

# Tapestry 5 Preview

---

Howard M. Lewis Ship  
TWD Consulting, Inc.

# A few observations

---

- “If ...
  - ... I have to type one more angle bracket I'll chop off my pinkie! ”
  - ... I have to run Ant/Maven one more time I'll staple myself to death! ”
  - ... I have to restart Tomcat on more @&\*\$\*\*@! time I'm going to become a sheep farmer! ”
  - ... I have to write any more &%@&^@ boilerplate code I'll start to listen to those voices in my head! ”
- You have a **budget** for mental investment

# Tapestry 5 Goals and Values

---

- Less code is better than more code
- Naming conventions are better than explicit configuration
- Java annotations are better than XML
- Easy is better than hard
- Fast is better than slow
- Things should **Just Work**<sup>™</sup>

# Things Should Be Easy

---

- How much work to create a form to edit a simple object?

## SessionEval

```
surveyId : long  
sessionId : String  
subjectMatter : Rating  
presentation : Rating  
materials : Rating  
comments : String
```

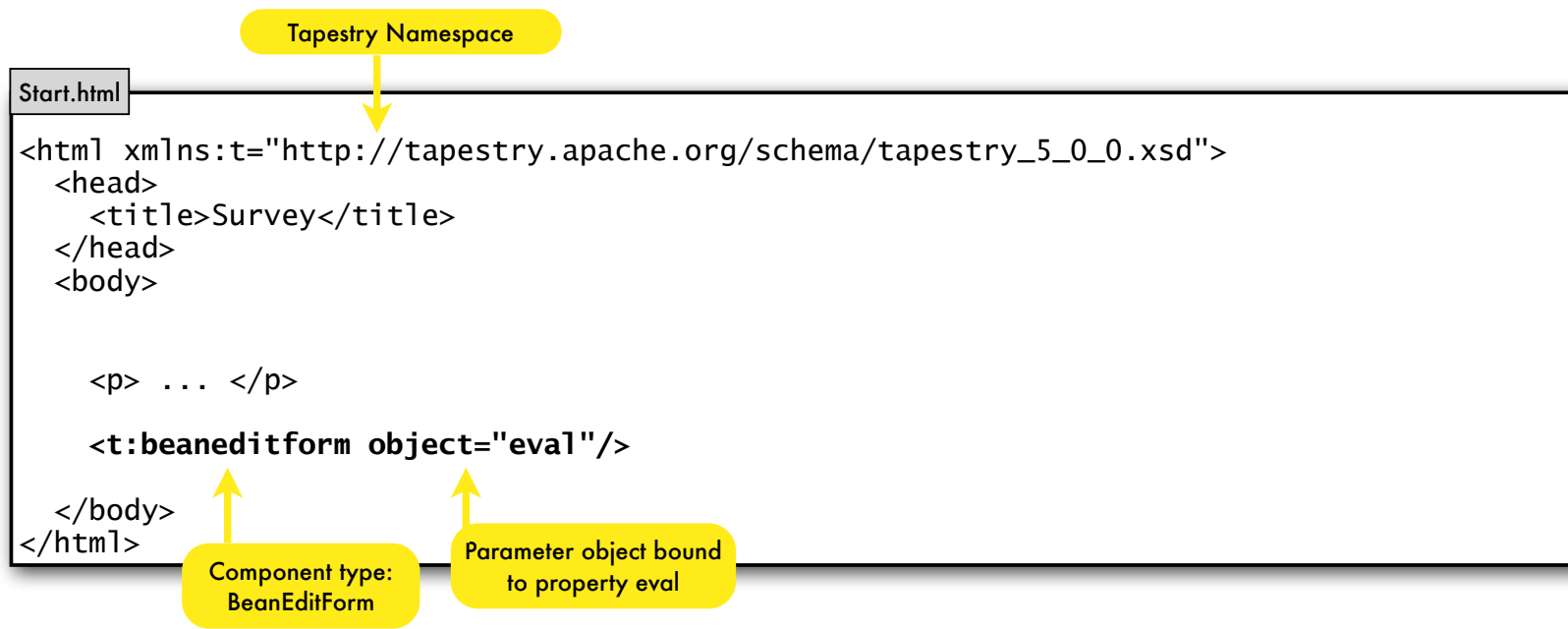
# DEMO #1

Basic BeanEditForm Component

# What did we just see?

---

- Tapestry application starts with a page named "Start"
- Start page template in WEB-INF/Start.html



