Collaborative Development on the LongJump Platform

February 27, 2014

1230 Midas Way, Sunnyvale, CA 94085

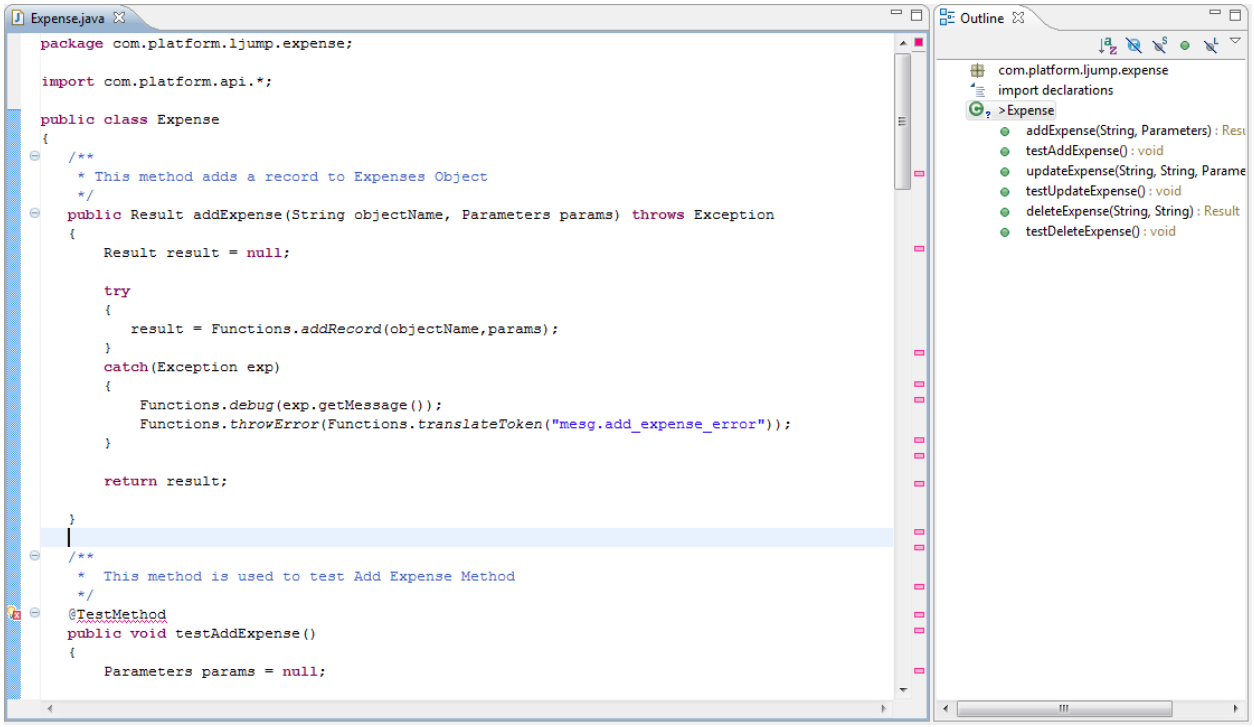
www.LongJump.com

Contents

1 Basic Development Model

1.1 Class and Pages Development

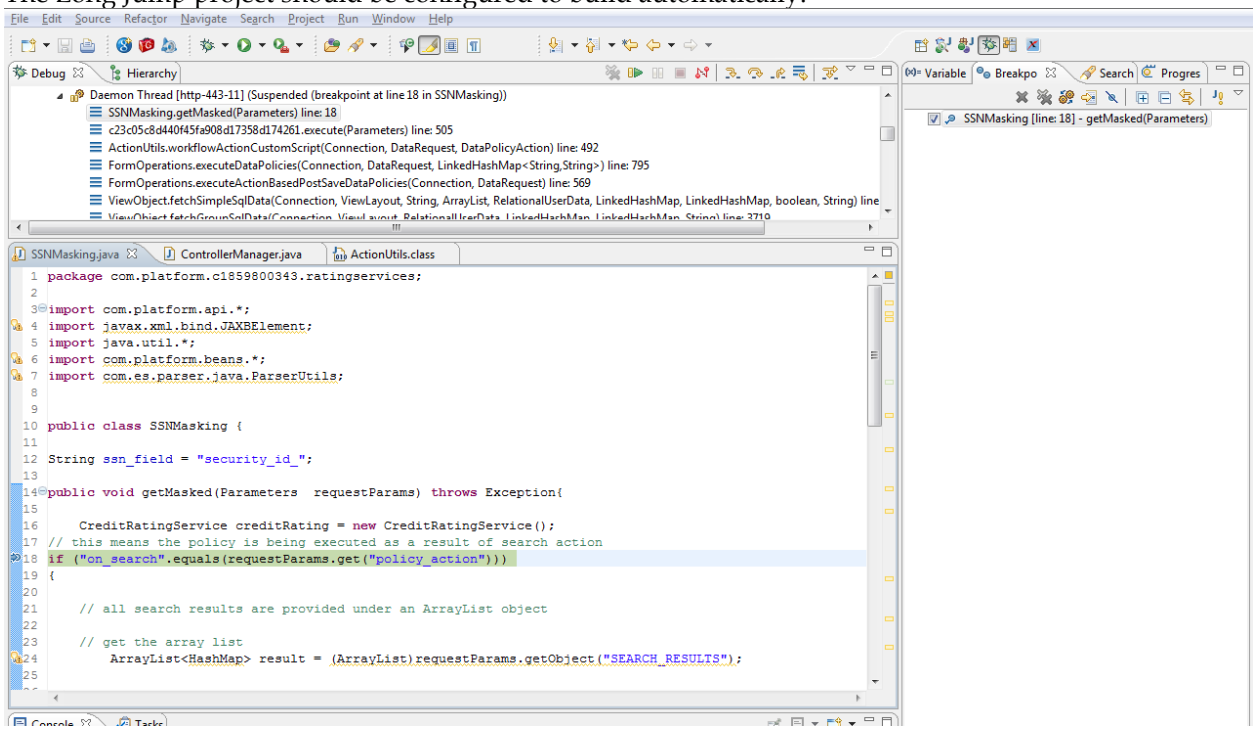
Developers can use the LongJump Eclipse plug-in to develop Classes and Pages:



1.2 Debugging

Long Jump allows classes and pages to be debugged using Eclipse Plugin in development mode. Standard debugging configuration in Eclipse is used for this purpose.

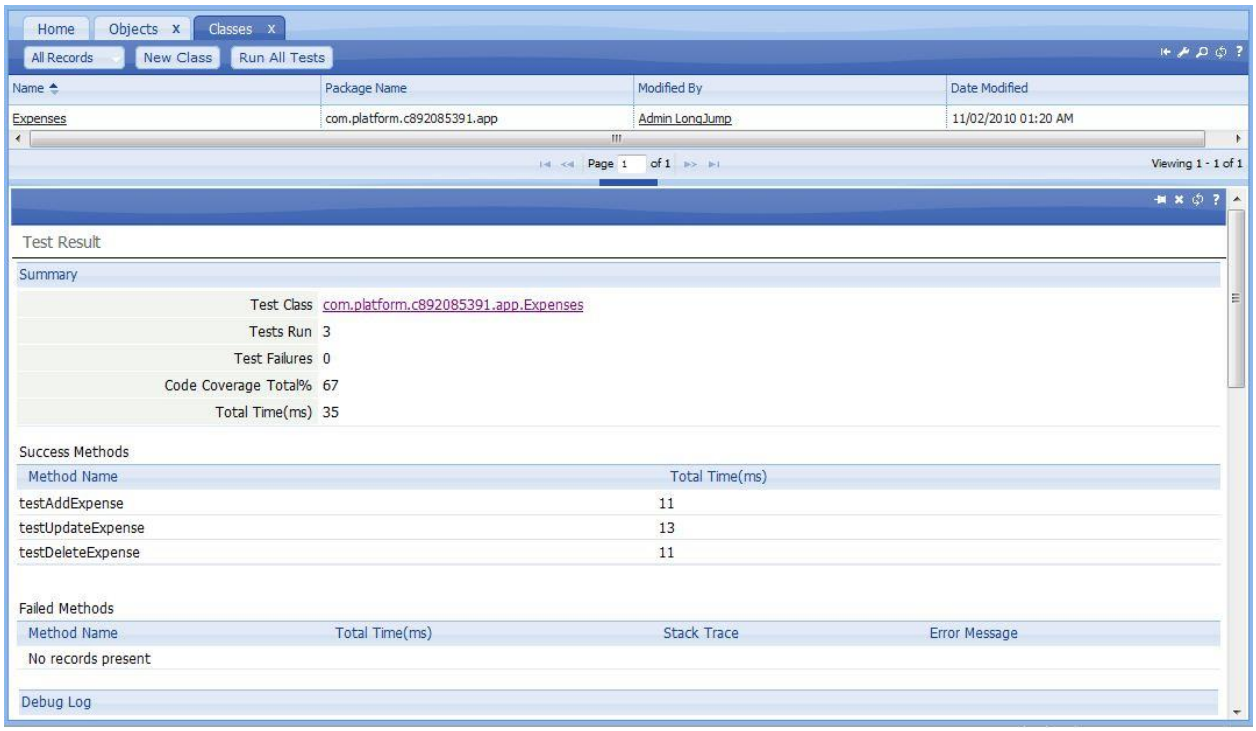
The Long Jump project should be configured to build automatically.



