

Tapestry 5.4

Teach An Old Horse New Tricks And
Why Full-Stack Frameworks Still Matter

Thilo Tanner, IT Lead
RepRisk AG

RepRisk

RepRisk is a leading business intelligence provider specializing in dynamic **environmental, social and governance (ESG)** risk analytics and metrics.

**Are you kidding me,
another web
framework?!**

spring

express



Symfony



Struts



APACHE WICKET

adding me

django

another framework



Mentawai Web Framework



Pyramid



yii framework

Comparing the hottest
713 web frameworks
are meant for getting
things done.

Why Tapestry matter?



apache
tapestry 5
Code less, deliver more.

Tapestry is service-oriented

Customizable

```
@ImportModule({ MySubModule.class })
public final class AppModule {
    public static void bind(ServiceBinder binder) {
        binder.bind(MyService.class, MyServiceImpl.class);
        binder.bind(OtherService.class, OtherService.class);
    }

    public static void contributeApplicationDefaults(
        MappedConfiguration<String, Object> configuration) {
        configuration.add(SymbolConstants.SUPPORTED_LOCALES, "en");
    }
}
```

```
public class MyPage {
    @Inject private MyService myService;

    public void onActivate() {
        myService.doSomething();
    }
}
```

Tapestry is component-oriented

```
public class Panel {  
    @Parameter(required = true, defaultPrefix = BindingConstants.LITERAL)  
    @Property  
    private String title;  
  
    @Parameter(defaultPrefix = BindingConstants.LITERAL, value = "default")  
    @Property  
    private String option;  
}
```

```
<t:if xmlns:t="http://tapestry.apache.org/schema/tapestry_5_4.xsd">  
    <div class="panel panel-{$option}">  
        <div class="panel-heading">${title}</div>  
        <div class="panel-body">  
            <t:body />  
        </div>  
    </div>  
</t:if>
```


Even pages are components

```
/hello?name=world
```

```
public class Hello {  
    @ActivationRequestParameter  
    @Property  
    private String name;  
}
```

```
<html t:type="layout" title="Hello World"  
    xmlns:t="http://tapestry.apache.org/schema/tapestry_5_4.xsd"  
    xmlns:p="tapestry:parameter">  
    <h1>${name}</h1>  
</html>
```

Tapestry in the wild

JPA - T5

```
public class EditFooBeans {  
    @PageActivationContext  
    @Property  
    private FooBean fooBean;  
  
    @Inject private FooBeanDAO fooBeanDAO;  
  
    @Inject private AlertManager alertManager;  
  
    @CommitAfter  
    public Object onSuccess() {  
        fooBeanDAO.merge(fooBean);  
        alertManager.success("success");  
        return IndexFooBeans.class;  
    }  
}
```

Eew, JPA! - jOOQ

```
public final class DatabaseModule {  
    public static Configuration buildConfiguration(DataSource dataSource) {  
        DefaultConfiguration configuration = new DefaultConfiguration();  
        configuration.set(dataSource);  
        configuration.set(new DefaultRecordMapperProvider());  
        configuration.set(SQLDialect.MYSQL);  
        return configuration;  
    }  
}
```

@Scope(ScopeConstants.PERTHREAD)

```
public static DSLContext buildDSLContext(Configuration configuration) {  
    return DSL.using(configuration);  
}  
}
```

```
public class CompanyDAOImpl implements CompanyDAO {  
    private DSLContext ctx;
```

public IssueRDAOImpl(DSLContext ctx) {

```
    this.ctx = ctx;  
}  
...  
}
```

Scaffolding

```
<t:grid source="users" row="user">
  <p:lastNameCell>
    <t:pagelink page="user/view" context=„user.id">
      ${user.lastname}
    </t:pagelink>
  </p:lastNameCell>
  <p:empty>
    <p>There are no users to display</p>
  </p:empty>
</t:grid>
```

```
<t:beaneditform t:id="user" reorder="lastname,firstname"/>
```

```
<t:beandisplay object="user" reorder="lastname,firstname"/>
```

