

Sexy Models! An API for Declarative Data Models on the Desktop

James Lemieux, Lombardi Software

Jesse Wilson, Google

<http://publicobject.com/glazedlists/>

TS-3057

How Does Glazed Lists Help Me?

Create a better user experience with less code

Implement deep binding to make your application feel smarter.

Decouple your domain model from your user interface.

Agenda

Introduction to List Transformations

“Deep Binding”

Inferring Hierarchy

Easy Chart Binding

API Highlights

Agenda

Introduction to List Transformations

“Deep Binding”

Inferring Hierarchy

Easy Chart Binding

API Highlights

Glazed Lists Project Profile

“List Transformations in Java™ Programming Language”

- Actively developed
- Free, open source
- Stable and well tested

You Know the API

- ArrayList + Observer Pattern = EventList

Step 1 of 4: BasicEventList

- Holds your data
- Observable

Step 1.5 of 4: TransformedList

- An EventList that decorates another EventList
- Observable

Step 2 of 4: SortedList

- One of the TransformedLists
- A view of your data

Comparator Interface

```
interface Comparator<T> {  
  
    int compare(T alpha, T beta);  
  
}
```

Step 3 of 4: FilterList

- Matcher, MatcherEditor interfaces for creating custom filters
- Built-in matchers for iTunes-style filtering

Matcher Interface

```
interface Matcher<T> {  
  
    boolean matches(T value);  
  
}
```

Step 4 of 4: EventTableModel

- TableFormat provides the columns
- EventList provides the rows

TableFormat Interface

```
interface TableFormat<T> {  
  
    int getColumnCount();  
  
    String getColumnName(int column);  
  
    Object getColumnValue(T baseObject, int column);  
  
}
```



DEMO

Source, Sort, Filter, Display



Agenda

Introduction to List Transformations

“Deep Binding”

Inferring Hierarchy

Easy Chart Binding

API Highlights

Deep Binding?

Your data and view don't line up perfectly

- Songs to artists
- Rockstars by band

Songs to Artists: Input

- Cake, *The Distance*
- Eminem, *Lose Yourself*
- ACDC, *Shoot to Thrill*
- Cake, *Short Skirt Long Jacket*
- ACDC, *Thunderstruck*