Note:
Sometimes the code optimization done by Eclipse is different than the optimization done by the JVM in tomcat. Source code linking issues can result. To work around that problem, make a small change to the source code while debugging (for example, by adding/removing a space) and save. The Eclipse debugger will immediately link to the source code correctly, allowing developers to step through the code as well as hot-deploy any changes.

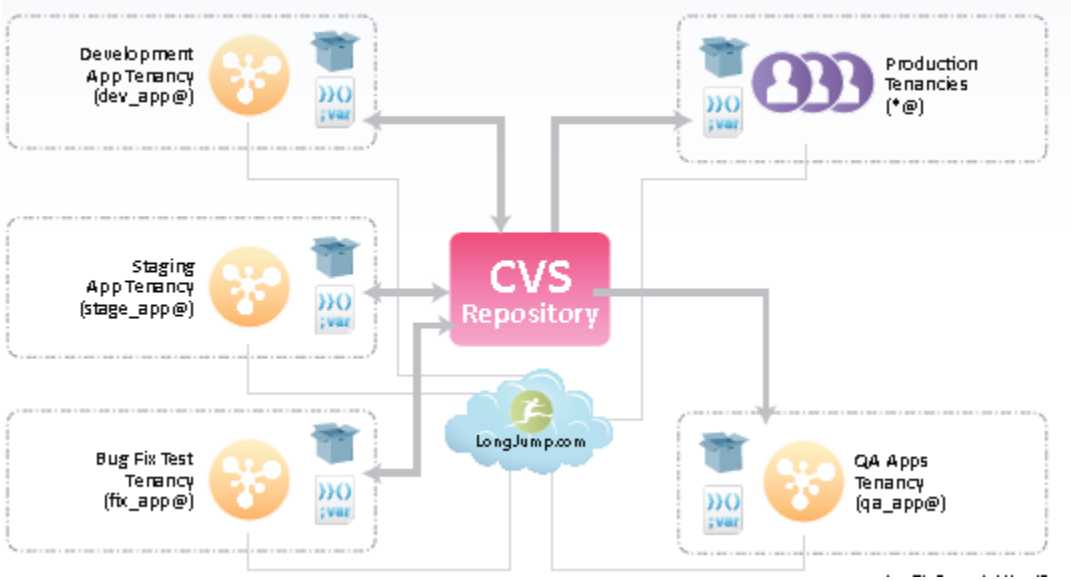
1.3 Unit Testing

Developers can use the integrated unit testing environment in LongJump to create and launch unit tests. Developers are be provided unit testing report as well as overall code coverage.



2 Cloud Development Model

In the Cloud Development model all the developers are located on the same tenancy in the LongJump cloud. They all have User IDs in a shared development environment. They have the option of turning on a light-weight version control system that runs a checkout-and-commit process in the shared environment. This process prevents simultaneous modification by multiple developers of same resources. It keeps a resource locked, until the developer who has checkout the resource commits it to the repository.



After developers have created the first version of their application, they create a package containing the application and publish it to the QA tenancy. After the QA has certified the package it can be promoted to staging and production tenancies or published into the catalog. Developers also have the option to download the package ZIP file and check it into their organization's version control repository.