JavaScript

```
@Import(module = { "bootstrap/modal", "modal" })
public class Modal {
    void beforeRenderBody(MarkupWriter writer) {
        writer.attributes("modal", "modal");
    }
}
```

```
define(['jquery'], function($) {
    if($('#modal').length == 0) {
        $('body').append('<div class="modal fade" id="modal"></div>')
    }

    $('[modal]').click(function(e) {
        e.preventDefault();
        $('#modal').load(e.currentTarget.href, function() {
            $('#modal').modal('show');
        });
    });
});
```

```
<t:pagelink t:mixins="modal">Open Modal</t:pagelink>
```

JavaScript

```
public class DeleteCompany {
    @PageActivationContext
    @Property
    private Company company;

    @Inject private Request request;
    @Inject private Block modal;
    @Inject private BlockRenderer blockRenderer;
    @Inject private CompanyDAO companyDAO;
    @Inject private AlertManager alertManager;

    public Object onActivate() {
        return request.isXHR() ? blockRenderer.render(modal) : null;
    }

    @CommitAfter
    public Object onDelete() {
        companyDAO.delete(company);
        alertManager.success("Company successfully deleted");
        return IndexRSSFeeds.class;
    }
}
```

JavaScript

```
<t:layout xmlns:t="http://tapestry.apache.org/schema/tapestry_5_4.xsd"
  xmlns:p="tapestry:parameter"
  title="Delete Company">

  <div class="page-header">
    <h1>Delete Company</h1>
  </div>

  <p>${deleteQuestion}</p>

  <t:eventlink t:id="delete" class="btn btn-danger">Delete</t:eventlink>

  <t:block t:id="modal">
    <t:util.modal event="delete" label="Delete"
      option="danger" icon="times" title="Delete Company">
      ${deleteQuestion}
    </t:util.modal>
  </t:block>

</t:layout>
```


Integration - Camel

```
public class TapestryRegistry implements Registry {
    private ServiceActivityScoreboard serviceActivityScoreboard;
    private ObjectLocator objectLocator;

    public TapestryRegistry(ServiceActivityScoreboard serviceActivityScoreboard,
                            ObjectLocator objectLocator) {
        this.serviceActivityScoreboard = serviceActivityScoreboard;
        this.objectLocator = objectLocator;
    }

    @Override
    public Object lookupByName(String name) {
        ServiceActivity serviceActivity = getActivityForName(name);
        if(serviceActivity != null) {
            return objectLocator.getService(serviceActivity.getServiceInterface());
        }
        throw new IllegalArgumentException("Unknown Service: " + name);
    }
}
```

...

Integration - Camel

@Advise

```
@Integration
public static void adviseTransaction(JpaTransactionAdvisor advisor
                                     MethodAdviceReceiver receiver) {
    advisor.addTransactionCommitAdvice(receiver);
}
```

```
from("direct:importRSS")
    .routeId("importRSS")
    .beanRef(s(TracerService.class), "createTracerIfNotExists")
    .setBody(body().regexReplaceAll("\\r", ""))
    .split(body().tokenize("\n"))
        .filter().method(s(ResourceService.class), "isURI")
            .filter().method(s(RSSFeedImportService.class), "isFeedNew")
                .beanRef(s(RSSFeedImportService.class), "createRSSFeed")
            .end()
        .end()
    .end()
    .beanRef(s(TracerService.class), "closeTracer")
;
```

Integration - Camel

```
<t:zone xmlns:t="http://tapestry.apache.org/schema/tapestry_5_4.xsd"
        xmlns:p="tapestry:parameter"
        t:mixins="zonerefresh" ZoneRefresh.period="1">
```

```
<h2>Tracer Counters</h2>
```

```
<ul class="list-group">
```

```
<t:loop source="tracer?.counters" value="counter">
```

```
<li class="list-group-item">
```

```
<span class="badge">${counterValue}</span>
```

```
  ${counterName}
```

```
</li>
```

```
</t:loop>
```

```
</ul>
```

```
<h2>Tracer Log</h2>
```

```
<pre style="overflow:y-scroll">
```

```
  ${tracer?.log}
```

```
</pre>
```

```
...
```

```
</t:zone>
```

