissions in the portal can be attached to roles.
Organization	Collects users by their position in a hierarchy. Organizations can be nested in a tree structure. You use organizations to represent things like a company's organizational chart.
Community	Collects users who have a common interest. Communities are single entities and can't be grouped hierarchically. By default, users can join and leave communities whenever they want, although you can administratively change this so users are assigned to communities (or invited) by community administrators.
User group	Collects users for purposes that cut across the portal. User groups are defined by portal administrators.

Roles are inherently linked to permissions. You use a role to collect users who have the same permissions. A good example is a Wiki Administrator role: this role contains users who have permission to administer wikis.

Organizations are hierarchical collections of users. Users can be members of one or many of them, up and down the hierarchy. Membership in organizations gives users access to the pages of that organization. If you picture a hierarchical structure that represents a company called Inkwell, some users might belong to Inkwell, Sales Department, Mid-Atlantic Region. This not only denotes the employees' position in the company, but also gives employees access to the content they need to do their jobs.

Communities are ad hoc collections of users. Users can join and leave communities, and membership in communities gives them access to the pages in the communities of which they're members. You may have a community called Photography: users of your site can join this community to share pictures.

User groups are defined by portal administrators. They're used to collect users for purposes that tend to cut across the portal. For example, you may want to grant some users the ability to create a blog on your site. To do so, you create a user group called Bloggers and create a page template for them that contains Liferay's Blog portlet. Regardless of these users' membership in other collections (as part of a hierarchy of organizations or as having joined several communities), user groups provide a separate way of granting specific access to functions that don't depend on membership in other collections or on specific portal permissions.

That's the high-level view of a Liferay portal structure. Although this describes a powerful system for building your web site, it's only the basics. Let's move on to the next level.

1.3.2 Adding content to a collection with pages

Three types of collections can have not only users, but also pages. Pages are, of course, clickable, navigable web pages. Organizations and communities can have any number of pages defined within them. Pages are organized into *layout sets*, which come in two types: public and private. Each organization or community can have public pages, allowing them to configure a public web site which can be used by members and non-members of the organization or community. And they can also have private pages, which are only accessible by the members of the organization or community. You can begin to see how you can build out your site and separate functionality by whoever is accessing the site.

User groups don't have pages per se but rather can have *page templates*. These are configured by portal administrators and become useful for users' personal communities. By default, each user gets a personal community, which itself has public and private layouts. This is a personal web site that the end user can configure (or that can be fairly static—or not exist at all, depending on how you've set up the portal). Portal administrators can create page templates for user groups. These page templates can be populated with the portlets that administrators want users to have. When users are then placed into the user group, any page templates are copied into those users' personal communities. If, for example, you want certain users to have a blog, you can create a Blog page with the Blogs portlet on it in a user group called Bloggers. Users you add to this user group will have this page copied automatically to their personal communities, and they can begin blogging immediately.

If you haven't already figured it out, a roles collection has no pages because roles are used solely to aggregate permissions. For example, you can define a role that has permission to view certain pages. This is how roles work together with organizations, communities, and user groups.

Liferay Portal also has the idea of *scope*, the topic of the next discussion.

1.3.3 Configuring a portlet's scope

Scope allows some of the concepts mentioned previously to be refined. One user collection that is refined by scope is roles. As stated earlier, roles are the only collection to which permissions can be attached. For example, you can create a role called Wiki Administrator. This role has permissions to the Wiki portlet, allowing users in this role to create new wikis and add, edit, delete, and move pages. This role can be created under one of two scopes:

- Portal role
- Community/Organization role

If you create this role as a portal role, then any members of the role have the defined permissions across the entire portal, in any community or organization. This allows users in this role to administer wikis in whatever communities or organizations they have access to. But you can define the role in another way, by scoping it only by community or organization. If the role is defined this way, users have the role's permission

in only the community or organization in which that role is assigned. As you can see, scope is important when it comes to how permissions are defined.

Scope also comes into play with regard to certain Liferay built-in portlets. If you go back up to the Dockbar and choose Add > More, you'll see that the portlets are marked with different icons. These icons tell you something about the portlets with which they're associated. But before we go over what they mean, let's look at some portal terminology.

Sometimes I feel like Dr. Seuss when I begin to discuss this topic:

*If a portlet in a portal on a page in an org,
Has a data-set saved as its own data-store,
And the data would be different for other users' chores,
We call that a non-instanceable portlet!*

And...