3 Deployment

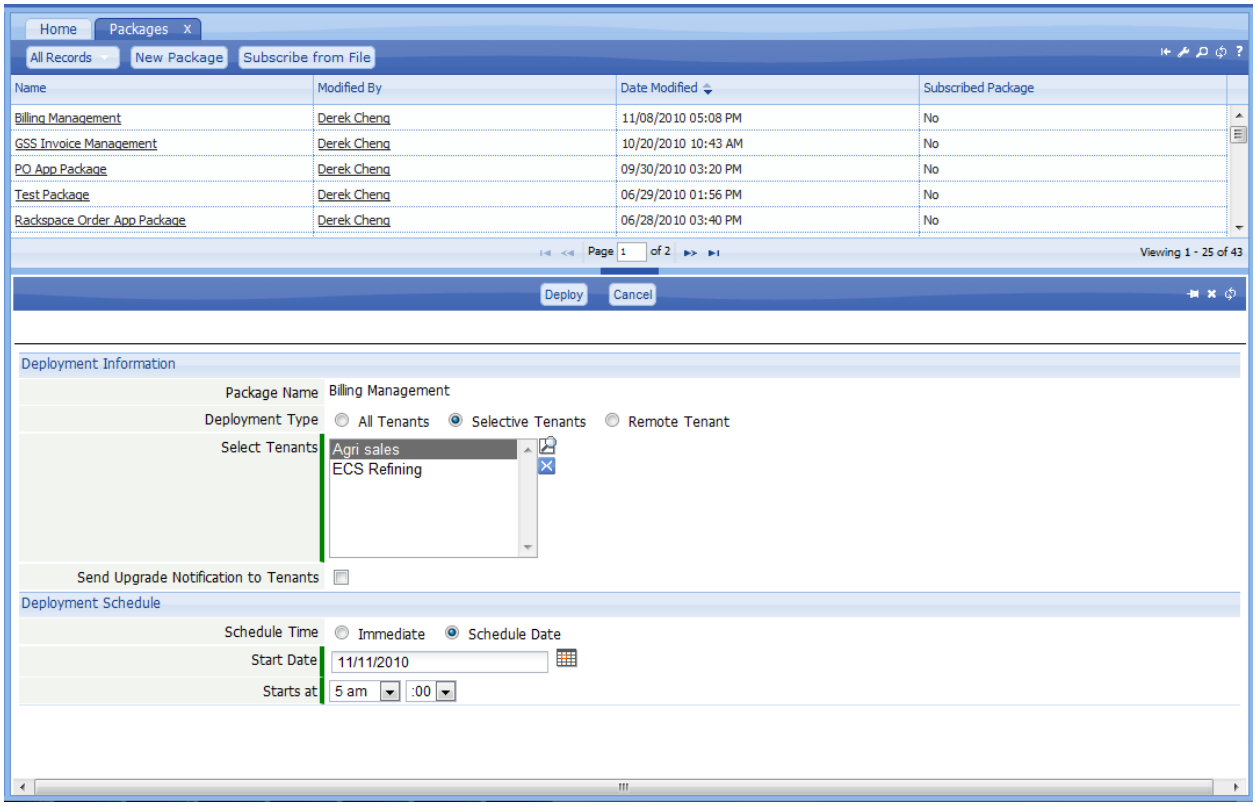
3.1 Publishing to Catalog

After a fully tested package has been loaded into the production tenancy, the developer has an option to use the LongJump application-listing management system to create a listing request. As part of the listing request, developer can set up pricing, demo videos, images, and test-drive dummy data, as well as specify terms and conditions of use. Once the request has been approved, the application appears in the market place.

3.2 Deployment to Tenants

Once a package has been published into the production instance, the developers with the proper rights have the option to automatically deploy the package to tenants either using Deployment REST APIs or through the LongJump User Interface. Using either of the methods, the operation personnel have a choice to do to the deployment immediately or at a scheduled time in the future.

