

Tapestry5HowTos

- [How to get started](#)
- [Preparing your development environment](#)
- [Components](#)
- [Code snippets](#)
 - [General](#)
 - [Select Components](#)
 - [Inversion of Control \(IOC\)](#)
 - [Binding Prefixes](#)
 - [Localization](#)
 - [Streaming Data](#)
- [Integration with other frameworks](#)
 - [Hibernate](#)
 - [JPA](#)
 - [Spring](#)
 - [Google Web Toolkit \(GWT\)](#)
 - [EJB / JEE](#)
- [Security](#)
 - [Authentication and Authorization related](#)
 - [Integrity](#)
- [Bug workarounds / patches](#)
- [Meta info](#)
- [Coding traps](#)
- [How to contribute](#)
- [IRC](#)

How to get started

In order to quickly get up to speed on Tapestry 5, there are numerous information sources besides this Wiki available:

- [The Tapestry Tutorial](#)
- [Tapestry for Nonbelievers](#)
- [Getting Started \(using the Maven Archetype\)](#)
- [Tapestry site](#) and [Component reference](#)
- Search the mailing list with Google by including Google entry: "YOUR KEYWORDS T5 site:mail-archives.apache.org/mod_mbox/tapestry-users/"
- [Tapestry 5 Training](#) - Tapestry 5 overview with practical work
- [Tapestry 5 Blogs](#) - A list of blogs dealing with Tapestry 5
- [Open Source Apps](#) - List of Tapestry 5 open source applications

Preparing your development environment

- [Useful configurations for development](#) - Tapestry 5 configurations that will make your development easier
- [Using Eclipse and Jetty](#)
- [Using Eclipse and Jetty with Jetty Runner](#)
- [Using IDEA and Jetty](#)
- [Running via a main class in any IDE](#)
- [Eclipse code templates](#)
- [IntelliJ IDEA live templates](#)
- [Create a skeleton Tapestry 5 project and run it \(Eclipse, Maven, m2eclipse\)](#)
- [Eclipse template editing](#) - Customize Eclipse for Tapestry .tml template editing
- [Eclipse Code Completion](#) - Using Eclipse JSP editor and custom tld file for code completion

Components

- [Components Reference](#)
- [Jumpstart - Components configuration examples](#)
- [Check Out the Source Code](#) - Examples and Demos of many components in the source code, like the Palette, Grid, BeanEditForm, etc.
- [Creating Your Own Components](#) - Creating your own components is really simple, and fun to do as well. (This example implements a www.dhtmlgoodies.com drag-and-drop-folder tree component)
- [Creating Your Own Component Library](#) - Steps to create a reusable library (JAR) of T5 components.
- [Outputting Locale-formatted Numbers](#) - Creating a new component for formatting (numbers) according to locale, using literals (integer, decimal or currency). It's based on Output component.
- [Creating a Tab Panel](#) - Explains how to create a tab panel.
- [Submit with Context](#) - A Submit component that passes a context
- [Submit with Context and Navigation](#) - A Submit component that passes a context and allows event handler to return navigational object.
- ["Any" Component](#) - A deprecated but still useful component that does ... nothing (by default)
- [Image Component](#) - Another deprecated component that's useful for making "previewable" templates
- [Tapestry 5 Components](#) - By Shing Hing Man
- [Tapestry 5 components at Google Code](#) - Several projects with Tapestry 5 components – share your components here.
- [Tree Component](#) - A tree component based on DHTMLgoodies.com's drag 'n drop folder tree
- [Enhanced Palette Component](#) - Based heavily on the existing core palette and uses Matt Kruse's Option Transfer JavaScript (Works in IE!)
- [Palette Events](#) - How to invoke a JavaScript function when the selected items are added/removed/reordered in a Palette

