

Hibernate Model/SQL

- Model class (with necessary relationship among tables)
 - Tool content
 - Tool session
 - Tool attachment
 - Tool users
 - Others
- Hibernate Xdoclet => Hibernate.hbm files + Tool table SQL script
- Insert tool default content in createToolTable.sql script
- Update SQL script in `${toolRoot}/db/sql`
 - lams_learning_activity
 - lams_learning_library
 - lams_tool

Service bean and DAO

- DAO class
 - HQL language
 - Initialized by spring and inject to tool service bean
- Tool service bean class
 - Spring managed bean class
- Spring (toolApplicationContext.xml)
 - messageServiceBean: for i18n
 - SessionFactory: hibernate session factory cover all hbm
 - Manage DAO beans
 - Inject DAO and lams core service (web.xml) into tool service bean
 - Transaction control for service methods

Action class/ servlet

- STRUTS action
 - Authoring
 - Learning
 - Monitoring
 - ClearSession
- Servlet
 - ExportPorfolio

Viewer

- JSP to create view page
 - JSTL tag
 - STRUTS HTML tag
 - Customized tag from lams, such as `<lams:AuthoringButton .../>`

More details in JSP pages

- Authoring page is most similar across all tools:
 - Use `<lams:head>` to include necessary style/JS
 - Use `<lams:Tabs>` to create 3 tab page
 - Use `<lams:AuthoringButton>` to create save/cancel button. It also redirect to success page and call `ClearSession` class automatically.
 - Use `<lams:FCKEditor>` to create standard and upload-file-allow FCKeditor
 - Suggest copy from existed tool:
 - Some Ajax for instruction file delete
 - Some JS and IFrame style for extra authoring items (such as forum topics)
- Use `<lams:Tabs>` for Monitoring
- Use `<lams:FCKEditor toolbarSet="Default-Learner">` to forbid file uploading. Most using in learner page if it need FCKEditor
- Other JSP pages :
 - Monitoring define it later page: one tab (basic) from authoring page
 - Learner define it later page: simply use `<lams:DefineLater/>` tag
 - Learner run offline page : provide run offline message and finish button

Other need care about

- Define it later
- Run offline
- Lock on finish
- Reflect on tool
 - Continue button to replace finish button on learner page
 - Inject lams core ICoreNotebookService
 - createNotebookEntry() for learner create reflection
 - getEntry() for monitoring read out reflection
- Learner page ToolAccessMode
 - Authoring preview
 - Learner mode
 - Monitoring read-only

Communication with LAMS core

- Service injection (web.xml and toolApplicatonContext.xml)
- Tool service class Implement interface from lams core:
 - ToolContentManager
 - copyToolContent : duplicate tool content
 - exportToolContent/importToolContent
 - ToolSessionManager
 - createToolSession: new record for tool session table

Web.xml configuration

- Web.xml
 - I18n configure for JSTL
 - Service import from lams core
 - Filter
 - SystemSessionFilter – provide shared session across multiple webapps
 - OpenSessionInViewFilter
 - Servlet
 - ExportPortfolio servlet
 - Tag definition
 - Security
 - Error page : 404, 403 etc

Ant

- Lams-build task
 - copyfiles: run once for JBOSS configuration
 - rebuild-db: reset lams-core tables. Need execute when developer want to redeploy tool
 - assemble-ear, deploy-ear: deploy lams core to JBOSS
 - deploy-tools: optional, deploy all available tools. For new tool, need manually add it to this tasks.
- Tool ant build.property
 - Tool signature
- Tool ant task
 - schemaexport: create *.hbm files and initial table script (need manually copy new script to deployable SQL script)
 - deploy-tool(explode-tool): deploy(explode) to JBOSS
 - synchronize-deploy: this task could synchronize JSP files if tool is exploded under JBOSS

Miscellaneous

- Use SessionMap instead of HttpSession to avoid multiple windows conflict
- Exception
 - STRUTS global-exceptions =>CustomStrutsExceptionHandler
- Rename tool to help you start from an exist tool
 - <http://wiki.lamsfoundation.org/display/lams/RenameTool>
- Recommend reading
 - <http://wiki.lamsfoundation.org/display/lams/Tool+Creation+Experience>