

IntelliJIDEA



SIMPLY EASY LEARNING

www.tutorialspoint.com





About the Tutorial

IntelliJ IDEA (hereafter referred as IntelliJ) is one of the powerful and most popular Integrated Development Environment (IDE) for Java. It is developed and maintained by JetBrains and available as community and ultimate edition. This feature rich IDE enables rapid development and helps in improving code quality.

This tutorial starts with basic introduction and slowly deep dives into its advance features. Logically tutorial is divided into 2 parts. Absolute beginners can start using it by following first 4 chapters. However to use it more effectively one can refer remaining chapters.

Audience

This tutorial is targeted for absolute beginners as well as moderate users. Absolute beginners will get fair understanding about IntelliJ and moderate users will take their existing knowledge to next level.

Prerequisites

This tutorial assumes preliminary knowledge of software development process and Java programming language. Additionally reader should have basic understanding of computer system. Reader should be able to install, uninstall and configure software packages on his/her system.

Later sections of tutorial discuss integration with build tools, unit testing frameworks, debugger, profiling, version control system and database. It is assumed that those tools are installed and configured on system and reader is familiar with those tools.

Copyright & Disclaimer

© Copyright 2017 by Tutorials Point (I) Pvt. Ltd.

All the content and graphics published in this e-book are the property of Tutorials Point (I) Pvt. Ltd. The user of this e-book is prohibited to reuse, retain, copy, distribute or republish any contents or a part of contents of this e-book in any manner without written consent of the publisher.

We strive to update the contents of our website and tutorials as timely and as precisely as possible, however, the contents may contain inaccuracies or errors. Tutorials Point (I) Pvt. Ltd. provides no guarantee regarding the accuracy, timeliness or completeness of our website or its contents including this tutorial. If you discover any errors on our website or in this tutorial, please notify us at contact@tutorialspoint.com



Table of Contents

About the Tutoriali
Audiencei
Prerequisitesi
Copyright & Disclaimeri
Table of Contentsii
1. INTELLIJ IDEA – INTRODUCTION1
What is IDE and its benefits?1
Features of IntelliJ1
Comparison of ultimate and community edition3
2. INTELLIJ IDEA – INSTALLATION AND CONFIGURATION5
Installation on Windows5
Installation on Linux5
Configure IntelliJ6
3. INTELLIJ IDEA – GETTING FAMILIAR8
Visual elements8
Working with plug-ins9
Working with settings10
Configuring JVM options and platform properties11
4. INTELLIJ IDEA – CREATE FIRST JAVA PROJECT12
Create project
Create package



Crea	ate Java class	13
Run .	Java application	14
5.	INTELLIJ IDEA – DEEP DIVE	15
Unde	lerstanding projects	15
Unde	lerstanding modules	15
Unde	lerstanding folders	16
Unde	lerstanding libraries	17
Unde	lerstanding facets	17
Artif	facts	17
Impo	orting existing project	18
Proje	ect formats	19
Com	nparison of directory and file based project format	21
6.	INTELLIJ IDEA – DEEP DIVE INTO EDITOR	22
Visua	al elements of editor	22
Edito	or actions	25
Linti	ing code	32
7.	INTELLIJ IDEA – CODE REFACTORING	35
Rena	ame	35
Repl	lace code duplicates	36
Сору	у	37
Mov	/e	38
Safe	delete	38
Chan	nge signature	39
Туре	e migration	40



8.	INTELLIJ IDEA – RUNNING PROJECTS	.41
Creat	e temporary configuration	. 41
Creat	e permanent configuration	. 41
Creat	e new configuration	. 42
Shari	ng the configuration	. 43
Run c	configuration	43
9.	INTELLIJ IDEA – BUILD TOOLS	.44
Creat	ing maven project	. 44
Creat	e Gradle Project	. 46
10.	INTELLIJ IDEA – UNIT TESTING	.47
Creat	e unit test	. 47
Run ເ	unit test	. 48
11.	INTELLIJ IDEA - DEBUGGING	.50
Break	kpoints	. 50
Step i	into	. 51
Step	out	. 51
Step	over	. 52
Resui	me program	. 53
Stop	action	. 53
Smar	t step into	. 54
Inspe	ecting variables	. 55
Evalu	ate expression	. 55
12.	INTELLIJ IDEA - PROFILING	.56
Intro	duction	. 56