*When a portlet in a portal saves its data on the disk,
And the user hits the data based on membership in this,
If the portlet is configured to have its own instances,
We call that an instanceable portlet!*

All of this has to do with scope.

NON-INSTANCEABLE PORTLETS

Let's stick with the wiki example. If you place a Wiki portlet on a page, based on what I've described, where is that page? Yes, it's in a community or organization. That wiki now belongs to that community or organization.

You can't place another Wiki portlet on the same page, because that portlet is what Liferay calls *non-instanceable*. The portlet's set of data belongs to that community or organization. This is in contrast to *instanceable* portlets, where the set of data belongs to the user. But we'll get to those in a minute.

You can place another Wiki portlet on a different page in that community or organization, but that Wiki portlet will display the same data as the first one (that is, unless you place it in another *scope*, which we'll also get to in a minute). Why is that? Because the portlet's data set has already been created, by placing it on a page in that community or organization. In other words, non-instanceable portlets by default all point to the same instance of data (see figure 1.12).

For non-instanceable portlets, the data a portlet contains belongs to the community or organization.

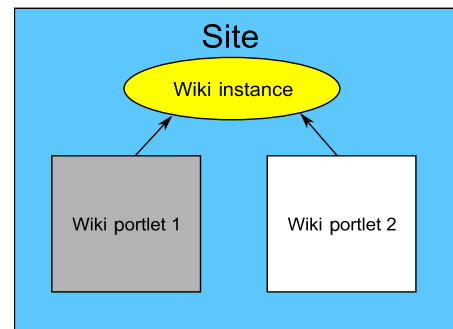


Figure 1.12 A non-instanceable, scopeable portlet has its data scoped by the community or organization to which it belongs. No matter how many times you add it to the community or organization, it points to the same data.

This may seem like a limitation, but it's a powerful benefit. You can have lots of wikis on your site, and they can all be kept completely separate from the others. For example, say you're building a portal for do-it-yourselfers. Your audience likes to build stuff, but the stuff they want to build differs wildly. Your site has communities for many different topics, from home renovation all the way to building model rockets and platforms for model railroads. To serve the needs of these users, you may want to give them a wiki so they can add helpful tips and articles based on their experiences. But the model-rocket group and a home-improvement plumbing group won't have much in common (or maybe they will, depending on the size of the rocket—but that's not what we're focusing on right now). You can easily give them separate wikis in their own communities with Liferay. And, of course, because of Liferay's powerful way of collecting users into communities, all your users will be members of your portal, but not necessarily of the same communities. They will have the freedom to navigate to the content that is most appropriate for them.

INSTANCEABLE PORTLETS

Other portlets in Liferay are *instanceable*. This means you can place as many of them as you like on the same pages in any community or organization, and by default they all have their own configurations. For example, the RSS portlet is designed to show RSS feeds. You can add as many RSS portlets as you want to any page and configure each portlet to display different feeds, because this portlet is instanceable. The Web Content Display portlet is the same way: you can place as many Web Content Display portlets on a page as you wish, and each portlet can display a different piece of web content. When you choose portlets from Liferay's Add > More window, the interface indicates which portlets are instanceable and which are non-instanceable, as shown in figure 1.13.

You can tell which portlet is which in the user interface by looking at the icons in the Add > More window. If there's a green icon with two windows, the portlet is instanceable. If there's a purple icon with one window, the portlet is non-instanceable.

PAGE SCOPES

Sometimes, Liferay's default scopes need to be enhanced with more flexibility. I'm going to backtrack a bit on what I said earlier: if you *really* need to have two non-instanceable portlets with different data sets in your

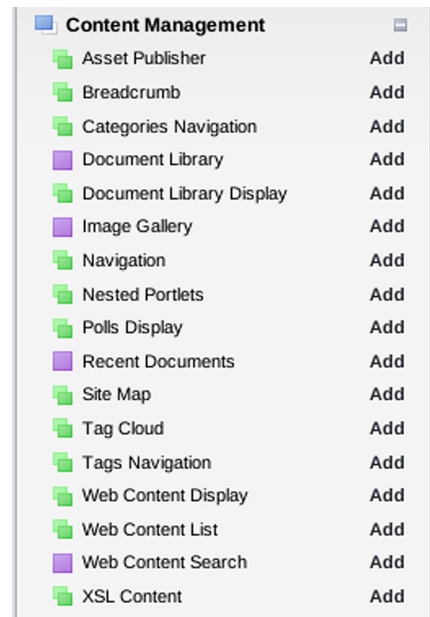


Figure 1.13 Instanceable and non-instanceable portlets in Liferay's Add window. Liferay's UI clearly shows you which portlets can be added multiple times to the same page (instanceable) and which can't (non-instanceable).



