

Idle Behaviour: How to Navigate and Understand IdleNotifierPlugin

IdleNotifier is a plugin that pings the user (player) immediately when their character becomes idle, or when certain crucial stats reach a defined threshold. The purpose of IdleNotifier is meant for players to re-interact with the game when they receive a notification, so that their character can continue doing what they are doing. Runescape is a subscription based MMO(Massively Multiplayer Online) game and thus idleness is important for the player because idleness can leave a player wasting precious subscription time. A plugin that maximizes time efficiency is important, so the IdleNotifierPlugin is a crucial plugin.

For more information on what the IdleNotifier does:

<https://github.com/runelite/runelite/wiki/Idle-Notifier>

The key files to the IdleNotifierPlugin are **IdleNotifierPlugin.java** and **IdleNotifierConfig.java**, both of which reside in the directory [runelite-client/src/main/java/net/runelite/client/plugins/idlenotifier/](#). The important function to inspect is IdleNotifierPlugin::onGameTick(). Two state variables in IdleNotifierPlugin called **notifier** and **config** are injected by the Notifier and IdleNotifierConfig class from [runelite-client/src/main/java/net/runelite/client/plugins/idlenotifier/IdleNotifierConfig.java](#) and [runelite-client/src/main/java/net/runelite/client/Notifier.java](#), respectively. These two variables are significant because they are used in onGameTick(), which is a method that checks if the animation state has changed to idle via config.animationIdle() (or any other idle method that returns a boolean for that matter) and subsequently calls notifier.notify(). A popup notification, which depends on the operating system running the client, shows up to the user with the corresponding notification message.

Injection for the notifier and config on lines 80 and 86 respectively:

```
79      @Inject
80      private Notifier notifier;
81
82      @Inject
83      private Client client;
84
85      @Inject
86      private IdleNotifierConfig config;
87
```

onGameTick() calling methods that checks for idleness on the config variable and subsequently calling notify() on notifier on lines 414-416 for example:

```

387     @Subscribe
388     public void onGameTick(GameTick event)
389     {
390         final Player local = client.getLocalPlayer();
391         final Duration waitDuration = Duration.ofMillis(config.getIdleNotificationDelay());
392         lastCombatCountdown = Math.max(lastCombatCountdown - 1, 0);
393
394         if (client.getGameState() != GameState.LOGGED_IN
395             || local == null
396             // If user has clicked in the last second then they're not idle so don't send idle notification
397             || System.currentTimeMillis() - client.getMouseLastPressedMillis() < 1000
398             || client.getKeyboardIdleTicks() < 10)
399         {
400             resetTimers();
401             return;
402         }
403
404         if (config.logoutIdle() && checkIdleLogout())
405         {
406             notifier.notify "[" + local.getName() + "] is about to log out from idling too long!";
407         }
408
409         if (check6hrLogout())
410         {
411             notifier.notify "[" + local.getName() + "] is about to log out from being online for 6 hours!";
412         }
413
414         if (config.animationIdle() && checkAnimationIdle(waitDuration, local))
415         {
416             notifier.notify "[" + local.getName() + "] is now idle!";
417         }

```

Within the Notifier class and within the notify() method, sendNotification is called which checks against the OSType so that the client knows what kind of notification to send to the user based on the current platform.

