Portofino - A Java Web Application Framework

Author: Carlo Cassinari carlo [at] celeweb.eu

Nowadays web applications are required both to be innovative and to interact with legacy systems. As usual, productivity and solid architectures are essential requirements.

Portofino allows you to develop web applications from existing databases, by connecting to them and automatically creating a series of CRUD pages. Then it allows you to improve the application by creating pages like in a CMS, using enterprise components defined out of the box, or developing new functionalities.

Web Site: http://www.manydesigns.com/en/portofino
Download: http://sourceforge.net/projects/portofino/
Version Tested: 4.0.10
License & Pricing: LGPL (GNU Lesser General Public License), Free
Support: Mailing list http://groups.google.com/group/manydesigns-portofino or commercial support http://www.manydesigns.com/en/portofino/support

Figure 1. An application developed with Portofino

Portofino is an open source web application framework written in Java. Its main purpose is to create a complete web application by connecting to an existing database. You use a web wizard, set the database connection, specify a few other options and the framework creates a basic application structure with CRUD pages on the tables. You, as a developer, will avoid wasting time to create pages for the common data access pages (e.g. search, creation and editing pages). But you can also create new tables, and, on the Portofino side, you just need to define a query to have new pages built.

This is just a starting point. Portofino can be used as a powerful CMS. The pages are organized in a hierarchical tree, so you add, delete, move or copy pages anywhere in the tree. Pages are of different types and can contain a single content or many contents visually arranged on the page.

You can make your website good looking and functional at the same time.

The operations so far introduced are all done via web in a RAD-like fashion.

Another purpose of Portofino is to increase the productivity and ease of development (typical of other languages such as PHP or Ruby), with the added benefit of using enterprise level libraries that are de facto standards. It uses Hibernate for the persistence [1], Stripes as the MVC framework [2], Apache Shiro for authentication and authorization [3]. Each page has an associated action written in Groovy that allows easy customization on a live system without the need to compile and redeploy.

In the next sections we will see how to start to develop a typical project in Portofino.

Connecting to an existing database

Download Portofino from SourceForge: http://sourceforge.net/projects/portofino. The package contains an Apache Tomcat with a Portofino webapp. Launch Tomcat and see Portofino running at http://localhost:8080. Portofino does not need a database to start and to be used as a CMS. It allows you to create pages with a WYSIWYG html editor and to upload resources like images or documents.

If you have an existing database (or more than one), then you can use the wizard and let Portofino create pages automatically. Launch the wizard procedure from the home page. Connect via JDBC and follow the four steps of the wizard.

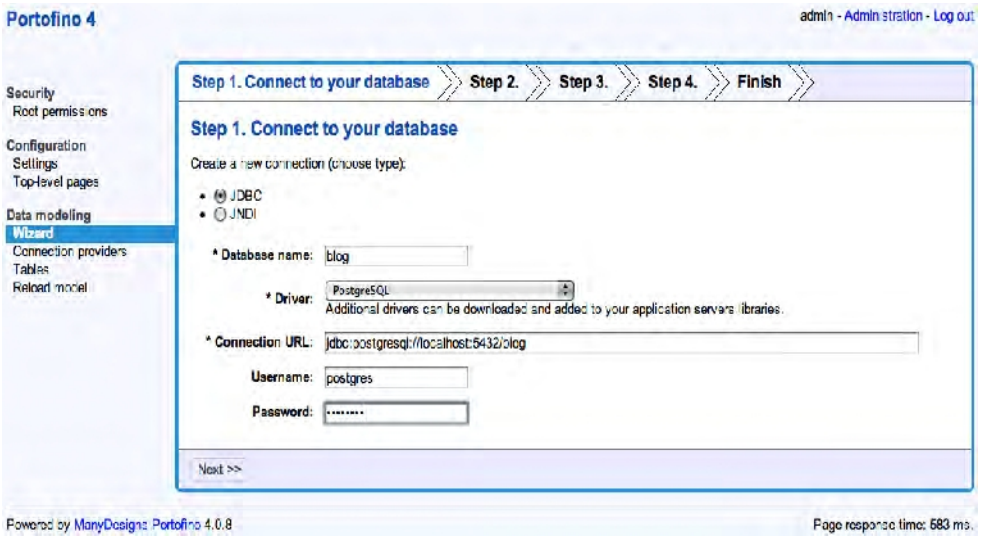

Figure 2. Connecting to an existing database

The wizard will ask you to select the schema and the tables of your database and allows you to select the users and groups tables, if you have them, to be used for authentication and authorization. At the end Portofino creates a series of CRUD pages, one for each table in the database.