Figure 1.14 Changing the scope in the Wiki portlet

community, you can do that. It's just not configured to be that way by default (and this functionality wasn't available in older versions of the product). This has to be configured on a per-portlet basis.

Still using the Wiki portlet as an example, if you click the configuration icon in the portlet window (which looks like a wrench in the default theme), a menu pops open. Click Configuration in this menu, and all the configuration options for the portlet are displayed. One of the tabs in this window is called Scope (see figure 1.14).

Here, you can change the scope from the default to the current page. This lets you tie the portlet data to the page *in* the community or organization instead of the community or organization globally. What this means is that you can add another page to this community or organization, place *another* Wiki portlet on that page, and set the portlet's scope to *page*, and it will have a different data set than the one on the other page. You can't add multiple wikis to the same page, but you can have multiple non-instanceable portlets per community or organization, all with different data, provided the portlet supports page scopes.

I don't want to delve too much into these concepts at this stage. You'll be taking advantage of scope soon enough in your code. For now, let it all sink in, and we'll turn to something more concrete: how to navigate around Liferay.

1.4 Getting around in Liferay

Liferay's user interface has a philosophy behind it: get out of the user's way. For that reason, it hides a lot of power behind what looks like a simple interface. One of the main UI elements is the Dockbar.

You've already been introduced to some of the functionality of the Dockbar, so let's see what other functions it provides. Figure 1.15 shows the Dockbar in full.

Let's take each element in order from left to right.

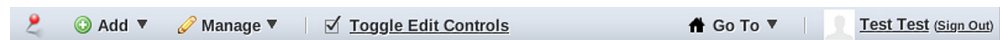


Figure 1.15 Liferay's Dockbar, which appears at the top of every page when a user is logged in

1.4.1 Pin icon

At the far left is a pin icon, which does what you would expect it to do: it pins the Dockbar to the screen so that no matter how far down you scroll, it stays at the top of the screen. This can be helpful if you're working with long pages and need to use the Dockbar's functionality to add portlets to the bottom of the screen. This is a toggle switch, so you can unpin the Dockbar by clicking the icon again.

Next in the Dockbar is the Add menu.

1.4.2 Add menu

You've already seen most of the functionality of this menu for adding applications to the page. It can add pages, too. If you choose Add > Page, a new page is added next to the page you're on, and a field appears, allowing you to name the page. There's a much more powerful page-administration screen, but this function allows you to quickly add pages to your web site as you're working on it.

The next item in the Dockbar is the Manage menu.

1.4.3 Manage menu

Use the Manage menu to manage pages, page layouts, and more. This is where you get access to the interface that lets you group pages in the order you wish—as well as nest them into subpage levels. You can also apply themes to entire layout sets or to single pages. The Manage Pages screen is shown in figure 1.16.

Perhaps the most important item in the Manage menu, however, is the Control Panel.

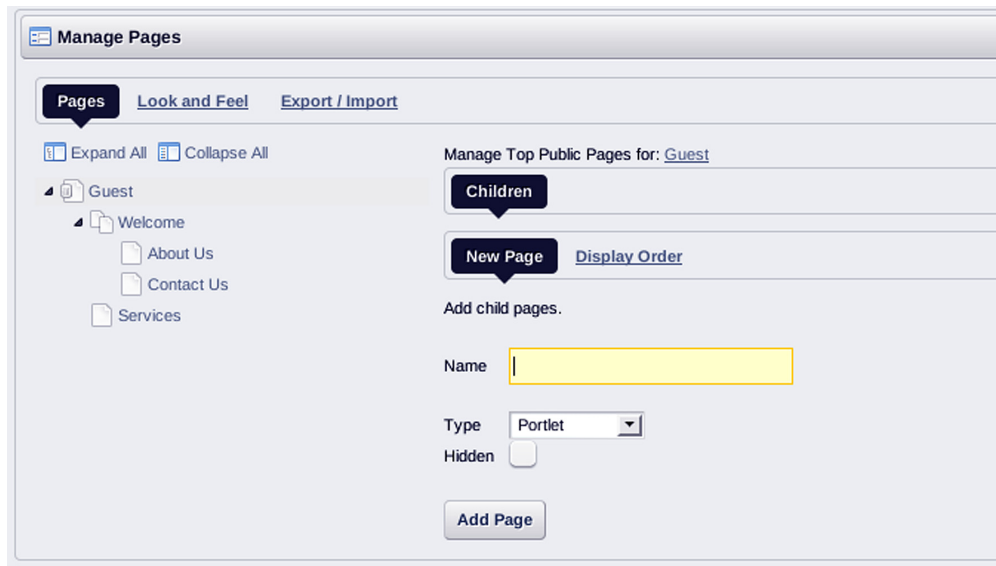


Figure 1.16 The Manage Pages screen allows you to nest pages, change the display order by dragging and dropping pages, change themes, and more.

