Eclipse Shortcuts

Work with ease

Nagarro Software Pvt. Ltd.
Java Software & Architecture Group

Revision History				
Version	Date	Author/Contributor	Reviewer	Comments
0.1	20-Dec-2012	Vaibhav Shukla	Kuldeep Singh	Released First version of Eclipse useful shortcuts
1.0	28-Jan-13	Vaibhav Shukla		Updated with some more useful shortcuts (no. 28 to 36)
1.1	18-03-2013	Vaibhav Shukla		Updated with 'show print margin' editor preference
1.2	17-07-2013	Vaibhav Shukla		Updated with 'automatic code cleanup and formatting on save'

Contents

Introduction

This document is built to make you aware of the useful eclipse shortcuts. As a developer, this will help you to explore eclipse and use it in more interactive way, because, as they say "the less you touch the mouse, the more you can write the code".

Shortcuts

Top 10 Shortcuts

1) **CTRL + D**

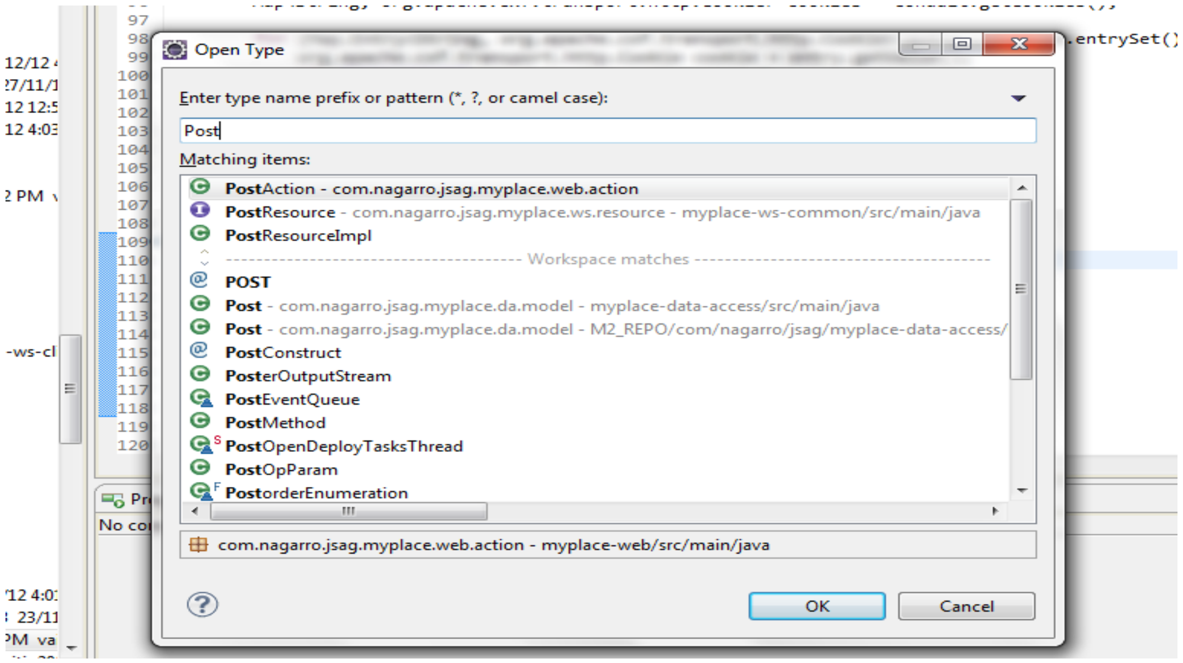
   Delete the row! No more selecting the whole line and pressing Delete key, this shortcut will remove the line you want. Try it.

2) **CTRL + SHIFT + O**

   Organize import statements. Very common to use. Imports all missing classes used in your class and removes all unused imports.

3) **CTRL + SHIFT + T**

   Open Type. Will help you locate any class of your application, you need to look for. No Need to explore the packages and folders, just use this combination and type the class name.



