

|  |
| --- |
| Spring Template |
| functions Rev: |

Contents

[**Introduction** 0](#_Toc525838385)

[**1.** **Functions** 0](#_Toc525838386)

[1.1. Get Setting Value 0](#_Toc525838387)

[1.2. Send notification 0](#_Toc525838388)

[1.3. Add Logs 1](#_Toc525838389)

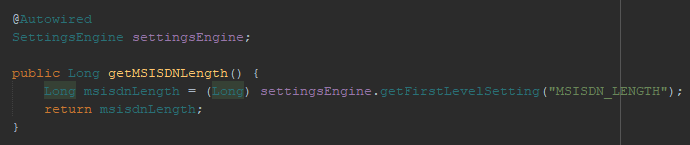
[1.4. Return message based on user language 1](#_Toc525838390)

**Introduction**

This document describes the functions used by the IVR Applications to communicate with the Spring Template application.

1. **Functions**
   1. Get Setting Value

All DB settings are loaded in memory for better performance. To access a setting value we should autowire the “settingsEngine” the call the function “getFirstLevelSetting” and set as input the setting name. This function will return an object and then the object can be casted to (Long, String or Boolean) as below:



Two controllers are added for settings engine one controller to refresh the settings in case any error in engine, second one to get all settings value (check API documentation).

* 1. Send notification

A new way to add notifications. On account creation, admin can give a user access to many notifications then user can enable/disable access to these notifications. For better performance all notifications with user collection are saved in memory using “notificationEngine”. To send a notification wen can simply call “addNotification” function played in “AbstractService” and send the notification key and the hyperlink to be redirected if user press on this notification. These function will check all needed checks such as send email or sms or web notification.



Two controllers are added for notification engine one controller to refresh the notifications in case any error in engine, second one to get all notifications (check API documentation).

* 1. Add Logs

Existent class “Logger” has been refactored to meet our logs. To add a log we have four types of logs (ERROR = 1, BRIEF = 2, NORMAL = 3 and DEBUG = 4). A controller has been created to set the logs level so in case log level is 3 (ERROR, BRIEF and NORMAL) logs are displayed. Another controller has been created to set the log history if every one hour a new file is created or every day. The default log type is 1. To add a new log we can simply call the static methods ( Logger.ERROR, Logger. DEBUG, Logger. BRIEF and Logger. NORMAL). These logs took the same input (output message, input data and subdirectory). Logger are saved in tomcat log folder under a folder same as our project name. These functions will automatically detect the function and class name that accessed the “Logger” class. The below example show the log saved in file:

27-09-2018 17:59:42 ==> Class: com.apliman.api.configuration.spirng.HibernateConfiguration ==> Method: hibernateProperties ==> Input[name: Mahmoud, type: sms] ==> Output[exception occured]

DateTime ==> Class: class name ==> Method: method name ==> Input[function input] ==> Output[output message]

* 1. Return message based on user language

A new feature added to Apliman template v2.0.0 where the response can be returned to the user based on his language. To return a message we should first insert the code followed by the message in “messages\_lang.properties” then we can simply call the function “getMessageBasedOnLanguage” and send the key followed by additional data in case needed. This function will return the needed message language in a string.