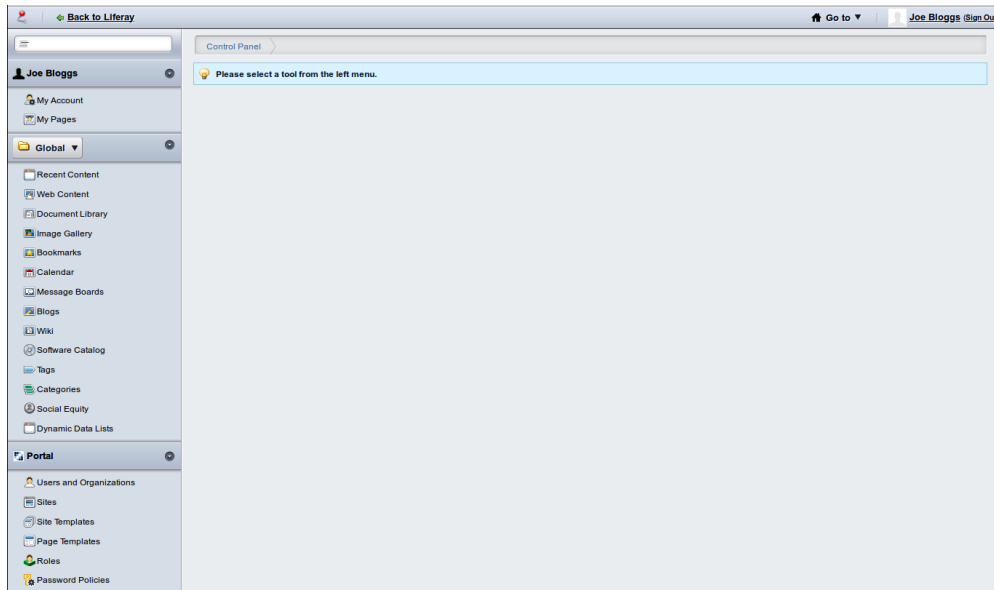


Figure 1.17 Liferay's Control Panel. It's divided into four sections: a section for the current user, a Content section, a Portal section, and a Server section, which isn't visible in the figure.

Liferay's Control Panel is the central location where just about everything can be administered. If you'll be administering a Liferay portal, you'll spend most of your time here. The Control Panel is easy to navigate. On the left side is a list of headings with functions under them. The headings are in alphabetical order, but the functions are in a logical order. Figure 1.17 shows the Control Panel, which purposefully uses a different theme from the default pages so you can instantly tell where you are.

The Control Panel's sections are as follows:

- *User Name*—The first heading is named for the logged-in user (Joe Bloggs in figure 1.17) and is used to manage the user's personal space. Here, you can change your account information and manage your own personal pages.
- *Content*—The Content section contains links to all of Liferay's content-management functions. You can maintain web content, documents, images, bookmarks, and a calendar; administer a message board; configure a wiki; and more. These links are scoped for the particular community from which you navigated to the Control Panel, but this can be changed using the select box.
- *Portal*—The Portal section allows portal administrators to set up and maintain the portal. This is where you can add and edit users, organizations, communities, and roles as well as configure the settings of the portal.
- *Server*—The Server section contains administrative functions for configuring portal instances, plugins, and more.

Next, we'll move to the right on the Dockbar to look at a toggle that becomes very important when you're administering the portal.

1.4.4 Toggle Edit Controls

The next function in the Dockbar isn't a menu; it's a toggle for the edit controls on the portlets. As an administrator, you get to see some icons in the title bars of the portlets on a page. These correspond roughly to the icons you might see in your operating system. There's an icon for closing a portlet, for minimizing it, and for the configuration menu you've already seen (you used this to change the scope of the Wiki portlet). If you're composing a page and would like to see something that more closely resembles what your users will see, you can use the Toggle Edit Controls link to turn off these controls.

