

Form building considerations and hints

Table of contents

1 Form building and backstoring using XQuery + Cocoon.....	2
2 Form building and binding in Cocoon.....	2
2.1 Form definition vs available data.....	2

This document lists some of the problems and issues I've met while developing forms for editing the XML backend.

1. Form building and backstoring using XQuery + Cocoon

The XQuery extensions in eXist are making for a very powerful editing environment. The present editing solution does all the interaction between the db and the browser in XQuery, using Cocoon to manage the session and i18n/i10n of the forms. The session management is implemented via xmldb, thus using the registered users in the eXist db to control access to and permissions in the database.

2. Form building and binding in Cocoon

My original plan was to use the CForms binding facilities in Cocoon, to automatically bind the form to the XML db backend. That would have taken care of updates, synchronisation and session management in an easy-to-maintain and elegant way. Unfortunately the XML db binding was in practice not available, which would have required me to start learning and do some serious development in Java. So I opted for the solution above. It might be worthwhile to reconsider this option in the future.

2.1. Form definition vs available data

When filling in a form with data bound to an XML source, you have to make sure that the form definition only *requires* data you are absolutely sure will be there. If the requirements in the model and the data source are not agreeing in this respect, the binding process will crash (and the user left with a very suspicious impression). Letting a field be optional removes this problem.