

# 1. Release Notes

## 1.1. Release notes for version 0.41

### 1.1.1. What's new since version 0.40

- Show a custom error page and provide the infrastructure to show ajax errors.

### 1.1.2. Resolved issues since version 0.40

- Katari was sending image/jpg instead of image/jpeg for the content type of jpeg files.

## 1.2. Release notes for version 0.40

### 1.2.1. What's new since version 0.39

- Fine grain permission configuration for non modules urls. Now you can add additional url-role mappings to urls that are outside modules.
- Fixed some double slashes when the application is deployed in the root context.

## 1.3. Release notes for version 0.39

### 1.3.1. What's new since version 0.38

- JsonRepresentation can now be rendered directly as json in freemarker.
- Added i18n section to the user manual.
- Bug fixed: If one menu item was a prefix of a sibling menu item (ex: Twitter\_Accounts and Twitter\_Account), katari selected both menus.

## 1.4. Release notes for version 0.38

This release just fixed a bug in menus: when a menu entry link has request parameters, the menu generated a link with an unescaped &, which is invalid html.

## 1.5. Release notes for version 0.37

This release continues adding internationalization related features.

### 1.5.1. What's new since version 0.36

- It is now possible to translate menus, and override their translations.
- Added more control on the selection of the client locale.
- The weblet instance can now be any object, instead of only a string.

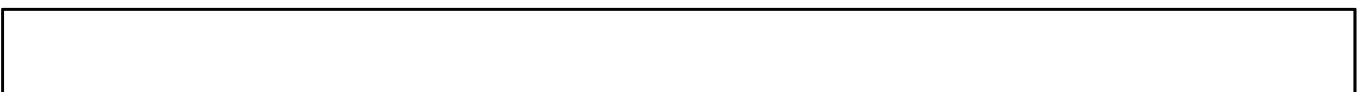
## 1.6. Release notes for version 0.36

### 1.6.1. What's new since version 0.35

- It is now possible to override the translations for each module, and to add additional translations for each module.
- Renamed the request parameter to select a language from katari-lang to lang.
- Upgraded htmlunit from 2.5 to 2.9.
- Upgraded http components from 4.0.3 to 4.1.2
- Spring 3 uses a freemarker class that has a bug in 2.3.16. Upgraded to 2.3.18, that fixes that bug.
- Spring mvc 3 changed the way to handle exceptions. If an exception is not handled, it is swallowed. Added a new DispatcherServlet that restores the throws the exception up. You must handle your exceptions in a filter.

## 1.7. Upgrading from 0.35

- The new version of htmlunit depends on http components instead of commons-httpclient. So commons-httpclient is no longer a dependency of katari. This makes PingTest (as generated by old katari archetypes) fail. You must upgrade your test to http components. This is a partial diff to guide you (excluding the imports):



```

@@ -16,13 +18,14 @@ public class PingTest {
    @Test
    public final void testPing() throws Exception {

-   HttpMethod method = null;
+   HttpClient client = null;
    try {
-   HttpClient client = new HttpClient();
-   method = new GetMethod(SimplePageVerifier.getBaseUrl() + "/ping");
+   HttpGet method = null;
+   client = new DefaultHttpClient();
+   method = new HttpGet(SimplePageVerifier.getBaseUrl() + "/ping");
+   HttpResponse response = client.execute(method);
+   String responseBody = EntityUtils.toString(response.getEntity());

-   client.executeMethod(method);
-   String responseBody = method.getResponseBodyAsString();
    assertTrue("Response doesn't match 'Loading spring context: SUCCESS'",
        responseBody.matches("(?s).*Loading spring context: SUCCESS.*"));
    assertTrue("Response doesn't match 'Application started successfully'",
@@ -32,7 +35,9 @@ public class PingTest {
        assertTrue("Response doesn't match 'this is a development database'",
            responseBody.matches("(?s).*this is a development database.*"));
    } finally {
-   method.releaseConnection();
+   if (client != null) {
+   client.getConnectionManager().shutdown();
+   }
    }
}

```

## 1.8. Release notes for version 0.35

### 1.8.1. What's new since version 0.34

This is the first release with internationalization features. I18 is still not complete.

- Translated style (the module and the archetype) and local-login modules to spanish.
- Upgraded spring from 2.5.6 to 3.0.5 and easymock from 2.2 to 3.0.
- Support unauthenticated smtp in katari-email.