Songs to Artists: Output

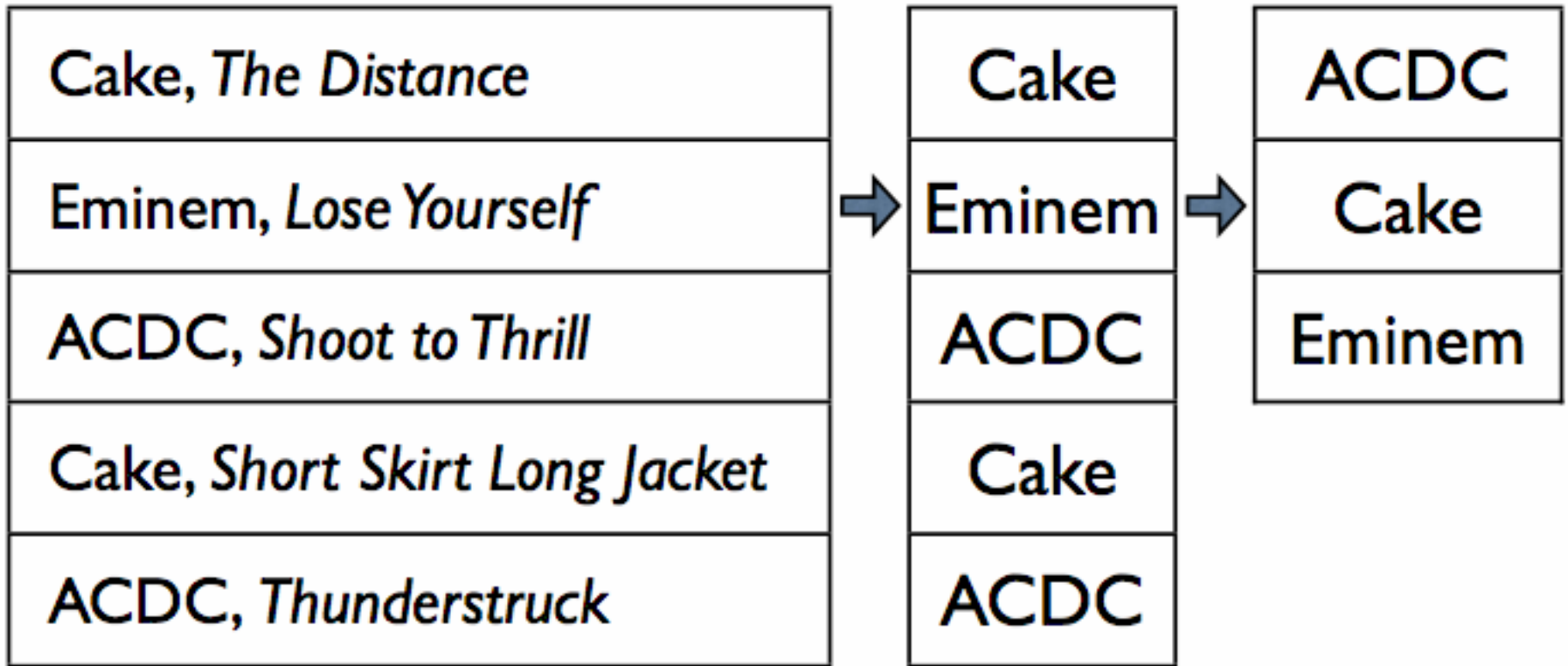
- ACDC
- Cake
- Eminem

Songs to Artists: Pipeline

- `EventList<Song>`
- `FunctionList` of artists
- `UniqueList` of artists

- Add more songs, the corresponding artists show up automagically

Songs to Artists



Rockstars by Band: Input

- Kurt Cobain, *Nirvana*
- Billy Joe, *Green Day*
- Mick Jagger, *Rolling Stones*
- Keith Richards, *Rolling Stones*
- Dave Grohl, *Nirvana*
- Tre Cool, *Green Day*

Rockstars by Band: Output

- *Green Day*: Billy Joe, Tre Cool
- *Nirvana*: Kurt Cobain, Dave Grohl
- *Rolling Stones*: Mick Jagger, Keith Richards

Rockstars by Band: Pipeline

- `EventList<Rockstar>`
- `GroupingList` groups by band using a `Comparator`

Rockstars by Band

Kurt Cobain, <i>Nirvana</i>
Billy Joe, <i>Green Day</i>
Mick Jagger, <i>Rolling Stones</i>
Keith Richards, <i>Rolling Stones</i>
Dave Grohl, <i>Nirvana</i>
Tre Cool, <i>Green Day</i>



<i>Green Day</i> : Billy Joe, Tre Cool
<i>Nirvana</i> : Kurt Cobain, Dave Grohl
<i>Rolling Stones</i> : Mick Jagger, Keith Richards



DEMO

Deep Binding



Agenda

Introduction to List Transformations

“Deep Binding”

Inferring Hierarchy

Easy Chart Binding

API Highlights

TreeList Overview

- Filterable and sortable
- Bind to trees and treetables
- Infers structure from a flat list

Song Hierarchy: Input

- Green Day: *Dookie, Basket Case*
- Green Day: *American Idiot, Welcome to Paradise*
- Cake: *Prolonging the Magic, Sheep go to Heaven*

Song Hierarchy: Output

- Green Day
 - *Dookie*
 - *Basket Case*
 - *American Idiot*
 - *Welcome to Paradise*
- Cake
 - *Prolonging the Magic*
 - *Sheep go to Heaven*

TreeList.Format Interface

```
interface Format<T> {  
  
    void getPath(List<T> path, T element);  
  
    boolean allowsChildren(T element);  
  
    Comparator<T> getComparator(int depth);  
}
```



DEMO

Inferring Hierarchy



Agenda

Introduction to List Transformations

“Deep Binding”