These pages allow you to search, create, read, update and delete data from the associated table (defining a Hibernate query). Fields are searchable, results are paginated, columns are sortable, there are bulk edit and delete, autocomplete fields, fields validation. All these features are out of the box and do not require writing a single line of code. Furthermore, you can customize what fields to view, what will be searchable and the sorting. You can also remap field types instead of

the default taken from the database (e.g. an integer with values 0-1 can be mapped to a boolean). Productivity increases avoiding any unnecessary wastage of time in the development of common operations.

Analyzing foreign keys, Portofino organizes the pages hierarchically. Subordinate records will be positioned under their master record (e.g. “Customers” will be under “Companies”, or “Comments” will be under “Blog Posts” like in the following picture).

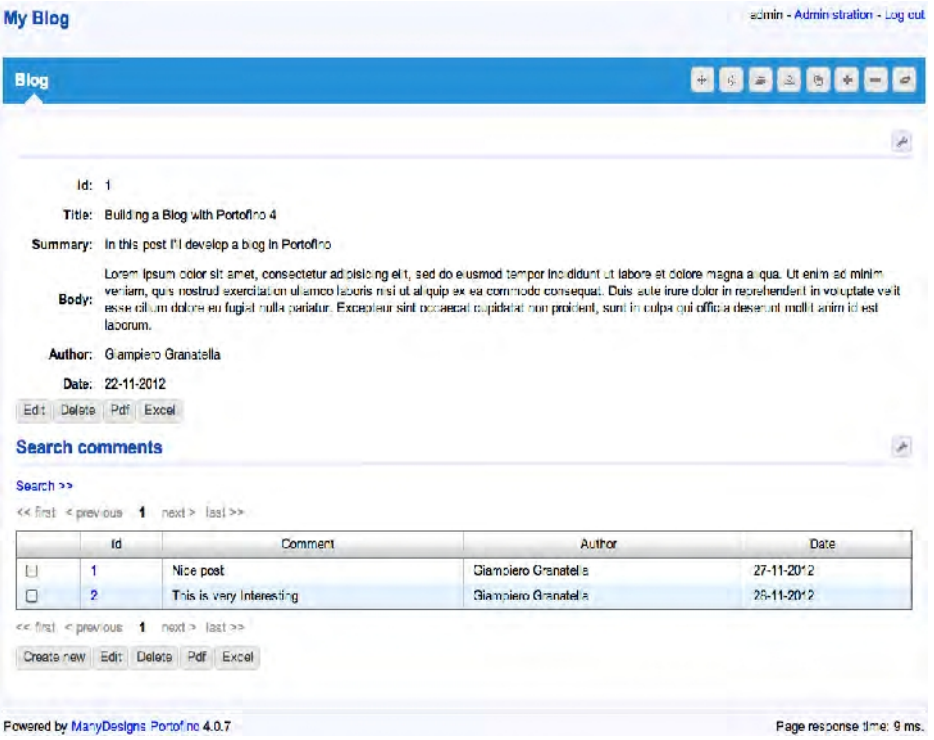

Figure 3. An example master-detail CRUD page

By default each page shows all the records but you can configure the generated pages and filter the data, you create a view for example with a “where” clause in the query.

You can also create connections to several databases, which allows you to use legacy data sources and new ones at the same time. Portofino allows virtual relationships between records in different databases. If you are building the database from scratch, Portofino supports Liquibase [4] for database versioning.