Asynchronous Processing - T5

```
@Startup
public static void
scheduleJobs(@Symbol(SymbolConstants.PRODUCTION_MODE)
             boolean productionMode,
             PeriodicExecutor executor,
             final MyService myService) {

    if(!productionMode) {
        return;
    }

    executor.addJob(new CronSchedule("0 0 * * * ?"), "Job",
        new Runnable() {
            @Override
            public void run() {
                myService.doSomething();
            }
        });
}
```

Asynchronous Processing - Akka

```
public class TapestryActorProducer implements IndirectActorProducer {
    private ObjectLocator objectLocator;
    private Class<? extends Actor> actorClass;

    public TapestryActorProducer(ObjectLocator objectLocator,
                                 Class<? extends Actor> actorClass) {
        this.objectLocator = objectLocator;
        this.actorClass = actorClass;
    }

    @Override
    public Actor produce() {
        return objectLocator.autobuild(actorClass);
    }

    @Override
    public Class<? extends Actor> actorClass() {
        return actorClass;
    }
}
```

Asynchronous Processing - Akka

```
@ServiceId("MyActor")
public static ActorRef buildMyActor(ActorSystem actorSystem,
                                   ObjectLocator objectLocator) {
    return actorSystem.actorOf(
        Props.create(
            TapestryActorProducer.class, objectLocator, MyActor.class), „myactor"
    );
}
```

```
public class MyActor extends UntypedActor
{
    @Inject
    private MyService service;

    @Override
    public void onReceive(Object object) throws Exception {
        service.doSomething(object);
    }
}
```

Reports

Live Reload

```
public StreamResponse onGenerateReport() {
    ...
    JasperPrint reportPrint = JasperFillManager.fillReport(
        supplierReport, parameters, new JREmptyDataSource()
    );

    JRPdfExporter exporter = new JRPdfExporter();

    exporter.setExporterInput(new SimpleExporterInput(reportPrint));
    exporter.setExporterOutput(new SimpleOutputStreamExporterOutput(baos));

    exporter.exportReport();

    return new ByteArrayStreamResponse(baos.toByteArray(), "application/pdf");
}
```

	Company Name / Sector / Location	ESG Risk Rating	Top Issues	Top Topics
6	Vinisol SAS Chemicals Colombia	B	<ol style="list-style-type: none">1. Violation of National Legislation2. Impacts on Communities3. Local Pollution4. Impacts on Ecosystems/Landscapes5. Waste Issues	<ol style="list-style-type: none">1. Genetically Modified Organisms2. Negligence3. Asbestos4. Indigenous People5. Endangered Species
7	BASF SE Chemicals	BB	<ol style="list-style-type: none">1. Impacts on Communities2. Controversial Products and Services3. Impacts on Ecosystems/Landscapes	<ol style="list-style-type: none">1. Genetically Modified Organisms2. Indigenous People3. Fracking

REST - Tynamo REST Module

```
@Path("/company")
public interface CompanyResource {
    @GET
    @Path("/search")
    @Produces({"application/json"})
    List<Company> search(@QueryParam("q") String query);
}
```

```
public class CompanyResourceImpl implements CompanyResource
{
    private CompanyDAO companyDAO;

    public CompanyResourceImpl(CompanyDAO companyDAO) {
        this.companyDAO = companyDAO;
    }

    @Override
    public List<Company> search(String query) {
        return companyDAO.findByNameLike(query);
    }
}
```


Security - Tynamo Security Module

```
public class JPAAuthorizationRealm extends AuthorizingRealm { ... }
```

```
@Contribute(HttpServletRequestFilter.class)
@Marker(Security.class)
public static void setupSecurity(Configuration<SecurityFilterChain> configuration,
                                SecurityFilterChainFactory factory) {
    configuration.add(factory.createChain("/login**").add(factory.anon()).build());
    configuration.add(factory.createChain("/rest/**").add(factory.basic()).build());
    configuration.add(factory.createChain("/**").add(factory.user()).build());
}
```

```
@RequiresRoles(value = { "ADMIN", "SUPERUSER" }, logical = Logical.OR)
```

```
public class IndexAdmin { ... }
```

```
<t:security.hasRole role="ADMIN">
```

```
...
```

```
</t:security.hasRole>
```