- [Grid Component Tricks](#) - Some tricks using the Grid component
- [Tapestry 5 and JavaScript Explained](#) - A detailed look at how Tapestry 5 integrates JavaScript into your application. In this article we create a mixin for adding a JavaScript confirmation box to link components, explaining each step.
- [Advanced Layout Component](#) - A sophisticated Layout component
- [Simple And Powerful Layouts](#) - Creates flexible layouts, using a simple reusable component.
- [Layout with Menu](#) - Layout with added menu bar
- [Enum as Component Parameter](#) - Defining a new enum as a component parameter, and adding the string -> enum type coercion for it
- [Captcha](#)

Code snippets

General

- [How to Format Date/Time, etc](#) - How to format Date, Time, timestamps, Decimals, Numbers, Currencies
- [Localized Date Field](#) - How to create a localized DateField for BeanModel.
- [Outputting Raw Html](#) - How to output raw, unfiltered HTML
- [Linking To Non Tapestry URIs](#) - How to link to non-Tapestry URIs (useful for Acegi integration)
- [Obtaining the HttpServletRequest](#) - How to obtain the HttpServletRequest, HttpServletResponse and HttpSession if you really need them
- [Creating A Property Edit Block](#) - Explains how to create a property editor and contribute it to the [BeanEditForm](#).
- [How to Use Forms](#) - Examples of how forms are connected to methods in your code.
- [Exception Page](#) - Displaying your own exception page.
- [Redirect Exception](#) - Redirect to another URL from anywhere inside a page or component.
- [Using Tapestry For Custom Error Pages](#) - How to serve a custom 404 page, or other error, using Tapestry
- [Changing Locale](#) - Change the locale of the user.
- [Including JavaScript](#) - Explains how to add your own custom JavaScript functions.
- [Integrating jQuery](#) - Include jQuery and it's plugins to spice up your pages!
- [Making a Basic Crud App](#) show you how easy is the conception of a CRUD cycle with Tapestry 5
- [Running Tasks in a Separate Thread](#) - run a task (Runnable) in a separate thread
- [Running Tasks in a WorkQueue](#) - run multiple task through a WorkQueue with one or more threads
- [Using a Persistent Field Strategy](#) - How to create a PersistentFieldStrategy for the Persist annotation.
- [Adding Validators](#) - How to add your own validators to tapestry
- [Outputting XHTML](#) - How to override the default markup model to always use XHTML compliant tags
- [Reading Symbols From a Property File](#) - How to have symbols coming from a property file and how to use them in your pages/components
- [Getting the Webapp Path](#) - How to get the real path of the webapp directory.
- [Reading a File from Webapp Context](#) - How a service reads a file from web app context folder
- [Using Tapestry Test](#) - How to actually use Tapestry test
- [Visualizing the Component Hierarchy](#) - How to visualize the component hierarchy just on your pages in the browser
- [Conditionally Rendering Attributes](#) - How to conditionally render HTML tag attributes in Tapestry templates
- [Send Error Page as Email](#) - How to send a dev-mode error page in an production-mode environment (tap version 5.2.6)
- [Get an HTML String from a render command parameter](#) - How to use tapestry's rendering engine to convert a parameter specified in TML to a serverside HTML string

Select Components

- [Displayable Selection Model](#) - A selection model for objects that implement an interface. Used for select components and more.
- [Select Component Alternative](#) - An alternative implementation of the Select core component based on the Daniel Jue's GenericSelectionModel and GenericValueEncoder classes.
- [Multiple-Select Component](#) - Explains how to build a multiple select component based on Daniel Jue's GenericValueEncoder classes.
- [Selecting with Objects](#) - another example on how to make <SELECT> filled with objects
- [Selecting with Objects, #2](#) - yet another example with on how to make <SELECT> on objects, now with annotations.