## 1.9. Upgrading from 0.34

- The style module (as generated by the archetype) now needs a message source injected to the FreemarkerDecoratorServlet. Add this to your style's module.xml:

```
<!-- Message source for this module, loaded from localized "messages_xx"
files.
Note: as this message source is not named messageSource, it is not used as
the default message source for the application context. -->
<bean id="decorator.messageSource"
  class="com.globant.katari.core.spring.KatariMessageSource"
  p:basename="classpath:/com/globant/katari/style/l10n/messages"
  p:debugPrefix='../katari-style/src/main/resources'
  p:debug-ref='debugMode'/>
```

## 1.10. And modify the FreemarkerDecoratorServlet configuration:

```
<bean class="com.globant.katari.core.sitemesh.FreemarkerDecoratorServlet">
  <constructor-arg index='0' ref='decorator.messageSource'/>
</bean>
```

Note also that the bean named 'decorator.messageSource' of the previous sample uses the 'p' namespace that was not added to the style module.xml by the archetype. Add this line:

```
<p:debug-ref='debugMode'/>
```

xmlns:p="http://www.springframework.org/schema/p"

to the beans xmlns declaration.

In this release, to be able to use `@spring.message` freemarker macro in your decorators, you must add

```
;class://org/springframework/web/servlet/view/freemarker
```

and

```
../katari-core/src/main/resources;
```

to the `AdditionalTemplatePaths` and `AdditionalDebugPrefixes` parameters of the `FreemarkerDecoratorServlet` (this will be needed until katari provides its own freemarker related macros.)

Finally, add your `messages.properties` in the directory specified in your `decorator.messageSource` bean.

## 1.11. Release notes for version 0.34

### 1.11.1. What's new since version 0.33

This a bug fix only release:

- Show an error message if the change password token is invalid.
- The title of the edit or create user page is now Create User for new users.
- Dropdown menu was not shown in ie7.
- User management was not working for ids over 1000.
- The registration module could not be configured.

## 1.12. Release notes for version 0.33

### 1.12.1. What's new since version 0.32

This release only includes one minor fix:

- Redirect the user after changing his password. This updates the decorated pages to reflect the fact that the user is logged in.

## **1.13. Release notes for version 0.32**

### **1.13.1. What's new since version 0.31**

This release only includes one minor fix:

- The forgot password and registration emails hardcoded the base url.

## **1.14. Release notes for version 0.31**

### **1.14.1. What's new since version 0.30**

This release only includes minor fixes and enhancements:

- Added a css class to menu items to identify them.
- Minor katari-user refactor to support delete event vetoes.
- Stop supporting taglibs in freemarker, it conflicted with weblets.
- Improved katari-email documentation.

## **1.15. Release notes for version 0.30**

### **1.15.1. What's new since version 0.29**

- New module katari-monitoring.
- Make tomcat ports externally configurable, to run multiple builds in parallel.
- It is now possible to have multiple instances of a katari based application running in the same tomcat.
- Make the sql parser more robust.
- Removed the dependency on apache commons logging.

## 1.15.2. Resolved issues since version 0.29

- The generated archetype now truncates the database owner to 14 chars. Previous versions generated database usernames that were too long if the project name was long enough.
- Fix the random resource loading error in the browser.
- Make sql parser more robust.

## 1.16. Upgrading from 0.29

- The development database checker should no longer be overridden to configure the marker table. Replace the `katari.developmentDataBaseChecker` bean in your `applicationContext` with:

```
<bean class='com.globant.katari.core.spring.StringHolder'  
  name = 'katari.markerTable' p:value = 'katari_marker_666' />
```

## 1.17. Release notes for version 0.29

### 1.17.1. What's new since version 0.28

- Make menus more customizable.
- Refactored the gadget container.

## 1.18. Upgrading from 0.28

### 1.18.1. If you are using the gadget container:

- Change the endpoint to obtain the gadget group from `socialPage.do` to `getGadgetGroup.do`.
- Change the overridden bean named `gadgetcontainer.viewGroupDecider` to `gadgetcontainer.viewOwnerRestriction`.

## 1.19. Release notes for version 0.28

## 1.19.1. What's new since version 0.27

- Support configurable views in the gadget container.

## 1.20. Upgrading from 0.27

### 1.20.1. This version updated the gadget container module database schema:

- The table `gadget_groups` has a new column `view_name`, a `varchar` 255 characters long. You need to add it to your database and set it to the view name of the group.
- There is a new table `supported_views`. Create it with:

```
create table supported_views (  
  application_id bigint not null,  
  view_name varchar(255) not null  
)  
alter table supported_views  
  add index (application_id),  
  add constraint  
  foreign key (application_id)  
  references applications (id);
```

Then add the supported views to all your applications.

## 1.21. Release notes for version 0.27

This is a bug-fix only release.