Benefits 56
Configuration56
Monitoring application57
Insights about threads57
Sampling application58
Profiling application59
13. INTELLIJ IDEA – VERSION CONTROL61
Git61
Subverion 66
14. INTELLIJ IDEA - DATABASES70
Create Database70
Connect to database70
Create table71
Insert data73
Retrieve Data73
15. INTELLIJ IDEA – MIGRATING FROM NETBEANS74
Import NetBeans project74
IntelliJ vs. NetBeans terminology74
Popular shortcuts74
Debugger shortcuts75
FAQs and Tips75
16. INTELLIJ IDEA – MIGRATING FROM ECLIPSE77
Import existing project77
IntelliJ vs. Eclipse terminology77



Popular shortcuts	77
Debugger shortcuts	78
FAO and tips	78



1. INTELLIJ IDEA – INTRODUCTION

IntelliJ is one of the powerful and most popular Integrated Development Environment (IDE) for Java. It is developed and maintained by JetBrains and available as community and ultimate edition. This feature rich IDE enables rapid development and helps in improving code quality.

What is IDE and its benefits?

IDE stands for Integrated Development Environment. It is a combination of multiple tools which makes software development process easier, robust and less error-prone. It has following benefits over plain text editor:

- Integration with useful tools like compiler, debugger, version control system, build tools, various frameworks, application profilers and so on.
- Supports code navigation, code completion, code refactoring and code generation features which boosts development process.
- Supports unit testing, integration testing and code coverage via plug-ins.
- Provides rich set of plug-ins to enhance IDE functionality further.

Features of IntelliJ

Deep insight into code

• Smart code completion

It supports context based code completion. It gives a list of the most relevant symbols applicable in the current context.

• Chain code completion

It is advanced code completion feature which lists applicable symbols accessible via methods or getters in the current context.

• Static members completion

It allows you to easily use static methods or constants and automatically adds required import statements to avoid compilation error.



Detecting duplicates

It finds duplicate code fragments on the fly and gives notification/suggestion about it to user.

Inspections and quick-fixes

Whenever IntelliJ detects that you're about to make a mistake, a little light bulb notification pops up on same line. Clicking it shows the suggestion list.

Developer ergonomics

Editor-centric environment

Quick pop-ups help in checking additional information without leaving the current context.

Shortcuts for everything

IntelliJ IDEA has keyboard shortcuts for nearly everything, including rapid selection and switching between tool windows and many more.

Inline debugger

Inline debugger allows you to debug application in IDE itself. It makes development and debugging process seamless.

Built-in developer tools

Version control

IntelliJ support most of the popular version control system like Git, Subversion, Mercurial, CVS, Perforce, and TFS.

Build tools

IntelliJ supports Java and other build tools like Maven, Gradle, Ant, Gant, SBT, NPM, Webpack, Grunt, and Gulp.

Test runner and code coverage

IntelliJ IDEA lets you perform unit testing with ease. The IDE includes test runners and coverage tools for major test frameworks, including JUnit, TestNG, Spock; Cucumber, ScalaTest, spec2, and Karma.



Decompiler

IntelliJ comes with a built-in decompiler for Java classes. When you want to take a look inside a library that you don't have the source code for, you can do it without using any third-party plug-ins.

Terminal

IntelliJ provides built-in terminal. Depending on your platform, you can work with command line prompt, like PowerShell or Bash.

Database tools

IntelliJ provides database tools which allows you to connect to live databases; run queries; browse and update data; and even manage your schemas in a visual interface from IDE itself.

Application server

IntelliJ supports major application servers: Tomcat, JBoss, WebSphere, WebLogic, Glassfish, and many others.

You can deploy your artifacts onto application servers and debug the deployed applications in IDE itself.

Docker support

Via a separate plug-in, IntelliJ provides a dedicated tool window that lets you connect to locally running Docker machines.

Comparison of ultimate and community edition

Feature	Community Edition	Ultimate Edition
License	Commercial	Open-source, Apache 2.0. It can also be used for commercial development.
Indented developers	Web and enterprise developers	JVM and Android developers
Java, Kotlin, Groovy, Scala	Supported	Supported
Android development	Supported	Supported



	<u> </u>	
Maven, Gradle, SBT	Supported	Supported
Git, SVN, Mercurial, CVS	Supported	Supported
Detecting Duplicates	Supported	Not supported
Perforce, TFS	Supported	Not supported
JavaScript, TypeScript	Supported	Not supported
Java EE, Spring, GWT, Vaadin, Play, Grails, Other Frameworks	Supported	Not supported
Database Tools, SQL	Supported	Not supported



2. INTELLIJ IDEA – INSTALLATION AND CONFIGURATION

Depending upon your requirements you can download community or ultimate edition. As name suggests community edition is absolutely free and we can use it for commercial development as well. However, ultimate edition is paid version and we can evaluate it freely for 30 day.