# What did we just see?

---

Start.java

```
public class Start
{
    private SessionEval _eval;

    public SessionEval getEval()
    {
        return _eval;
    }

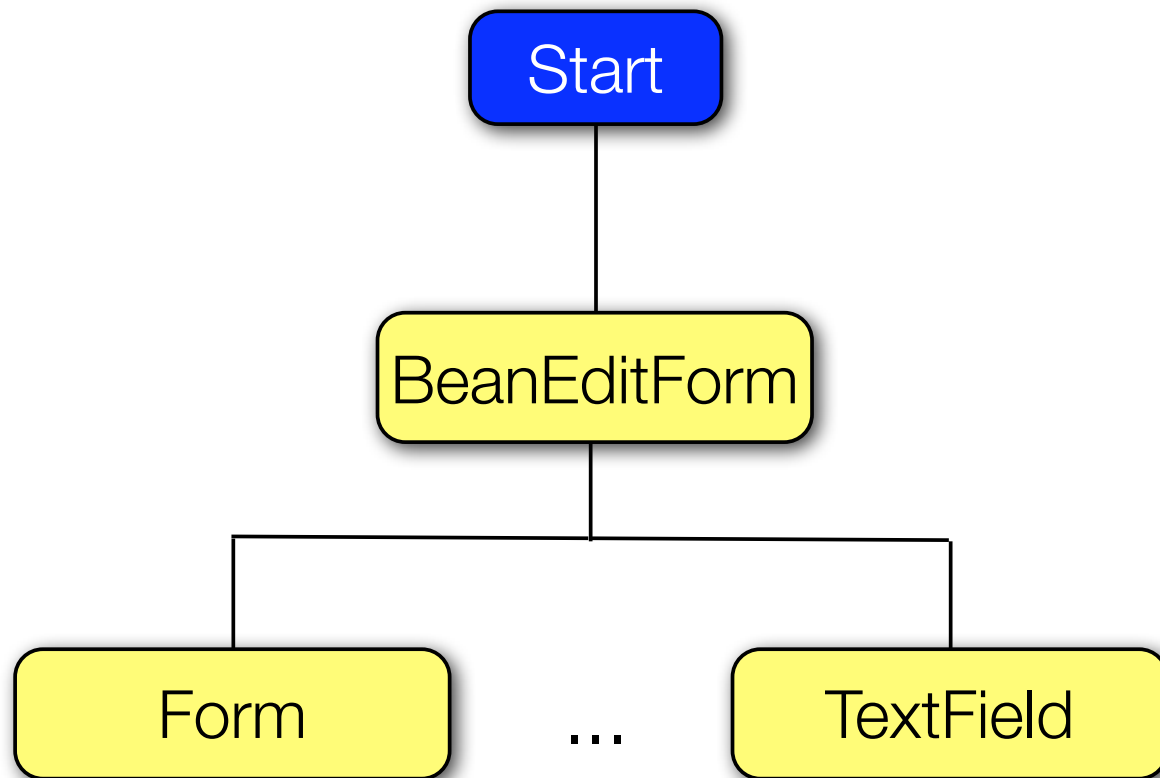
    public void setEval(SessionEval eval)
    {
        _eval = eval;
    }
}
```