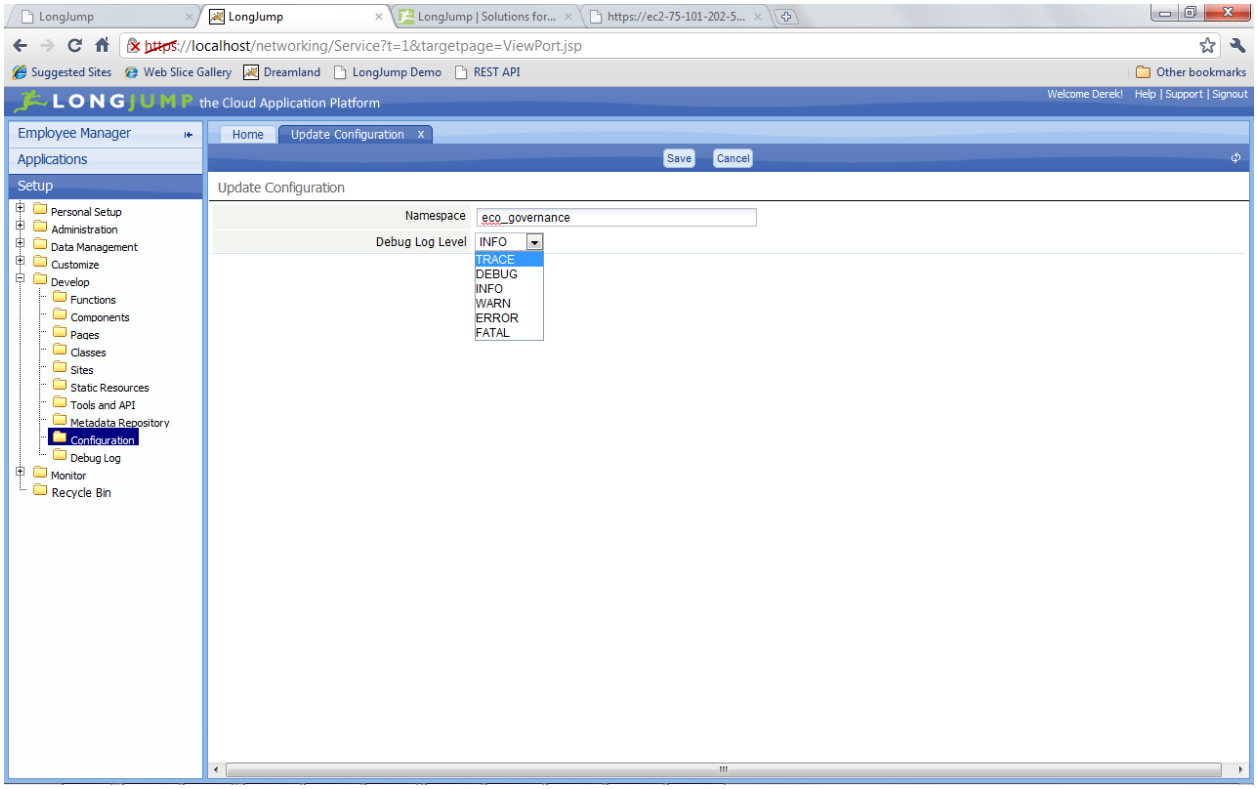
The following screen capture shows the UI for deploying to multiple tenants:



REST APIs are also available to submit the deployment job through the API. The API Call returns a Job ID using which can used to check the status of the job. At the end of the deployment process, the system sends summary deployment information to the job submitter.

4 Tracing Issues in a Production Environment

Once Applications have been deployed into customer tenancies, it is possible for developers to "proxy-login" to those tenancies using the built-in Long Jump Customer Support system. Once there, they can enable a variety of debug-log levels.



5 Change Control

LongJump can be configured to allow application change rights for authorized system roles. For example, some roles might be able to add Java classes, some may only be able to add or customize layouts.

6 Sarbanes-Oxley Act Compliance

Several aspects of SOX compliance are present in LongJump.

1. Adherence to current release management processes.
   Because LongJump is configured with separate tenancies for development, staging and production, the release management processes can match those used for traditional software release management. Code can be promoted from one instance to the next using APIs or the LongJump User Interface.

2. Separation of Duties.
   Because each instance of LongJump has fine grained user access controls, different and separate roles and responsibilities can be assigned to people developing software, performing quality assurance or people involved in release management. This helps companies adhere to a key SOX control objective.

3. Control Audits.
   LongJump keeps a complete audit log of all system activities and changes. This level of transparency provides an automated control for SOX compliance. In addition to the system log, field change logging is present (past and current values), and a complete record change tracking is present with optional record digital signatures.