Settings Subdomain
Table of Contents

Screenshots .......................................................... 2
  App Settings ....................................................... 2
  User Settings ..................................................... 4
How to configure/use .................................................. 6
  Classpath ........................................................... 6
  Bootstrapping ...................................................... 6
API ................................................................. 7
  ApplicationSettingsService and ApplicationSettingsServiceRW .................................................. 7
  UserSettingsService and UserSettingsServiceRW ................................................................. 7
Implementation ........................................................ 9
Known issues ............................................................ 10
  Dependencies ...................................................... 10
This module (*isis-module-settings*) provides the ability to persist application- and user- configuration settings.

With *ApplicationSettingsService* these settings have global scope; for the *UserSettingsService* the settings are scoped per user.

The settings themselves are keyed by a simple string, and can store any of boolean, String, int, long and *LocalDate*. The implementation persists these values in a single raw format, but the API exposed by the services aims to be type-safe.
The module’s functionality can be explored by running the quickstart with example usage using the `org.incode.domainapp.example.app.modules.ExampleDomDomSettingsAppManifest`.

A home page is displayed when the app is run:

```
HTTP/1.1 200 OK
Date: Tue, 10 Jul 2012 16:17:28 GMT
Content-Type: text/html
Content-Length: 969

<!DOCTYPE html>
<html>
  ...
</html>
```

The remaining screenshots below do demonstrate the functionality of this module, but are out of date in that they are taken from the original isisaddons/incodehq module (prior to being amalgamated into the incode-platform).

**App Settings**

List all (demo) application settings:
listed in a table:

<table>
<thead>
<tr>
<th>Key</th>
<th>Description</th>
<th>Type</th>
<th>Value/Row</th>
</tr>
</thead>
<tbody>
<tr>
<td>demoSettingBaseUrl</td>
<td>Demo base URL for RESTful access</td>
<td>STRING</td>
<td><a href="http://secure.mycompany.com">http://secure.mycompany.com</a></td>
</tr>
<tr>
<td>demoSettingIsDebug</td>
<td>Demo: Whether it's debug mode enabled</td>
<td>BOOLEAN</td>
<td>true</td>
</tr>
<tr>
<td>demoSettingLastData</td>
<td>Demo: Latest data available in this application</td>
<td>LOCAL_DATE</td>
<td>2010-01-01</td>
</tr>
<tr>
<td>demoSettingMaxOrderNumber</td>
<td>Demo: Number from which to start</td>
<td>LONG</td>
<td>100000000000</td>
</tr>
<tr>
<td>demoSettingMaxServicesRetries</td>
<td>Demo: #times to retry submitting web service retries before failing</td>
<td>INT</td>
<td>3</td>
</tr>
</tbody>
</table>

and inspect detail:
User Settings

List all (demo) user settings:

Listed in a table:
and inspect detail:

<table>
<thead>
<tr>
<th>User</th>
<th>Key</th>
<th>Description</th>
<th>Type</th>
<th>Value Raw</th>
</tr>
</thead>
<tbody>
<tr>
<td>dick</td>
<td>demoSettingNotifyByEmail</td>
<td>Denies whether to notify changes by email</td>
<td>BOOLEAN</td>
<td>false</td>
</tr>
<tr>
<td>swen</td>
<td>demoSettingNotifyByEmail</td>
<td>Denies whether to notify changes by email</td>
<td>BOOLEAN</td>
<td>true</td>
</tr>
<tr>
<td>swen</td>
<td>demoSettingHasRecentOutMessageSent</td>
<td>Denies most recent that message sent</td>
<td>STRING</td>
<td>Grab some lunch?</td>
</tr>
<tr>
<td>swen</td>
<td>demoSettingHasRecentEmail</td>
<td>Denies whether to notify changes by email</td>
<td>BOOLEAN</td>
<td>true</td>
</tr>
<tr>
<td>swen</td>
<td>demoSettingPlaceOrderDefaultQuantity</td>
<td>Denies Default quantity when placing an order</td>
<td>INT</td>
<td>1</td>
</tr>
<tr>
<td>swen</td>
<td>demoSettingThemeRGBa</td>
<td>Denies theme, as RGBA format</td>
<td>LOONG</td>
<td>12306425404D</td>
</tr>
</tbody>
</table>

**Example:**

- **demoSettingNotifyByEmail**
  - **User**: dick
  - **Key**: demoSettingNotifyByEmail
  - **Description**: Denies whether to notify changes by email
  - **Type**: BOOLEAN
  - **Value**: false
How to configure/use

Classpath

Update your classpath by adding this dependency in your dom project's pom.xml:

```
<dependency>
    <groupId>org.incode.example.settings</groupId>
    <artifactId>incode-example-settings-dom</artifactId>
</dependency>
```

Check for later releases by searching Maven Central Repo.

Bootstrapping

In the AppManifest, update its getModules() method, eg:

```
@override
public List<Class<? extends ?>> getModules() {
    return Arrays.asList(
        ...,
        org.incode.example.settings.SettingsModule.class,
        ...)
};
```
API

ApplicationSettingsService and ApplicationSettingsServiceRW

The module defines two interfaces for application settings. The first, ApplicationSettingsService, provides read-only access:

```java
public interface ApplicationSettingsService {
    ApplicationSetting find(String key);
    List<ApplicationSetting> listAll();
}
```

The second, ApplicationSettingsServiceRW, extends the first and allows settings to be created:

```java
public interface ApplicationSettingsServiceRW extends ApplicationSettingsService {
    ApplicationSetting newBoolean(String name, String description, Boolean defaultValue);
    ApplicationSetting newString(String name, String description, String defaultValue);
    ApplicationSetting newLocalDate(String name, String description, LocalDate defaultValue);
    ApplicationSetting newInt(String name, String description, Integer defaultValue);
    ApplicationSetting newLong(String name, String description, Long defaultValue);
}
```

UserSettingsService and UserSettingsServiceRW

The module defines two interfaces for user settings. These are almost identical to the application settings above, the significant difference being each setting is additional identified by the username that 'owns' it.

The first interface, UserSettingsService, provides read-only access:

```java
public interface UserSettingsService {
    UserSetting find(String user, String key);
    List<UserSetting> listAll();
    List<UserSetting> listAllFor(String user);
}
```

The second, UserSettingsServiceRW, extends the first and allows settings to be created:
public interface UserSettingsServiceRW extends UserSettingsService {
    UserSetting newBoolean(String user, String name, String description, Boolean defaultValue);
    UserSetting newString(String user, String name, String description, String defaultValue);
    UserSetting newLocalDate(String user, String name, String description, LocalDate defaultValue);
    UserSetting newInt(String user, String name, String description, Integer defaultValue);
    UserSetting newLong(String user, String name, String description, Long defaultValue);
}
Implementation

The ApplicationSettingsServiceJdo implements ApplicationSettingsServiceRW (and therefore also ApplicationSettingsService).

Similarly, the UserSettingsServiceJdo implements UserSettingsServiceRW (and therefore also UserSettingsService).
Known issues
None known at this time.

Dependencies
Maven can report modules dependencies using:

```
mvn dependency:list -o -pl modules/dom/settings/impl -D excludeTransitive=true
```

which, excluding the Apache Isis modules, returns no direct compile/runtime dependencies.

The module does use icons from icons8.