### 1.21.1. What's new since version 0.26

- Html validation error when the menu was open.

## 1.22. Release notes for version 0.26

## 1.22.1. What's new since version 0.25

- New module katari-dropdown-menu.
- Removed the analytics features from katari-shinding. They were using external scripts.
- New katari:null spring extension to define null beans.
- Removed katari-cas module. This was not maintained for a long time.
- Added extension points to the katari Activity service, so that application writers can influence what activities are returned to the gadgets.

## 1.23. Release notes for version 0.25

### 1.23.1. What's new since version 0.24

- New module katari-email.
- New module katari-registration.
- New module katari-cas-login.
- Camel integration to decouple modules through events.
- New simpler way to add a quartz job.
- The quartz console now lets you manually start jobs.
- Upgraded shindig to version 2.0.2, jasperreports to 3.5.3 and quartz to 1.6.3.
- User name and email are now unique.
- Various bug fixes.

## 1.24. Upgrading from 0.24

- The new shindig version now needs to know the host and port where the application is running. You need to add:

```
<bean id='shindig.hostAndPort'  
  class='com.globant.katari.core.spring.StringHolder'  
  p:value='localhost:8088' />
```

to your applicationContext.xml file.

- You need to add a new column in the user table: is\_active, a not null 1 bit column.

## 1.25. Release notes for version 0.24

## 1.25.1. What's new since version 0.23

- New module katari-jsmodule. Supports katari packaged js libraries.
- New module katari-quartz. This module makes it possible for other modules to schedule tasks sharing a quartz instance. It also provides a simple monitoring panel. Thanks Emiliano Arango for this.
- Various bug fixes, most from the project archetype.
- More information shown in the ping service (memory, debug configuration).

## 1.26. Release notes for version 0.23

### 1.26.1. What's new since version 0.22

- New module katari-user. This was extracted from the sample web app. Now, the application generated by the archetype no longer includes the user operations.
- The gadget dashboard can now be configured by adding and removing gadgets from a directory.
- Many improvements to the project generated by the archetypes.

## 1.27. Release notes for version 0.22

### 1.27.1. What's new since version 0.21

- Modules can be disabled by a configuration option.
- Two new modules to support social applications.
- Now the user is a core katari entity. This makes it possible for modules to have direct references to user objects.
- Removed the trails module from katari.

### 1.27.2. Upgrading from version 0.21

- Your user implementation now needs to extend from Katari's CoreUser.
- Your DomainUserDetails implementation now needs to extend from Katari's CoreUserDetails.
- If you want to use the social modules, make sure to use an updated version of commons-lang. Previous version of katari used version 2.1. Upgrade to version 2.5. Otherwise, you will get `java.lang.NoSuchMethodError: org.apache.commons.lang.StringUtils.join( ...`

## 1.28. Release notes for version 0.21

### 1.28.1. What's new since version 0.20

- New archetype to scaffold a new module.

### 1.28.2. Upgrading from version 0.20

- If you overrode `katari.httpSessionContextIntegrationFilter`, you now need to override `katari.contextHolderIntegrationFilter`.

## 1.29. Release notes for version 0.20

### 1.29.1. What's new since version 0.19

- New search module.
- New console module.
- Katari is now compatible with spring 3.0. It still uses spring 2.5.
- Documentation improvements.
- The maven generated site now looks like the sample katari application.

## 1.30. Release notes for version 0.19

The project has been renamed to Katari. This is the first release under the new name.

### 1.30.1. What's new since version 0.18

- Improved the editable pages module: rollback and per page preview.
- Minor fixes to the captcha feature.
- New logo to match the new name.

### 1.30.2. Upgrading from version 0.18

Renaming to katari involved changing the java package names and the maven artifact name. To upgrade from version 0.18 just change anubis for katari in package name, artifactId and groupId.

### 1.30.3. What's new since version 0.17

- Katari now integrates wicket.
- Added support for captcha in the login page.
- The editable pages module now provides a preview and a revert mechanism.
- New ui for the sample applications.
- Added https support.
- Changed the way modules registers persistent classes.
- Updated yui from 2.6 to 2.8.0r4.
- Updated hibernate and selenium dependencies.