Installation on Windows

System requirements

- It is compatible with windows 10/8/7/Vista/2003/XP
- Minimum RAM requirement is 1 GB but 2 GB is recommended for seamless usage
- It takes 300 MB of disk space for installation and addition 1 GB for caches
- For better visualization 1024x768 screen resolution is ecommended

Downloading and installation

Downloading

You can download windows installer from their official website. Click here to go to downloads section, choose appropriate edition and click on download button.

Installation

Let us begin with installation followed by configuration steps. Installation of IntelliJ is similar to other software packages. Just double click on installer and follow the on screen instructions to complete installation process.

Installation on Linux

System requirements

- GNOME, KDE or XFCE desktop environment
- Minimum RAM requirement is 1 GB but 2 GB is recommended for seamless usage



- It requires 300 MB of disk space for installation and addition 1 GB for caches
- For better visualization 1024x768 screen resolution is ecommended.

Downloading and installation

Downloading

You can download IntelliJ for Linux from their official website. Click here to go to downloads section, choose appropriate edition and click on download button.

Installation

I have downloaded tar.gz bundle. Note that in my case bundle's name was **ideaIC-2017.2.5.tar.gz** it may change with edition/version. Please use appropriate bundle name.

First extract it using following command:

\$ tar xvf ideaIC-2017.2.5.tar.gz

It will create new directory with **idea-IC-172.4343.14** name. Now change directory to **idea-IC-172.4343.14/bin/** and execute **idea.sh** shell script as shown below:

\$ cd idea-IC-172.4343.14/bin/

\$./idea.sh

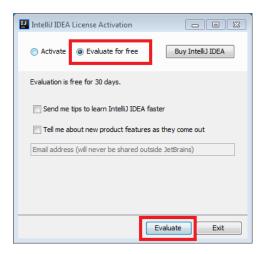
Follow on-screen instructions to complete installation procedure.

Configure IntelliJ

Configuration steps are similar on both platforms. To begin configuration, launch IntelliJ application. Optionally you can import existing configuration from this wizard. Click on next button to continue.

If you are using ultimate edition then it will pop-up license activation window. Select evaluate for free option and click on evaluate button as shown in below image.

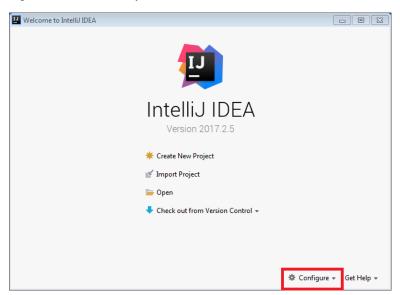




Accept the license agreement to proceed and follow on screen instruction to start IntelliJ. You'll see welcome screen of IntelliJ.

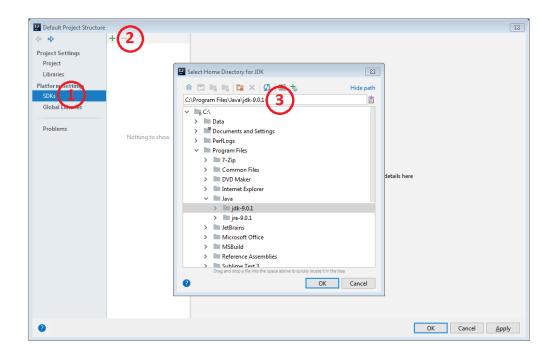
Now it's time to configure Java Development Kit (hereafter we'll refer it as JDK) with IntelliJ. If JDK is not installed already then follow instruction from here.

- 1. On welcome screen click on configure
- 2. Select project defaults from drop-down list
- 3. Select project structure option.



- 4. Select SDKs option from platform settings menu
- 5. Click on plus icon and select JDK option.
- 6. Select JDK's home directory and follow on-screen instruction to complete this process.





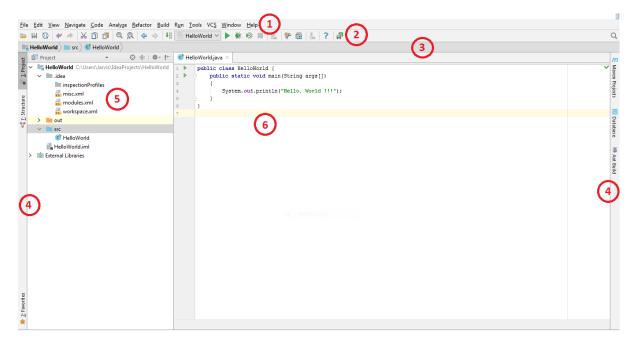


3. INTELLIJ IDEA – GETTING FAMILIAR

To leverage functionality of any tool one must be familiar with that tool and IntelliJ is not exception for that. This chapter gives overview about IntelliJ. Chapter begins with discussion about IDE's visual elements, configuration settings and finally ends by discussing JVM and platform properties.