Property bound to object parameter



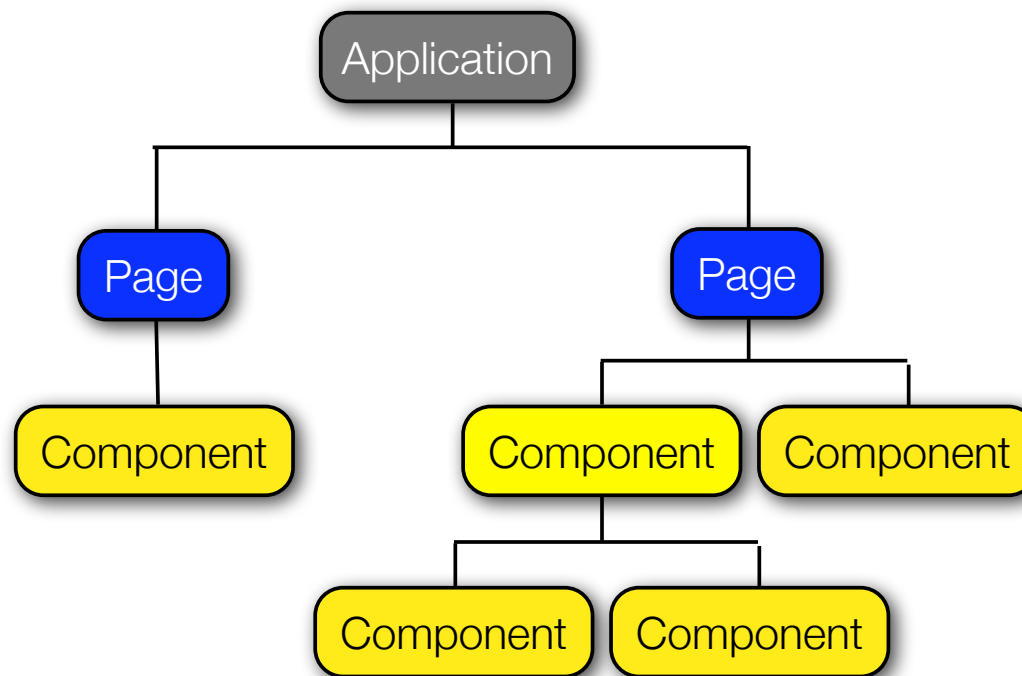
What did we just see?