### 1.30.4. Upgrading from version 0.17

Persistent classes are no longer registered using persistentClasses property in the ConfigurableModule class (this operation was removed.) You now must append your persistent classes to a special bean in spring named anubis.persistentClasses. Replace, in all your modules, fragments like:

```
<property name='persistentClasses'>
  <list>
    <value>com.globant.katari.sample.user.domain.User</value>
  </list>
</property>
```

with:

```
<bean class='com.globant.katari.core.web.ListFactoryAppender'>
  <constructor-arg value='anubis.persistentClasses' />
  <constructor-arg>
    <list>
      <value>com.globant.katari.sample.user.domain.User</value>
    </list>
  </constructor-arg>
</bean>
```

This version has updated a couple of dependencies, mainly related to hibernate and selenium:

- Change the following dependencies:
  - hibernate to hibernate-core-3.3.2.GA.
  - update hibernate-annotations to version 3.4.0.GA.
  - update hibernate-validator 3.1.0.GA.
  - remove hibernate-commons-annotation.
  - jboss.javassist to javassist.javassist-3.8.0.GA.
  - org.openqa.selenium.client-drivers.selenium-java-client-driver to org.seleniumhq.selenium.client-drivers.selenium-java-client-driver-1.0.
- Remove the selenium-server dependency from the selenium-maven-plugin configuration in your web app module.
- If you are using yui loader, replace yuiloader-beta-min.js for yuiloader-min.js.

### **1.30.5. What's new since version 0.16.1**

- Katari now has an archetype to generate a project. This lowers the barrier for new anubis developers.
- Lots of improvements in the editable pages module.

### **1.30.6. What's new since version 0.16**

- It is now possible to override the location of the message resource of the trails module. This is a simple workaround until we find a method of merging listis in spring.
- Improved html validation error message.

### **1.30.7. What's new since version 0.15**

- There is a new module katari-editablepages that allows users with the appropriate privileges to edit web pages with a web based WYSIWYG editor.
- Changes to freemarker templates loaded from spring mvc are immediately picked by the browser. It is no longer necessary to restart the container to see the result of a change in freemarker templates. The same holds for static content like css and javascript files.
- Now anubis incorporates a filter that automatically validates all html content sent to the client. This makes it easier to generate valid html content.

### **1.30.8. Resolved issues since version 0.15**

- Some pages generated invalid html content.
- Fixed the random fail in trails pages when using jetty. This was a long standing problem only manifestating in jetty.

## 1.30.9. What's new since version 0.14.1

- The local login module now shows a message if the user entered invalid credentials (merged from 0.14.2).
- Split anubis core in two, one with the core related to the webapp and spring integration, and the other with hibernate support. Now you can have an anubis application without a database.
- Improved the documentation, adding a getting started section in the user manual.
- Upgraded htmlunit to version 2.5 and cargo plugin to version 1.0.

## 1.30.10. Resolved issues since version 0.14

- Weblet rendering was ignoring the encoding.
- Due to an unexpected change in doxia, pdf documents where generated with the images not properly scaled.
- The menu module was showing menus that the user did not have access to.

## 1.30.11. Upgrading from version 0.14

There is a new artifact called katari-hibernate that contains hibernate dependencies. This stuff was removed from katari-core, so you must add a new dependency in your old application:

```
<dependency>
  <groupId>com.globant.katari</groupId>
  <artifactId>katari-hibernate</artifactId>
  <version>0.15</version>
</dependency>
```

This version changed the way that some filters are used. The acegi, encoding and serviceUrlFilter are now declared in the core anubis bean factory and they no longer belong to the web.xml file. You should remove those filters from your web.xml.

The way the security filter is configured has changed: it is no longer necessary to declare the 'Acegi Filter Chain Proxy' filter in the web.xml file.

Keeping the 'Acegi Filter Chain Proxy' filter will throw an error stating that the bean anubis.securityFilter no longer exists.

Keeping the encoding filter will throw an error stating that com.globant.katari.core.web.EncodingFilter was not found.

Keeping the `serviceUrlFilter` will throw an error stating that `com.globant.katari.core.web.ServiceUrlFilter` was not found.

## 1.30.12. What's new since version 0.14

- One test succeeded or failed depending on the order it was ran.

## 1.30.13. What's new since version 0.13

- Upgraded spring to version 2.5.6.
- Added prototype to the ajax module.
- Improvements to the reporting module.
  - Combos can now be nested.
  - The filled data is not cleared when the form is redrawn in case of error.
  - It is now possible to download the jrxml file.

## 1.30.14. Resolved issues since version 0.13

- Non privilege users where able to modify other users data.

## 1.30.15. Upgrading from version 0.13

If you where using the constructor:

```
public ServletAndParameters(final HttpServlet servlet,  
    final Map<String, String> parameters);
```

now, parameters cannot be null. You have to use

```
public ServletAndParameters(final HttpServlet servlet);
```

instead

## 1.30.16. What's new since version 0.12.1

- Upgraded trails to version 1.2.1
- Now it is possible to define a query criteria for the drop downs generated by trails.
- It is now possible to override the security permission of the trails and report modules.
- Integrated c3p0 pool in the sample application.
- Improved menu access validation logic, now a user must have permission to see at least one leaf in a menu for anubis to show a parent menu.