Inversion of Control (IOC)

- [Using Tapestry-IOC Alone](#) - Example of using tapestry-ioc standalone (without web part). And the same, but with tapestry-hibernate added [Tapestry5HowToLocAndHibernate](#)
- [Creating a Dispatcher](#) - How to create a completely transparent access control system by implementing a Dispatcher.
- [Creating a Dispatcher, part 2](#) - A supplemental article showing how to access request-specific data (ASOs) from a singleton service. To demonstrate we expand on the code shown in the first article.
- [Decorating a Service](#) - An example of a simple service decorator
- [Creating a Configurable Service](#) - How to use symbols to create a sandbox and production version of a service. Requires [Tapestry5HowToReadSymbolsFromPropertiesFile](#) (PayPal example)

Binding Prefixes

Bindings are used in templates `$(message:myMessageKey)` or as attribute to components `<t:mycomponent value="literal:Hello world"/>` to access message in resource bundles, properties, pass String literals and many more. The following pages explain how to write your own bindings.

- [Adding a "list:" Binding Prefix](#) - An example of a list: binding prefix. Useful for Action/Event/PageLink context etc.
- [Adding a "map:" Binding Prefix](#) - An example of a map: binding prefix. Useful for supplying key=value pairs to components.
- [Adding a "cycle:" Binding Prefix](#) - An example of an cycle: binding prefix. Makes zebra effect simple to do `<t:grid rowClass="cycle:line1,line2">`
- [Adding a "messageformat:" Binding Prefix](#) - Describes how to add a messageformat prefix for specifying values for use in messages inline.

Localization

- [Domain-level Localization](#) - An example of how to overwrite the Locale domain-dependent
- [Changing Locale](#) - Change the locale of the user.
- [Outputting Locale-formatted Numbers](#) - Creating a new component for formatting (numbers) according to locale, using literals (integer, decimal or currency). It's based on Output component.

Streaming Data

- [Creating Pie Charts in a Page](#) - An example of dynamic charts inside a page (Using [StreamResponse](#))
- [Creating a Simple Graph Component](#) - An example of simple pie chart component (Using [StreamResponse](#))
- [Creating a Generic Graph Component](#) - A component that takes in a JFreeChart as a parameter
- [Creating a Dynamic PDF](#) - A simple example of how to return a dynamically generated PDF from an event (Using [StreamResponse](#))
- [Streaming an Existing Binary File](#) - A simple example of how to stream an existing file, like an image. (Using [StreamResponse](#))

Integration with other frameworks

- [Module Registry](#) - List of different extensions and modules

Hibernate

- [Using Tapestry-Hibernate](#) - Example/Tutorial of how to use Tapestry-Hibernate in your code.
- [Using Tapestry-IOC and Tapestry-Hibernate Standalone](#) - Example of using tapestry-oc+tapestry-hibernate standalone (without web part).
- [First Project with Tapestry 5, Spring & Hibernate](#) - A translation of a french article about integrating Spring, Hibernate and Tapestry
- [Manipulating Classes at Runtime](#) show you how easy is to add class methods at runtime.
- [Using HibernateGridDatasource](#) how to use the [HibernateGridDatasource](#) class from tapestry-hibernate module.
- [Using HibernateGridDatasource with HQL](#) simple hibernated grid datasource using HQL (Hibernate Query Language).
- [Using HibernateGridDatasource with Criteria API](#) simple hibernated grid datasource using hibernate criteria API.
- [Flexible Abstract GridDataSource](#) very flexible abstract [GridDataSource](#) for using any org.hibernate.Query object, so you can use HQL, SQL, Criteria, DaO, etc.
- [Brainless Entity Timestamping](#) - How to have entities automatically time stamped using tapestry-hibernate, simply by adding certain fields!
- [Avoiding Lazy Initialization Exceptions](#) - How to avoid "Lazy Initialization" exceptions
- [Avoiding DOWSID Exceptions](#) - How to avoid "Different Object with same ID" exceptions

JPA

- [Tapestry JPA Guide](#)

Spring

- [Integrating with Spring](#) - How to inject Spring beans into Tapestry pages and components
- [Integrating with Spring and JPA](#) - Quick tutorial on using Tapestry with Spring Transactions and JPA
- [Integrating with Spring Transactions](#) - brief tutorial of how to integrate Tapestry5 components with Spring transactions.
- [Testing with Tapestry-Spring](#) - How to test Tapestry pages which use Spring Beans

Google Web Toolkit (GWT)

- [Integrating with GWT](#) - An example based heavily on Pär Dahlberg's blog example on <http://www.pmdit.se/blog>, with generalized design tips.