---



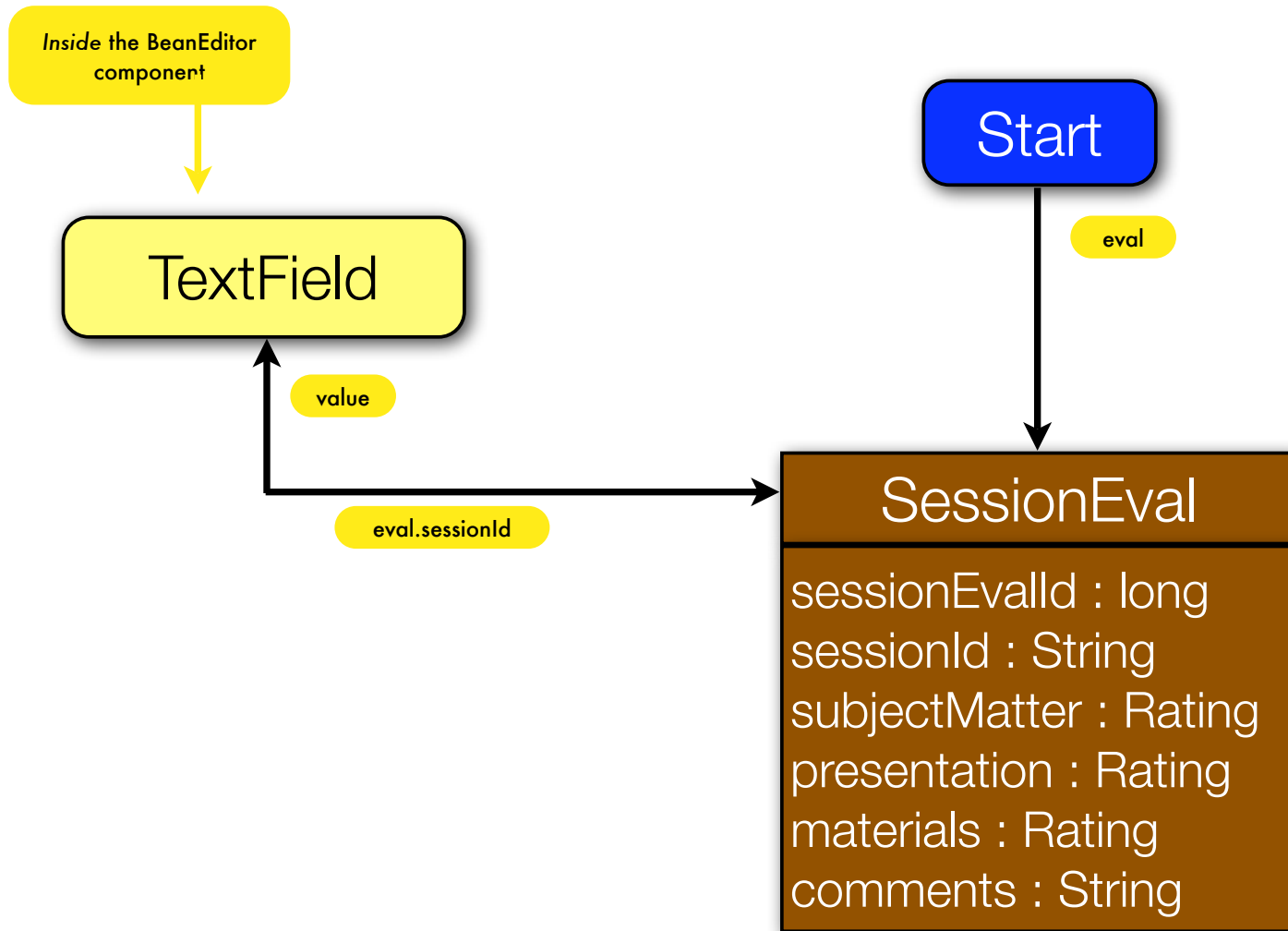
# Tapestry Applications

---



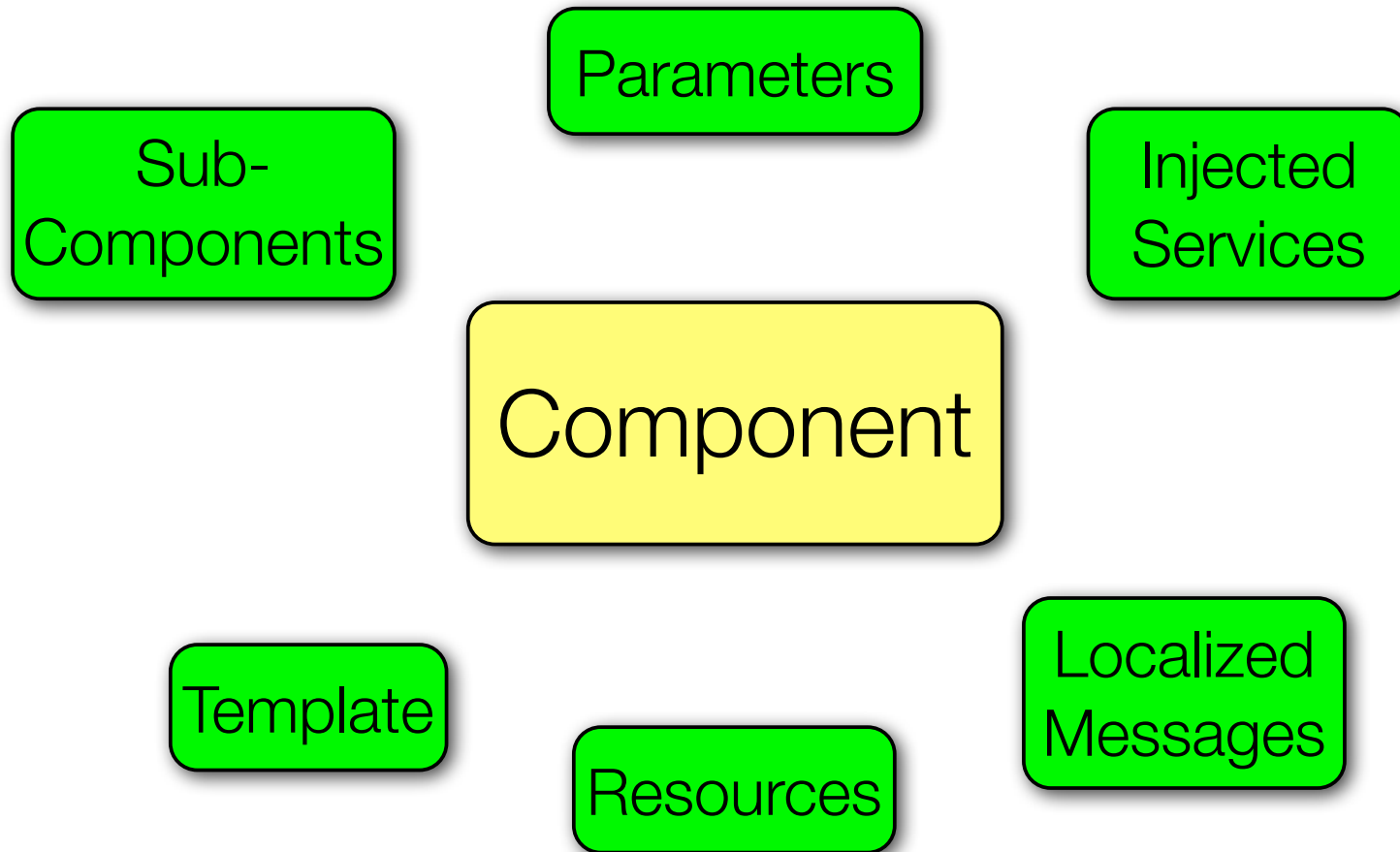
# Connected Objects

---



# Component 'n Stuff

---



# What Else?

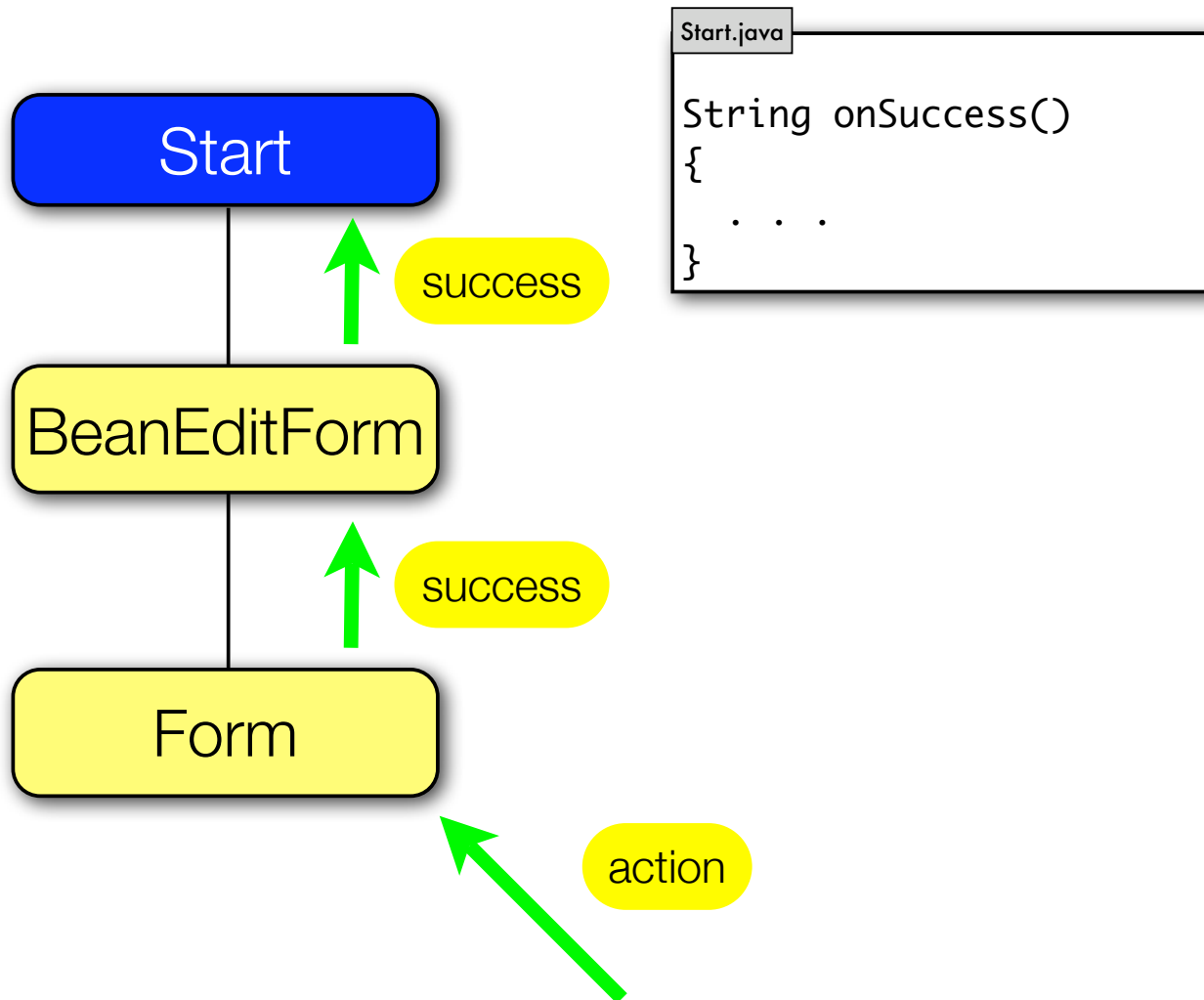
---

- Validation would be nice
  - Via `@Validation` annotation on bean property methods
- Customize the Form
  - Use a `TextArea` component for the comments field
  - Use a drop-down list for the session id

# DEMO #2

Adding validation and customizing the form

# Components & Events



# Tapestry and IoC

---

- Keep Business Logic out of the Page
- Inject Services from the IoC Container

Start.java

```
private SessionEval _eval;  
  
@Inject  
private SessionEvalDatabase _database;  
  
String onSuccess() Event Handler  
{  
    _database.add(_eval);  
    return "Thanks";  
}
```

# DEMO #3

Processing the form submission