## 1.30.17. Upgrading from version 0.12.1

The only changes for this version are related to the update to trails 1.2.1. This version of trails now includes `InitialValue` and `PossibleValues`, so classes that uses those annotations must use:

```
import org.trails.descriptor.annotation.InitialValue;
import org.trails.descriptor.annotation.PossibleValues;
```

## 1.30.18. What's new since version 0.12

- Added the possibility to split the application in frontoffice and backoffice features with different security implementations.

## 1.30.19. Upgrading from version 0.11.2

If you want to use the new local login module, the old sample functional tests will fail due to the differences between the local and CAS login pages. This version has a modified `SimplePageVerifier` that works against both login modules.

## 1.30.20. What's new since version 0.11.2

- New module `katari-local-login`. CAS is now optional.
- Made the `fckeditor` configurable from the web application context.
- Show the report title in the report parameters page.

## 1.30.21. Resolved issues since version 0.11.2

- Fixed a typo in the `cas.validatorUrlTransformer` bean name test.

## **1.30.22. What's new since version 0.11.1**

No new features were added to this release.

## **1.30.23. Resolved issues since version 0.11.1**

- When katari-report validated a jrxml report, the validator tried to get the DTD file from its uri.

## **1.30.24. What's new since version 0.11**

- It is possible to configure the host name of the cas validator server.

## **1.30.25. What's new since version 0.10**

- The page shown after the user logs out is now configurable.
- It is now possible to add additional editors and viewers to trails.
- The report now supports time data.

## **1.30.26. Resolved issues since version 0.10**

- Fixed the default view page in trails module.
- Fixed the default list page in trails module.
- Reworked the dependencies.

## **1.30.27. What's new since version 0.9.1**

- Upgraded trails to version 1.1.99 (a preview release of version 1.2).
- Removed the people search module, it did not add any real value.

## **1.30.28. Resolved issues since version 0.9.1**

- Fixed the logout link in the sample.
- The footer obscured the bottom of the page when the content was too big to fit in the screen.

## **1.30.29. What's new since version 0.9**

- katari-ajax now includes FCKEditor scripts.

### **1.30.30. Resolved issues since version 0.9**

- Modifying the name of a report forced the user to upload a new report file.

### **1.30.31. What's new since version 0.8.1**

- The cas module is now in its own package katari-cas.
- The reports can now display a drop down in the parameters page.

### **1.30.32. What's new since version 0.8**

- The html pages in the sample application now uses 100% of page height.

### **1.30.33. Resolved issues since version 0.8**

- The top level menu was hardcoded in the core spring application context and it was difficult to change.
- The report module now validates if the uploaded file is a valid jasper report.

### **1.30.34. What's new since version 0.7**

- Split the spring configuration in 'core' and 'user' beans. This will make it easier to upgrade to newer anubis versions.

### **1.30.35. What's new since version 0.6**

- 'Flatten' the trails component structure for default pages, making it easier to customize the look and feel.
- Menus do not have roles any more, access is directly checked on the url that the menu points to.
- You can now check in an ftl template if a url is accessible before rendering an html fragment. This is like the acegisecurity tag lib, but checks for a url instead of a role.
- The ui is now fully modularized, including the sitemesh decorators and their mappings.
- Some improvement in the user management module.
- New katari-ajax module that packages the yui framework.

- There is a new option that, if you do not have permissions to see a button, it shows it anyway with a special mark. This makes it easier to test if an action is effectively being validated for permissions, instead of only hiding the button. This is only available for a development environment for security reasons.
- There is a freemarker engine intended to be used from unit tests. This engine makes it possible to test the ftl rendering process without starting a servlet container, speeding up the development cycle.
- Draft of a first approach to generate pdf files from the maven site.

## **1.30.36. What's new since version 0.5**

- There is a new syntax to define menus, making the menus more readable and compact.
- Menus are now modularized. There is a classic-menu module that implements simple button like menus.
- New module katari-report, that implements reporting with JasperReports.
- Modules can add filters to the request processing.
- Security is now defined in each module.
- Modules now register their persistent and crud JPA classes. It is no longer necessary for module users to manually add these classes to the application context.
- Added events through Mule support.

## **1.30.37. Resolved issues since version 0.5**

- MenuNodes could be created with null authorities, violating the MenuNode contract.