Inferring Hierarchy

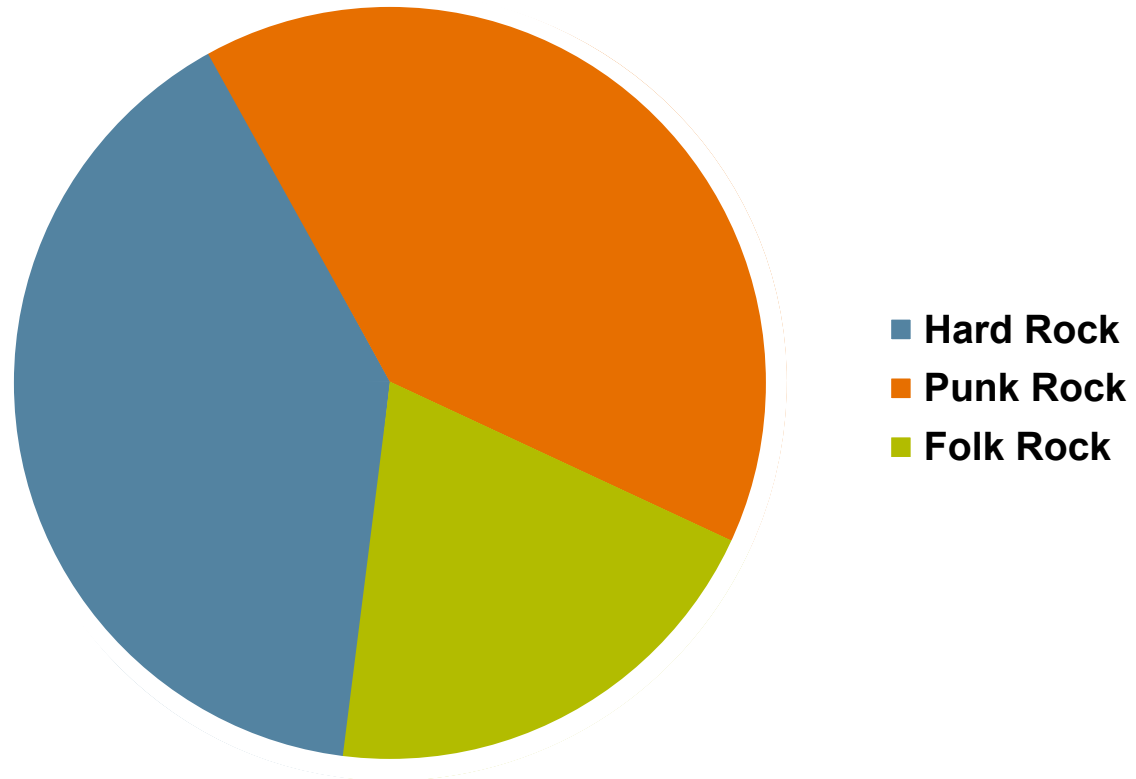
Easy Chart Binding

API Highlights

Songs by Genre: Input

- Punk Rock: Green Day, *Basket Case*
- Punk Rock: Green Day, *Welcome to Paradise*
- Folk Rock: Cake, *Sheep Go to Heaven*
- Hard Rock: ACDC, *High Voltage*
- Hard Rock: ACDC: *Thunderstruck*

Songs by Genre: Output





DEMO

Chart Binding



Agenda

Introduction to List Transformations

“Deep Binding”

Inferring Hierarchy

Easy Chart Binding

API Highlights

Declarative

- Tell Glazed Lists what to do
- Not how to do it

Concurrent

- Read/Write locks are built-in
- Background fetching is easy

- Perceived and actual performance is higher

Productive

- Goal: Write less code
- You provide the business logic, Glazed Lists does the rest
- Common pattern to most EventLists: implement the corresponding interface
- Bean properties for implementing APIs



DEMO

Productive APIs



Agenda

Introduction to List Transformations

“Deep Binding”

Inferring Hierarchy

Easy Chart Binding

API Highlights

Wrapping Up

It's Fast

- Fine-grained events allow Glazed Lists to sort and filter tables with 100,000+ elements
- Uses custom data structures internally to manage indices efficiently

Plays Well With Others

- Swing
- SWT
- SwingLabs
- JFreeChart
- Layout
- Beans Binding (JSR 295)

Interfaces for JavaBeans™ Architecture

```
// uses the method, Song.getArtist()
Comparator<Song> songsByArtist =
    GlazedLists.beanPropertyComparator(
        Song.class, "artist");

// uses Song.getName(), getArtist() and getAlbum()
TableFormat<Song> songsTableFormat =
    GlazedLists.tableFormat(
        Song.class,
        new String[] { "name", "artist", "album" },
        new String[] { "Song Name", "Artist", "Album" });

// true if "Green Day" equals Song.getArtist()
Matcher<Song> artistIsGreenDay =
    Matchers.beanPropertyMatcher(
        Song.class, "artist", "Green Day");
```

Glazed Lists

- Easy for developers
- Great for users
- High performance

- A powerful tool for your toolbox

For More Information

- <http://glazedlists.dev.java.net/>
- <http://publicobject.com/glazedlists/>

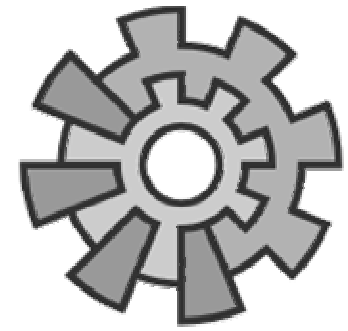


Q&A

James Lemieux

Jesse Wilson





Sexy Models! An API for Declarative Data Models on the Desktop

James Lemieux, Lombardi Software

Jesse Wilson, Google

<http://publicobject.com/glazedlists/>

TS-3057