# More Components

---

- **PageLink:** Link to another page
- **Grid:** Data table with paging & sorting
- **ActionLink:** Trigger an action event

# Grid Component

---

- Page navigation & sorting
- Lots of configurability
- Starts very simple

EvalList.html

```
<t:grid source="evals"/>
```

EvalList.java

```
public class EvalList
{
    @Inject
    private SessionEvalDatabase _database;

    public List<SessionEval> getEvals()
    {
        return _database.getSessionEvals();
    }
}
```

# DEMO #4

Linking to the EvalList page, adding the Grid

# Customizing Grid Output

---

- Customize number of rows per page with **rowsperpage** parameter
- Move pages around: **pagerposition** parameter
- Generates CSS class for each cell
- Class name from property name

Start.html

```
<style>
  TH.comments-header { background-color: black; }
  TD.comments-cell { color: silver; }
</style>
```

# Customizing Grid Output

---

- Special parameters to change cell renderer:

Start.html

```
<t:grid source="evals" row="eval"  
  rowsPerPage="10" pagerPosition="top">  
  
  <t:parameter name="commentscell">  
    <t:summarize value="eval.comments"/>  
  </t:parameter>  
  
</t:grid>
```

# Summarize Component

---

Summarize.java

```
public class Summarize
{
    @Parameter(required = true)
    private String _value;

    @Parameter
    private int _maxLength = 25;

    boolean beginRender(MarkupWriter writer)
    {
        if (_value != null)
        {
            String value = _value;

            if (value.length() > _maxLength)
                value = value.substring(0, _maxLength) + " ...";

            writer.write(value);
        }

        return false;
    }
}
```

# DEMO #5

Customizing Grid output

Is it ready yet?