EJB / JEE

- [Support @EJB Annotation](#) - Support @EJB Annotation in Tapestry pages / components

Security

Security is the condition of being protected against danger, loss, and criminals.

Authentication and Authorization related

- [Using Acegi Security and CAS](#) - Francois Armand explains how he combines T5, Acegi Security and CAS (A Central Authentication Service, providing intranet single sign-on).
- [Securing Pages with ACEGI Annotations](#) - How to secure Tapestry5 pages with Java5 annotations using the Acegi security framework
- [Securing Pages with ACEGI XML](#) - How to use Acegi Security with Tapestry 5, in the classical way (no annotations, some XML)
- [Controlling Access with Annotations](#) - Another supplemental article based on [Tapestry5HowToCreateADispatcher2](#) showing how to control access to pages based on an annotation.
- [Securing with ACEGI & LDAP](#) - How to protect you application using Acegi and LDAP
- [Securing with Spring & LDAP](#) - How to protect you application using Spring Security and LDAP
- [Spring Security & OpenID](#) - How to provide OpenID authentication to your clients using tapestry-spring-security
- [Mitigating Login Attacks](#) - How to track failed logins and mitigate brute force or dictionary attacks

Integrity

- [Preventing Cross-site Request Forgeries](#) - How to protect against 'Cross-site request forgeries' (CSRF)
- [Preventing Client Side Changes](#) - How to protect server side generated values against client side changes using a HMAC

Bug workarounds / patches

- [Using UTF-8 Encoding](#) - How to use UTF-8 encoding to support foreign characters
- [Root Path Empty Response Patch](#) - A runtime patch for the root path empty response bug
- [TAPESTRY-1745](#) - Tapestry5.0.5 Palette Component IE Bug Patch
- [Running Tapestry on JBoss 5](#) - How to run Tapestry 5 on JBoss 5
- [Running Tapestry on JBoss 6.0](#) - How to run Tapestry 5 on JBoss 6.0
- [Running Tapestry on JBoss 6.1](#) - How to run Tapestry 5 on JBoss 6.1
- [Running Tapestry on JBoss 7.1](#) - How to run Tapestry 5 on JBoss 7.1
- [Running Tapestry on WebSphere](#) - How to run Tapestry 5 on Websphere (6.1)
- [Tapestry5 bug tracker](#)

Meta info

- [Quickstart with Maven 2.0.6](#) - If you've had some trouble getting maven2.0.6 to compile the quickstart, here's how to do it.
- [Where to Store External Resources](#) - Where to store resources like CSS, JS and Images in your application
- [Where to Store Configuration Resources](#) - Where to store resources like Hibernate configuration files, logging properties, and web application context information in your application
- [Searching the Mailing List](#) - Force Google to only search the mailing list (results found in these lists are often hidden due to the limited #links)
- [Overriding the Default Error Message Banner](#) - How to override the default error message banner
- [Dealing with Trolls](#) - How to deal with trolls on the list, and one in particular

Coding traps

- [How Not to Make a Custom Component](#) - If you are making a custom component, there are few things you shouldn't do...
- [Component Class Resolver Error](#) - When organizing your component (or page) directories, pay special attention!
- [Caveats](#) - or more literally, "let him beware".

How to contribute

- [Contributing New Translations](#)

IRC

Chris Lewis mentioned Tapestry's IRC: #tapestry AT irc.freenode.net: "There's usually 5 ~ 8 of us in there, and we work/talk about Tapestry."

Please add more 'how tos'. Remember to prefix the page name with Tapestry5.