Next, toward the end of the Dockbar, is the Go To menu.

1.4.5 Go To menu

Use the Go To menu (shown in figure 1.18) to navigate to the various community and organization pages to which you have access. Each community or organization name appears, along with its public and private layout sets, if they have them.

The final link in the Dockbar takes you to your user account information in the Control Panel.

1.4.6 User Account

The User Account menu item opens a page where you can change your name and email address, upload a profile picture, and maintain all information about you. You can also sign out of the portal from here.

As you can see, Liferay packs a lot of power in a deceptively simple user interface. The intent of this small tour was to give you an idea of where you can go in Liferay and how to get there as you begin to build your site.

Even though we've now touched on several of the constructs that provide you with the building blocks you'll use to build a web site in Liferay, it's sometimes difficult to begin imagining how your site could be built using these building blocks, because many of the concepts are new and unique to Liferay. So, let's spend a little time figuring out how you can imagine your site running in Liferay Portal.

1.5 Imagining your site in Liferay

Every successful web site does something unique or does something in a way that is better than anyone else has done it before. Although Liferay has tons of functionality out of the box, much of that functionality is a default implementation of features that are under the hood. What do I mean by that? Let me answer by giving you some examples.

Liferay Portal has a collaboration API that contains features allowing users to post discussions, rate items, or tag content. This API has been used to provide everything

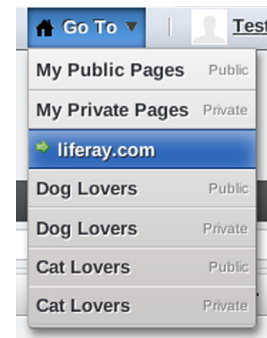


Figure 1.18 The Go To menu displaying public and private layouts for three communities: the default Guest community, Dog Lovers, and Cat Lovers. Notice that the default community only has public pages, so only one link appears.

from the Message Boards portlet, to tagging wiki articles, to rating shopping cart items. This book will introduce you to these APIs so you can consider what kinds of applications you can build with these powerful features.

That, of course, is not the only API we'll cover. You'll also see Liferay's Social API, which gives you the ability to make your applications—indeed, even your entire web site—social. Your users can connect with each other and share content and activities, and even share content and applications on other social networks. Again, the question remains: what will you do when given the power to build such applications?

The point is that when you're finished reading this book, you'll have the ability to make Liferay sing to your tune. And because there's so much power in the Liferay platform, you'll get a head start on building your site because the functionality you need is already built into the platform—all you need to do is use it.

We'll go from the ground up in familiarizing you with Liferay development throughout the course of this book. For now, let's use the information you already have to begin imagining your site from within the Liferay constructs described in the preceding sections. Then later, as you discover the full power of Liferay's development platform, you'll see how easy it is to use Liferay as the foundation of your web site, and you can plan how to integrate the features of your applications with the power of the platform.

Portal design is best done by breaking up your site into small chunks and then designing each chunk individually. That way, you don't get overwhelmed by the largeness of your task, and before you know it, breaking it into smaller chunks has enabled you to design the entire site!

In this section, we'll walk through a design process that is based on a set of forms that I've used with success to design many portals.

TIP For your convenience, you can find the portal design forms as a download along with the book's source code. Print them, and then fill them out as you work through this section.

We'll break the design process into three main portal chunks:

- User groupings
- Organizations and communities
- Content

1.5.1 Asking the right questions

The first thing you need to do is figure out how you can divide all your ideas into neat, organized chunks that you can then focus on in more detail. Ask yourself the following questions:

- Will users be given freedom to sign up on the site?
- Will your user groupings be ad hoc, static, or both? (If your user groupings will be ad hoc, you know you'll be creating communities for your users to join and leave.)

- Will some regular users have access to things others won't? (If so, you know you'll be using roles.)
- Will you be delegating administrative tasks to some users? (If so, you may have community or organization administrators.)

When you've answered these questions, go ahead and brainstorm the groupings or collections of users you may have.

1.5.2 Defining and categorizing collections

Don't worry about trying to define user groups, communities, organizations, or roles. Start figuring out some groupings. Some examples are anonymous visitors (potential customers), customers, community members, and specific groupings based on your web site. For example, if you're building a web site for do-it-yourselfers, you may come up with categories such as carpentry, plumbing, model rocketry, or even old computers.