---

# Still alpha!

---

- Stable
- Basic Spring / Hibernate integration
- Many features yet to be implemented
- Code stability has been great 5.0.1 → 5.0.5
  - Rethink on Templates
  - Rethink on Inversion of Control container
- Several people building production applications already!

# What's Coming

---

- Ajax Support
- Improved Hibernate integration
  - Request in Session Pattern
- Spring Web Flow Integration

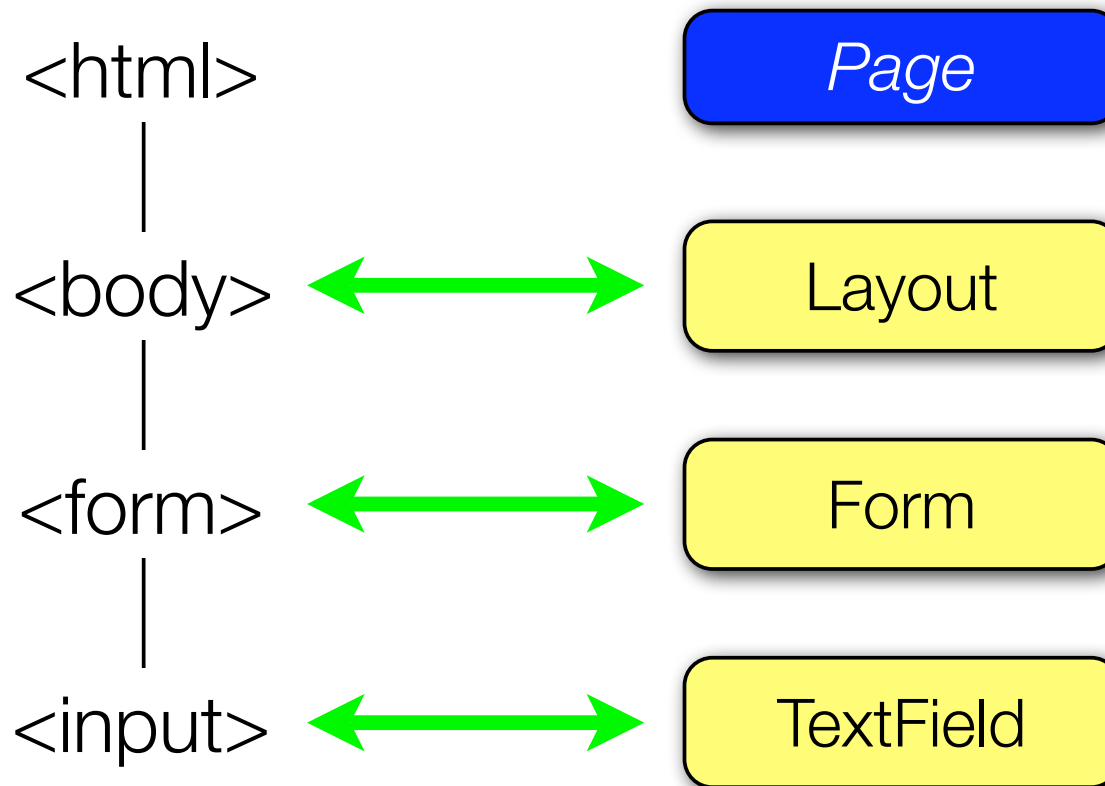
## IDE Support

---

“ I don't have time  
for Drag and Drop ”