Deployment - Embedded Jetty

```
java -jar myapp.war 9090
```

Ingredients

- ▶ StartServer Main Class (Setup and Start Jetty)
- ▶ Maven WAR Plugin (MANIFEST)
- ▶ Maven Antrun Plugin (Copy StartServer.class)
- ▶ Maven Dependency Plugin (Unpack Jetty Classes)
- ▶ HTTP Proxy Server (nginx)

Your code is fully tested, right?

```
@BeforeClass
public void buildMetadata() throws Exception {
    String appPackage = "org.example.app";
    String appName = "LocaleApp";
    PageTester tester = new PageTester(appPackage, appName, "src/main/webapp");

    HtmlMetadataExtractor extractor =
        tester.getService(HtmlMetadataExtractor.class);

    uri = new URI("http://techcrunch.com/2013/03/26/embedly-beyond-embedding/");

    // mock HTML
    URL url = Resources.getResource("org/example/app/html-test.html");
    URIResource resource = mock(URIResource.class);
    when(resource.getContent()).thenReturn(Resources.toByteArray(url));
    when(resource.getUri()).thenReturn(uri);
    metadata = extractor.extract(resource);
}

@Test
public void textTest() {
    Assert.assertTrue(metadata.getText().length() > 100);
}
```

**Tapestry is so great, I
want to use it for all
my projects**



Source: <http://blog.codinghorror.com/the-php-singularity/>

**Houston,
we have a problem**

Books

Igor Drobiazko

Tapestry 5 - Rapid web application development in Java

Tapestry 5: Die Entwicklung von Webanwendungen mit Leichtigkeit

<http://www.tapestry5book.com/>

It's source code, stupid

- * The source code is still the best documentation
- * Learn how to create good components
- * Understand Tapestry Internals

Documentation / Mailing Lists

<https://tapestry.apache.org/documentation.html>

<https://tapestry.apache.org/community.html>

The screenshot shows the Apache Tapestry 5 Documentation website. At the top, there is a navigation bar with links: Home | Getting Started | Documentation | Download | About | Community | Apache | Sponsorship | Thanks. Below the navigation bar is the Apache Tapestry 5 logo, which features a green unicorn icon and the text "apache tapestry 5 Code less, deliver more." To the right of the logo is the word "Documentation" in green. Further right is a search box with the text "Tapestry docs, issues, wikis & blogs:" and a "Search" button. Below the search box is a breadcrumb trail: "Apache Tapestry > Documentation" with an "edit" link. The main content area is divided into two columns. The left column has a heading "Highlights" and a paragraph: "Welcome to the Tapestry 5 Documentation. A rich collection of guides is available to teach beginners the basics and help even experienced developers deepen their understanding of Tapestry's power. Code less, deliver more!". Below this paragraph is a table with four rows, each representing a guide. The right column has a heading "All Topics" and a list of links, including "Introduction", "Getting Started", "Principles", "Tapestry Tutorial", "Dependencies, Tools and Plugins", "Creating The Skeleton Application", "Loading the Project Into Eclipse", "Exploring the Project", "Implementing the Hi-Lo Guessing Game", "Using BeanEditForm To Create User Forms", "Using Tapestry With Hibernate", "User Guide", "Project Layout", "Configuration", and "Runtime Exceptions".

Home | Getting Started | Documentation | Download | About | Community | Apache | Sponsorship | Thanks

apache
tapestry 5
Code less, deliver more.

Documentation

Tapestry docs, issues, wikis & blogs:
 Search

Apache Tapestry > Documentation [edit](#)

Welcome to the Tapestry 5 Documentation. A rich collection of guides is available to teach beginners the basics and help even experienced developers deepen their understanding of Tapestry's power. *Code less, deliver more!*

Highlights

These are the most useful starting points for common needs.