At this point, you should have a good list of groupings. Now combine that list with your answers to the previous questions. Will any of the groups require pages? If so, you know which ones are communities or organizations. Are the groupings associated in any way? If so, how? You're beginning to identify a possible organization hierarchy.

Do some groupings cut across the entire portal (such as a bloggers group)? If so, that's a likely candidate for a user group, and you can begin thinking about whether these users should have page templates defined for them. Or it may be a good candidate for a community, if the grouping should have its own set of pages. When you've categorized your collections of users by organizations, communities, user groups, and roles, you can begin designing your content.

1.5.3 Designing content

Pages can be part of organizations or communities. By default, each can have public pages that everyone can see, as well as private pages that only authenticated members of the organization or community can see. Take each organization and community you've identified and determine the page hierarchy that will exist for it. This may even help you to further define your roles and user groups.

When you're finished with this process, you should have a nice, high-level design for your web site. You may have something very simple, like Liferay's default: one community called Guest for everyone to use. Or you may have something more complex. The point is, it's a start. From here, you can delve into the custom applications you need to write to make your site unique, as well as the customizations to Liferay that you need to make to satisfy all your requirements. That's what the rest of the book is about.

1.6 Summary

Liferay Portal is an ideal choice for building your web site. Using the unique constructs the platform gives you, you can design a site that can handle any situation you can throw at it. Liferay Portal also offers you an unbeatable platform for building web applications, as well as a ton of applications that are already implemented, in order to help jump-start the creation of your site.

In addition, Liferay Portal frees you from the limitations of the old Java portal standard. As an open source project, it enables you to be as lightweight or as heavyweight as you want to be. And because it provides a multitude of tools and utilities for increasing developer productivity, you can get your site done more quickly.

Liferay gives you a powerful paradigm for organizing users and getting them access to the content they want to see. You can use communities, organizations, roles, and user groups to make sure the right content gets to the right people and that restricted content is protected so only the proper users can view it.

Because Liferay is so easy to use, you can create complex web sites quickly. Because all the common applications you need to run a web site are included, it's a simple matter to pick the applications you need and drop them onto your pages. Because no further work is needed to integrate these applications, your time is freed to focus on the applications you need to build that are unique to your web site.

As you move further into this book, you'll learn how to customize Liferay to make it look the way you want it to look, act the way you want it to act, and host the applications you design and write. This will be an interesting journey, and I'm sure you'll find it as rewarding as I have. I hope you'll come along and take the red pill with me—it's going to be an exciting ride.

In the next chapter, you'll install Liferay Portal 6, unpack and configure the Plugins SDK, and dive into creating your first portlet application.

The Official Guide to Liferay Portal Development

Liferay IN ACTION

Richard Sezov, Jr.

Liferay in Action is the official guide to building Liferay portal applications using Java and JavaScript. If you've never used Liferay before, don't worry. This book starts with the basics: setting up your development environment and creating a working portal. Then, it builds on that foundation to help you discover social features, tagging, ratings, and more. You'll also explore the Portlet 2.0 API, and learn to create custom themes and reusable templates.

Experienced developers will learn how to use new Liferay APIs to build social and collaborative sites, use the message bus and workflow, implement indexing and search, and more. This book was developed in close collaboration with Liferay engineers, so it answers the right questions, and answers them in depth.

What's Inside

- Complete coverage of Liferay Portal 6
- Covers both the commercial and open source versions
- Custom portlet development using the Portlet 2.0 spec
- Liferay's social network API
- Add functionality with hooks and Ext plugins

No experience with Liferay or the Portlets API is required, but basic knowledge of Java and web technology is assumed.

Rich Sezov is Liferay's Knowledge Manager and is the author of the *Liferay Portal Administrator's Guide*. He leads Liferay's documentation and training materials team.

For access to the book's forum and a free ebook for owners of this book, go to manning.com/LiferayinAction

“Flat-out the best guide for Liferay 6.0 and the upcoming 6.1 release.”

—From the Foreword by
Brian Kim, Liferay COO

•
“Excellent in-depth treatise on the most popular CMS on the planet.”

—Sumit Pal, LeapFrogRx Inc.

•
“Harness the full power of this juggernaut technology.”

—Tariq Ahmed
Amcom Technology

•
“A great companion to Liferay's Admin Guide.”

—John J. Ryan III
Princigratation LLC

•
“Expertly written, thorough coverage.”

—John S. Griffin, Coauthor of
Hibernate Search in Action



ISBN 13: 978-1-935182-82-5
ISBN 10: 1-935182-82-X



9 781935 182825