Visual elements

One of the important things about IDE is its visual elements. Identifying and understanding visual elements enables to you do action in quicker and easier manner. Below screen shows the main interface of IntelliJ.



1. Menu bar

The menu bar provides options to create new projects and other important actions related to projects like code refactoring, builds, run, debug, version controlling options and so on.

2. Tool bar

Tool bar provides to shortcut to compile, debug and run option. You can customize it according to your requirements.



3. Navigation bar

Navigation bar enables easier navigation within project. This feature comes really handy as code base increases.

4. Tools tab

Tools tab is shows on either side of the main window. From here you can access important tools like databases, Maven/Ant builds and so on.

5. Project perspective

Project perspective window shows various elements of projects like packages, modules, classes, external libraries and so on.

6. Editor window

This is a place where developer spends most of his/her time. Editor window allows you edit/write code with syntax highlighting and other useful features.

At the bottom of the main window there is a status bar. It shows some additional attributes about file, like its format and encoding type. It also provides option to toggle read-only attribute of current file. You can also manage inspection level from here.

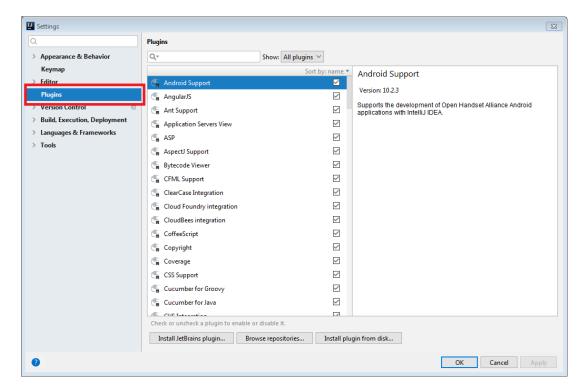
Working with plug-ins

Plug-ins helps to extend functionality of IntelliJ. It provides large number of plug-ins ranging from databases, version controlling, profiling and this list goes on.

To manage plug-ins:

- 1. Go to **File->Settings** menu
- 2. Select plug-ins tab from left side.





- 3. This window lists all installed plug-ins. There is a check-box on right side of each plug-in name. Toggling that check-box enables/disables plug-ins.
- IntelliJ online plug-in repository is available here. To add/remove plug-in repository just click on browse repository button and it will provide a way to do needful.
- 5. In addition to this it allows offline plug-in installation. To do that download plug-in and select install plug-in from disk button and navigate to the download path.

To perform other action on plug-ins like uninstalling, updating and sorting just right click on any plug-in. It'll show drop down menu from which you can select one of the action.

Working with settings

This section provides some tips and trick to manage settings. It enables you to import, export and share IntelliJ settings.

1. Export settings

It allows exporting current settings as a jar file.

- 1. Go to File->Export Settings.
- 2. Export setting windows list the all available settings related to UI, debugger, SDK along with others.



3. It provides check-box for selection. Once selection is done click on ok button to save setting on local disk.

2. Import settings

It allows importing setting stored in jar file.

- 1. Go to File->Import settings.
- 2. Select setting jar by navigating folder structure
- 3. Click on OK button.

3. Sharing settings

Intellij IDEA allows you to share your IDE settings between different instances. This is particularly useful when you want to apply same settings within team or organization.

Prerequisite for this is **Settings Repository** plug-in must be enabled. It installed and enabled by default. To ensure its status:

- 1. Go to File->Settings->Plugins
- 2. Search settings repository plug-in

We can store current setting on GitHub/Bitbucket and apply them on other instances. To store current setting:

- 1. Go to File->Settings Repository
- 2. type Git repository URL in upstream URL dialog box
- 3. Click on overwrite remote button.
- 4. To apply same settings on other instances just click on overwrite local button.

Configuring JVM options and platform properties

We can configure JVM options by navigating to the **Help->Edit Custom VM Options** menu. Following are the some important JVM options we can set.

- -server : Allows selection of the Java HotSpot Server VM
- o -Xms<size>: It sets initial Java heap size. Default value is 128 MB
- o -Xmx<size>: It sets maximum Java heap size. Default value is 750 MB
- -Xss<size> : It sets Java thread stack size
- -XX : It allows setting GC algorithm and other properties.

Please refer Java's official document to tweak JVM options.



End of ebook preview If you liked what you saw... Buy it from our store @ https://store.tutorialspoint.com

