Quartz Enterprise Job Scheduler

```
What is the Quartz Job Scheduling Library?
```

Quartz is a richly featured, open source job scheduling library that can be integrated within virtually any Java application - from the smallest stand-alone application to the largest e-commerce system. Quartz can be used to create simple or complex schedules for executing tens, hundreds, or even tens-of-thousands of jobs; jobs whose tasks are defined as standard Java components that may execute virtually anything you may program them to do. The Quartz Scheduler includes many enterprise-class features, such as support for JTA transactions and clustering.

Features

Runtime Environments

- Quartz can run embedded within another free standing application
- Quartz can be instantiated within an application server (or servlet container), and participate in XA transactions
- Quartz can run as a stand-alone program (within its own Java Virtual Machine), to be used via RMI
- Quartz can be instantiated as a cluster of stand-alone programs (with load-balance and fail-over capabilities) for the execution of jobs

Job Scheduling

Jobs are scheduled to run when a given Trigger occurs. Triggers can be created with nearly any combination of the following directives:

Header 1	Header 2	Header 3	Header 4	Header 5	Header 6
Column 1, row 1	Column 2, row 1	Column 3, row 1	Column 4, row 1	Column 5, row 1	Column 6, row 1
Column 1, row 2	Column 2, row 2	Column 3, row 2	Column 4, row 2	Column 5, row 2	Column 6, row 2

Java Training - Beginner

NOTE	This repository only contains **Java Training - Beginner** classes example code. Other relevant training materials is available at Java Training repository and Training Website.

Class wise code details

WARNING	All classes example code is available in respective class wise git branch. Please switch to respective class branch to see details code. Commit message will give the hint of code progress.

Details curriculum is available at Java Beginner Curriculum

- Class 1 - Java Basic. Code branch name `1-java-basic`

IntelliJ IDEA

IntelliJ IDEA is one of the most popular IDE among java developers. *Ultimate and Community both editions are available for IntelliJ IDEA. In this article developers will find some useful plugins to boost their development experience as well as most used keymaps to keep their hands mostly on the keyboard.*

Useful IntelliJ IDEA Plugins

- To generate boilerplate code (e.g. Getter, Setter and so on) during compile time Lombok plugin is the best choice for the developers. It is very easy to use due to its annotation based code generation.
- AsciiDoc plugin brings Asciidoc supports for the developer. It helps to write the technical documentation easily on the same IDE they have used for the development.
- Extra Icons plugin helps to visually recognize different file formats almost instantly by their icons.
- If you are struggling to find opening and closing brackets in your code, Rainbow Brackets plugin may help you. It helps developers to find brackets through rainbow coloring.
- CodeGlance plugin is a code mini map. It helps to navigate fast across the editor window.

Keymaps

TIP

Keymap learning and presentation plugins for IntelliJ IDEA

- Key Promoter X plugin shows a popup notification with the corresponding keyboard shortcut whenever a command executed using the mouse. It also shows most used keymap statistics.
- Presentation Assistant plugin shows a popup with the corresponding description whenever a keymap pressed. It can be very useful during presentation, screencast.

Both plugin can be used together for learning shortcuts.

NOTE

Download the default keymap reference card by **Jetbrains**. This keymap card is also available inside your IDE under **Help › Keymap Reference**.

beginner:page$introduction-to-ide/intellij-idea-keymap.adoc

Additional Resources

Watch

- Pluralsight - Getting Started with IntelliJ CE by David Starr.
  - Duration **2h 47m**.
  - Released **1 Sep 2017**.

Read

- IntelliJ IDEA Getting started
- IntelliJ IDEA keyboard shortcuts