

Philly Grails Hack-a-Thon

Welcome to the Hack-a-thon. During this two-hour event you will be given time to create and experiment with your own application. Below are some thoughts to get you started.

Sample application: Time Entry

Develop an on-line time entry system. The system should allow for setup of projects, team members, and time entries for each user. Some use cases:

- Create a project, specifying the name, starting date, expected end date
- Add users to the timesheet system
- Associate users with one or more projects
- Allow users to create time entries for their projects
- Create reports of time by user, project

Alternative Sample Application: Conference Tracking

If time entry is not your thing (is it anyone's?) this is an alternative. Track conferences, tracks, sessions, speakers, attendance, etc...

Some use cases:

- Create conferences, specifying number of rooms, date range, cost
- Create conference rooms (Hall 1A, etc)
- Define conference sessions
- Define speakers and associate with sessions, rooms
- Render a conference agenda based on data entered in the various domains

This one can get very fancy. For example, you could drive an entire conference site, including registration, speaker and talk admin, and rendering of the public content, all from a Grails application. You can experiment with rich text editing plug-ins (take a look at the fck editor for example) to define abstracts in HTML styling. Essentially it is a content management system hiding in a Grails application.

For inspiration, download the grails.org application from the grails.org website.

<http://github.com/grails/grails/tree/master/grails-samples/grails.org/>

Suggested approaches / ideas

- Start by building the domain model, and scaffold controllers for each domain. You can do this by creating a blank controller, and then replacing the body of the controller with `"def scaffold = domainclassname"`.

Philly Grails Hack-a-Thon

- Experiment with constraints and relationships. To keep your data between restarts, set the dbCreate parameter in the BootStrap.groovy file to 'update' rather than 'create-drop'
- Generate your controllers and views as needed. The generation process (grails generate-all, generate-views, generate-controller) creates physical controller and view objects and then lets you modify them to suit your needs. For example, create a field with several choices (low/med/high). Then, generate your controller/views and edit the edit.gsp file, and experiment with the various ui widgets (drop down, radio buttons, etc).
- Use the Grails Layouts to organize your user interface. The layouts are referred to in each view GSP via the meta tag 'layout'. The default layout (used by the grails generators) is 'main', and can be found in grails-app/layouts/main.gsp.
- Install the Spring Security plugin and experiment with authentication. Details on configuration and usage can be found on the plugin page.
- Always build a useful toString() method for your domain objects. These toString() methods are called when building drop-downs in the default scaffolding process. Be careful not to dig into relationships within the toString() or you may be fetching more than you bargained for.
- Consider adding AJAX to your application to spice up the process. For example, when entering time, show your 'entered time' as a table of information. When you add your time in the data entry form, consider using an AJAX form to post the time, then update the DIV in place with the resulting new table of time entries.
- Install the grails templates - 'grails install-templates' and modify the default scaffolding to add your own features. Then issue 'grails generate-all <domain class>'
- If you have Tomcat on your system, try generating a WAR file using 'grails war' and copying the resulting war into tomcat's webapps directory. Deployment is relatively easy with Grails. To deploy the production environment, use **grails war** to generate the WAR file, then drop it into tomcat.

•
Most of all, have fun, experiment, and thank you for attending the Philly Grails seminar!