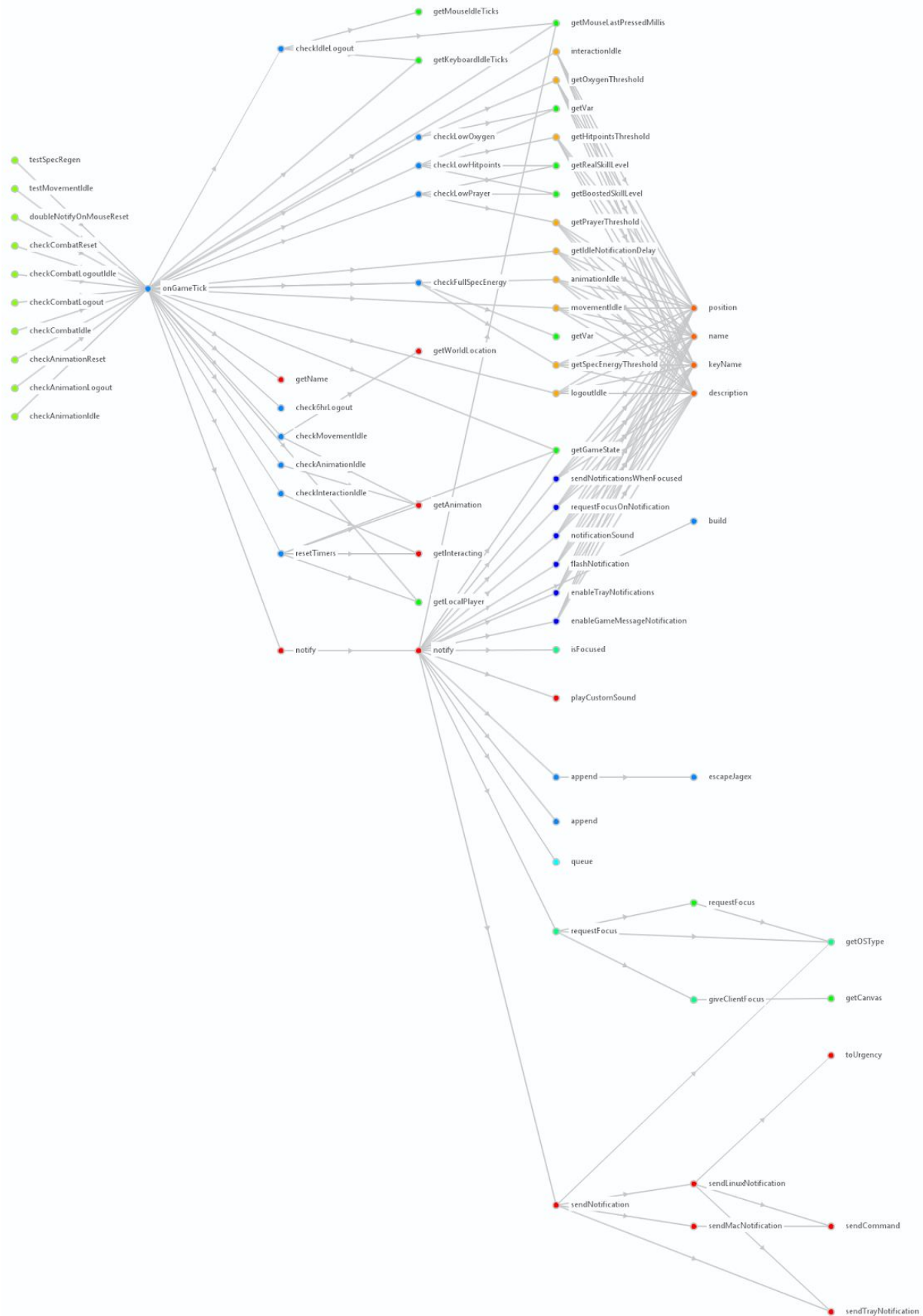
```

private void sendNotification(
    final String title,
    final String message,
    final TrayIcon.MessageType type)
{
    final String escapedTitle = SHELL_ESCAPE.escape(title);
    final String escapedMessage = SHELL_ESCAPE.escape(message);

    switch (OSType.getOSType())
    {
        case Linux:
            sendLinuxNotification(escapedTitle, escapedMessage, type);
            break;
        case MacOS:
            sendMacNotification(escapedTitle, escapedMessage);
            break;
        default:
            sendTrayNotification(title, message, type);
    }
}

```

Call Graph for IdleNotifierPlugin generated from onGameTicks():



Sequence Diagram(truncated) for IdleNotifierPlugin generated from onGameTicks():

