

Java 5 Support

We should consider moving to Java 5 which would provide a number of benefits including:

- type safe lists and maps in Click API
- ability introduce annotations for field/property autobinding

I don't think we have enough resources to support both Java 1.4 and 1.5 builds, so we could look at keeping the SF code base as the 1.4 line and moving the Apache code base to 1.5. I think this would provide motivation for people to make the upgrade from SF to Apache.

Autobinding

Re-examining this feature like it and I hate it at the same time. Why I like it, is because its quick to declare a public field/control in your Page class, and it will appear in your template. However after maintaining some large Click applications, this quick little hack can become a real problem. It is often not obvious when reading source code as to what is happening with all the autobinding black magic being performed. For instance you could have a public String field which is being autobound with a request parameter, or maybe its just used to display something in the template, and controls can also be auto bound, which can become very confusing if you have a HiddenField which is a public variable but is also added to a Form. If I had my time again I would not have included this feature with Java 1.4, its simply too much black magic and it has burnt me plenty of times.

Now Java 1.5 provides the opportunity to clean up this mess by adding annotations to Page fields and properties.

```
public class CrudPage extends Page {

    // Adds the control to the page
    @Bindable(BindingType.CONTROL)
    private Form form = new Form();

    // Adds the control to the page
    @Bindable(BindingType.CONTROL)
    private Table table = new Table();

    // Adds the value to the page model before the page template is rendered
    @Bindable(BindingType.MODEL)
    private String pageTitle;

    // Binds the request value "customerId" to this method performing type conversion
    @Bindable(BindingType.REQUEST)
    public void setCustomerId(Integer id) {
        CustomerDao customerDao = DaoFactory.getCustomerDao();
        Customer customer = customerDao.getCustomerForPK(id);
        form.copyFrom(customer);
    }
}
```

Thanks Fredrik Jonson for the BindingType enum suggestion.