Introduction	An overview of Tapestry's general approach and philosophy
Getting Started	A quick guide to creating your first Tapestry project, using Maven
Tapestry Tutorial	Picks up where <i>Getting Started</i> leaves off, explaining in greater detail how Tapestry works
User Guide	Detailed articles on every Tapestry feature

All Topics

- [Introduction](#)
- [Getting Started](#)
- [Principles](#)
- [Tapestry Tutorial](#)
 - [Dependencies, Tools and Plugins](#)
 - [Creating The Skeleton Application](#)
 - [Loading the Project Into Eclipse](#)
 - [Exploring the Project](#)
 - [Implementing the Hi-Lo Guessing Game](#)
 - [Using BeanEditForm To Create User Forms](#)
 - [Using Tapestry With Hibernate](#)
- [User Guide](#)
 - [Project Layout](#)
 - [Configuration](#)
 - [Runtime Exceptions](#)

Tapestry Jumpstart

<http://jumpstart.doublenegative.com.au/jumpstart7/>

[What's This?](#) [Download](#) [Installation](#) [Orientation](#) [Tips](#) [Troubleshooting](#) [FAQ](#) [Bugs and Enhancements](#)

JumpStart 7.0 Preview 23 For Tapestry 5.4 Beta 22

[Hello, World](#)
["Hello, World" Explained](#)

[Bootstrap Styling](#)
[Stylesheets](#)
[Less](#)

[Output](#)
[Better Output](#)
[Easy Output \(Using BeanDisplay\)](#)
[Easy Output \(Entity Bean\)](#)
[Total Control Output](#)

[Methods](#)
[Annotations](#)

[PageLinks](#)

[Activation Context](#)
[More Parameters](#)
[Variable Parameters](#)

[Activation Request Parameters](#)

[Grid](#)
[Styled Grid](#)

[Input and Events](#)

[Forms](#)
[Forms Explained](#)

[EventLinks](#)
[EventLinks Explained](#)

[@OnEvent](#)
[Multiple Method Matches](#)

[ActionLinks](#)

[Storing Data in a Page](#)

[Core Input Components](#)
[HTML5 Input Types](#)

[Return Types](#)

[Translators](#)
[Primitives Translators](#)
[Augmenting Translators](#)
[Contributing Translators](#)

[Validators](#)

[onActivate and onPassivate](#)
[Handling a Bad Context](#)

[Exception Report Page](#)

[Multiple Submits](#)
[Multiple Forms](#)

[LinkSubmits](#)

[Select](#)
[Varied Select](#)
[Easy Object Select](#)
[Total Control Object Select](#)
[Id Select](#)
[Easy Id Select](#)

[Putting It All Together](#)

[Easy CRUD](#)
[Total Control CRUD](#)
[One Page CRUD](#)
[Filter CRUD](#)

[Kaptcha](#)
[Alerts](#)

[Creating Components](#)
[Component Around Content](#)
[The Layout Component](#)

[Third-Party Components](#)
[Combining Components](#)
[Wrapping Tapestry Components](#)

[Event Bubbling](#)
[Event Bubbling with Context](#)

[Putting It All Together](#)
[Components CRUD](#)
[Smaller Components CRUD](#)

[Sub-Form Component](#)
[Sub-Form Validation](#)
[Sub-Form As a Field](#)

[MarkupWriter](#)
[Delegate to Renderable](#)

[Mixins](#)
[Creating Mixins](#)

[AJAX EventLinks in a Loop](#)
[AJAX Forms in a Loop](#)
[AJAX Forms in a Loop \(With DEH\)](#)

[AJAX Filtered Grid](#)

[Autocomplete Mixin](#)

[Tree From Database, With Zones](#)

[ProgressiveDisplay](#)
[ProgressiveDisplay Variations](#)

[AjaxFormLoop](#)
[AjaxFormLoop With Holders](#)
[AjaxFormLoop Tailored](#)

[AJAX: What is Called and When](#)

[JavaScript](#)
[Reusable JavaScript](#)
[Robust JavaScript](#)
[JavaScript Mixin](#)
[Creating Mixins: ClickOnce](#)

[jQuery DataTables](#)

Questions?

Thank you for your attention!

Rep Risk AG

Thilo Tanner

Stampfenbachstrasse 42, Zürich, Switzerland

thilo.tanner@reprisk.com