# Ajax: Coming Soon!

---



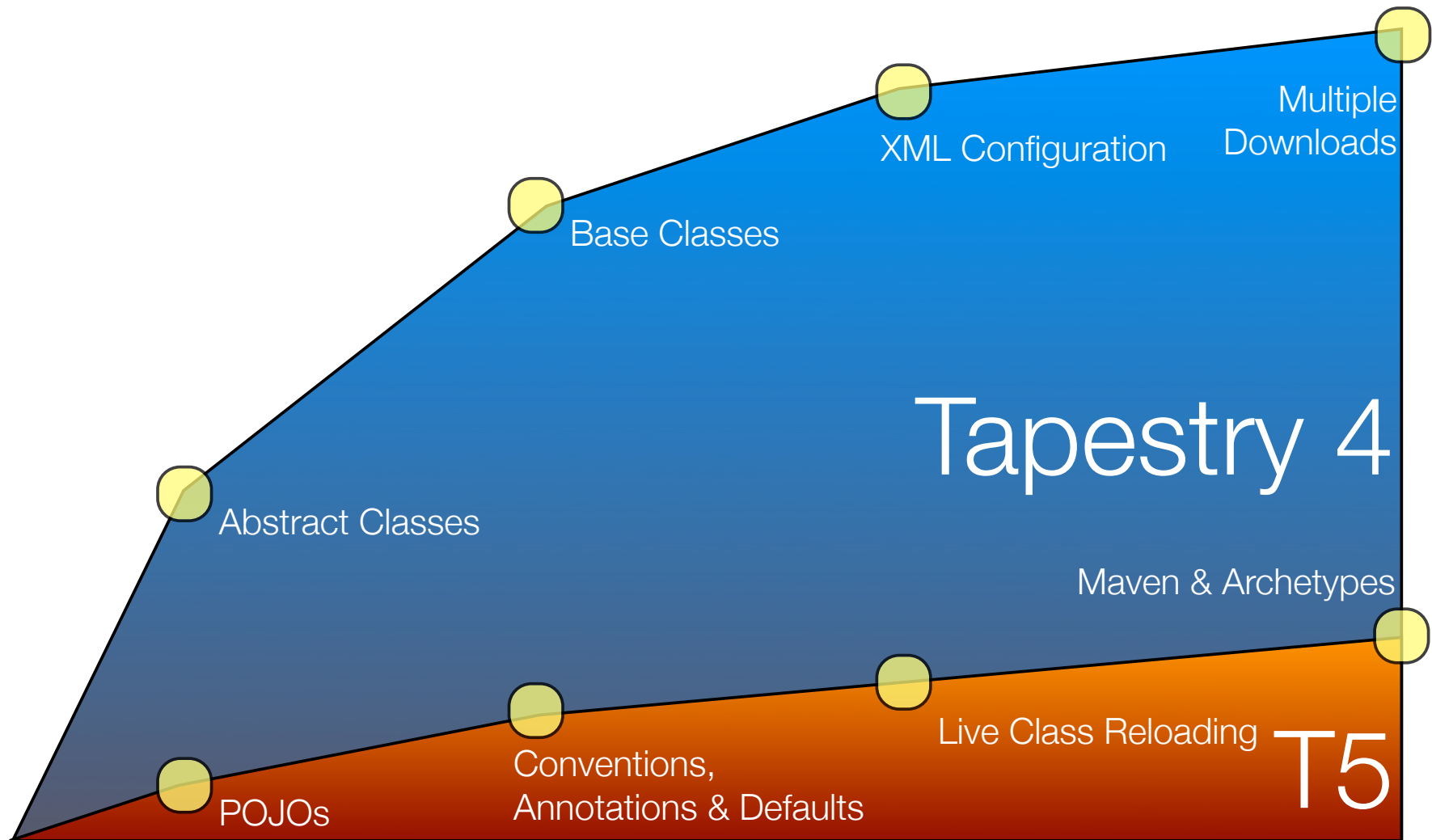
# Tapestry 4 ➡ Tapestry 5

---

- Clean Slate
- **Not** backwards compatible
- T4: Hard ➡ T5: Easy
- T4: Easy ➡ T5: Automatic

# Overcoming “Steep Learning Curve”

---



# More Information

---

- Home Page:

<http://tapestry.apache.org/tapestry5/>

- Dev & User Mailing Lists:

[dev-subscribe@tapestry.apache.org](mailto:dev-subscribe@tapestry.apache.org)  
[users-subscribe@tapestry.apache.org](mailto:users-subscribe@tapestry.apache.org)

- Howard's Blog:

<http://tapestryjava.blogspot.com/>

- Howard's Site:

<http://howardlewisship.com/>

# How you can help ...

---

- Give it a whirl!
- Blog!
- Contribute patches!
- JUG talks!

# Q & A

---