Improving the page content and appearance

The wizard is the first tool to use. It creates a working, although not yet complete application. When it finishes you can start working on the generated pages and create new ones (html pages, charts, ...).

This is a common refinement step. The pages, created by the wizard, are reorganized and new pages are created. You can create html pages as introductory text, summary or documentation.

Pages are organized hierarchically and you can see this structure in the left menu, which offers a navigation from the root page to the leafs. To further simplify navigation, each page shows breadcrumbs.

You can create text pages and attach them to the navigation tree. Page contents can be edited on-line. Simply visit a page, click on "edit", make the changes using the on line html editor. You can apply formatting, link other pages, embed images in the text or attach files to pages.

Familiarizing with the application structure

On the file system, the app is located under /apps/default which has the following directories:

- **blobs**, a directory with blobs (documents stored on the file system and available to pages);
- **dbs**, the Liquibase files that track database changesets and version;
- **groovy**, a common Groovy classpath;
- **pages**, all the pages organized in a directory structure that reflects the URL structure; each page has an associated Groovy action and a number of XML configuration files;
- **web,** custom html, jsp, css, images, javascript, and other web resources.

The main editing and administration operations are done via web, but you can work off-line on the app's files as well. A text editor is enough or you can use an IDE (e.g. Eclipse, NetBeans, IntelliJ Idea).

The app directory can be versioned using a VCS systems (CVS, SVN, Mercurial, Git...). This is a straightforward and a good practice.

Improving the application

Portofino offers a many features that you will discover and learn starting use it.

- Authentication and authorization

Portofino uses Apache Shiro for authentication and authorization. You can authenticate users using a table on the database, LDAP, OpenId or Facebook. Users belong to groups and authorizations are based on groups. Each page has a set of permissions (which can be improved or customized) that can be granted to a group. You can set the permissions via web directly on the page.

- The calendar page

Many components of Portofino let you show the data with a particular view. A calendar page shows events in a monthly or agenda view. Events may be taken from the database or from other sources. Any temporal data can be an event, e.g., the milestones of a project, the due date of a payment, the shipment date of a product. An application can have more calendars and events can be related to a user or a group of users.

- Emails

Portofino has an email queue that provides enhanced resilience to ensure that messages are not lost even during a temporary SMTP failure.

- Scheduled operations

Portofino integrates Quartz [5] so that you can write tasks and schedule them to be executed

- Chart page

A chart page uses the integrated JFreeChart library, with different chart types, tooltips, captions and colors. You can visualize the data with pie charts, bar charts, or other kinds and you can make the charts interactive with portions of the chart pointing to a customizable URL.

- Many to many page

This kind of page allows the user to check or uncheck from a list of options that is backed by a many-to-many relationship on the database.

- Custom action or definition of new kind of pages

Portofino is a flexible framework. You can create a "Custom page" that let you define a business logic, interact with the model, and finally redirect to a custom jsp to view the results. This is simple and can be done with a Groovy action (without recompiling or redeploying). At the same time, you can create new kinds of pages to be reused in your applications.

Deeper customizations

Each page has a dedicated groovy action. For example, a '/company' url has a dedicated /pages/company/action.groovy script on the file system. Portofino provides hooks for every page, which are standard methods exposed to allow customization such as business logic, validation, special actions or redirections. When you modify a groovy action, it is immediately reloaded in the webapp with a fast turnaround and a significant increment of the productivity.

Hooks are not the only method to customize an application. Using certain Java annotations annotations in the Groovy action, you can add new buttons and button handlers in the interface, or you can respond to AJAX calls.

If you have utility classes that you want to use across many actions, there is a Groovy classpath in the application where you can put them (the directory Groovy under the default app). Again, any modifications made to the groovy files are immediately picked up by the system.

You can edit the Groovy actions directly in Portofino, using a simple on-line editor. Basic syntax highlighting is provided. You can also use your favorite IDE and debug Groovy scripts on the live system.

Conclusions

Portofino is an innovative web framework that lets you preserve your database and at the same time to have a solid and modern web applications with CRUD, CMS and many other features.