Enter type name prefix or pattern (*, ?, or camel case):

Post

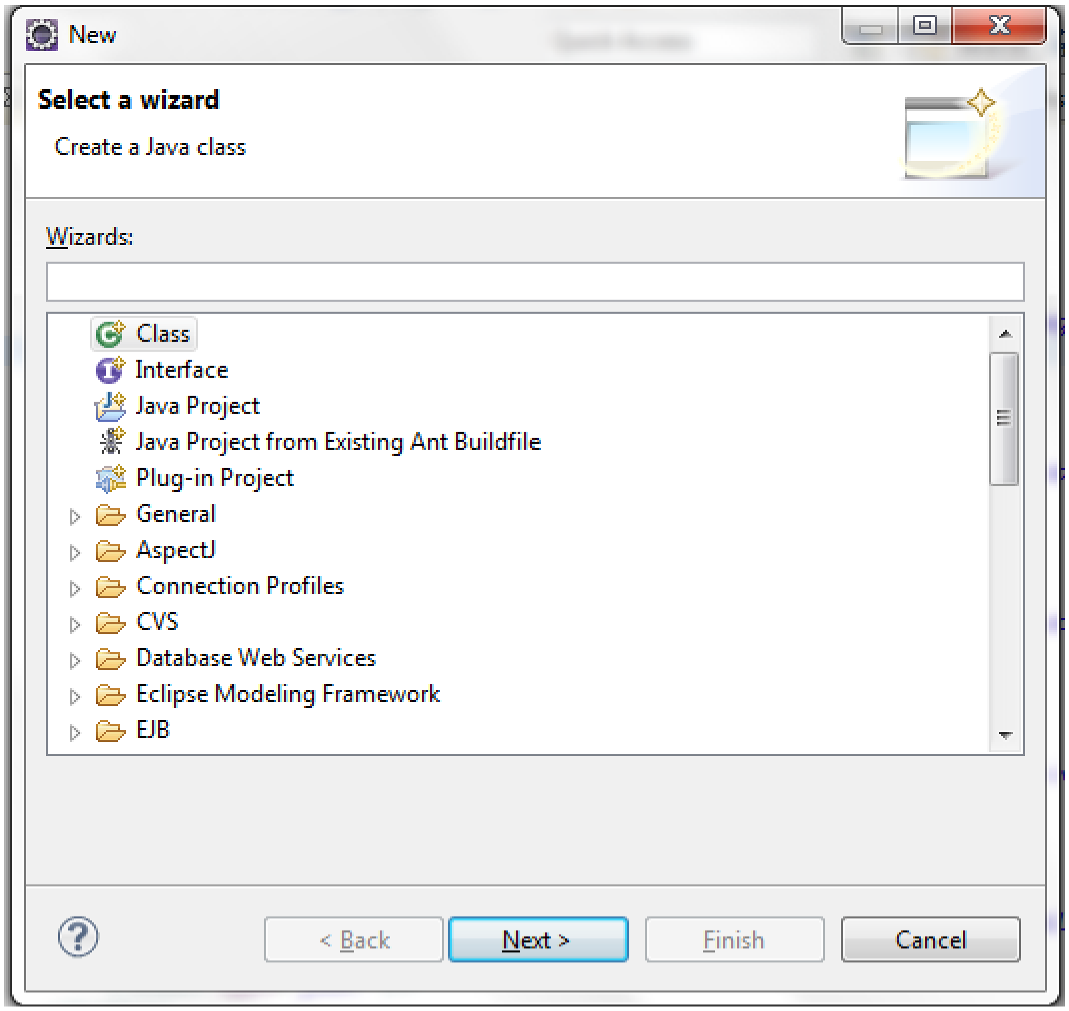
Matching items:

```
PostAction - com.nagarro.jsag.myplace.web.action
PostResource - com.nagarro.jsag.myplace.ws.resource - myplace-ws-common/src/main/java
PostResourceImpl
------------------------------------ Workspace matches ------------------------------------
POST
Post - com.nagarro.jsag.myplace.da.model - myplace-data-access/src/main/java
Post - com.nagarro.jsag.myplace.da.model - M2_REPO/com/nagarro/jsag/myplace-data-access/
PostConstruct
PosterOutputStream
PostEventQueue
PostMethod
PostOpenDeployTasksThread
PostOpParam
PostorderEnumeration
```

com.nagarro.jsag.myplace.web.action - myplace-web/src/main/java

4) CTRL + N

Opens the 'New Type' wizard. No need to right click the project and select New. This combination will open the wizard. Just enter the type and go on.



5) CTRL + F11

Runs the application. The launch resource will depend on your settings. By Default, it will launch the currently selected resource.

6) CTRL + SHIFT + F

Most widely used eclipse shortcut. This formats and beautifies your code. You can add your own formatting style under

Window-> Preferences-> Java-> Code style-> Formatter.

7) CTRL + SPACE

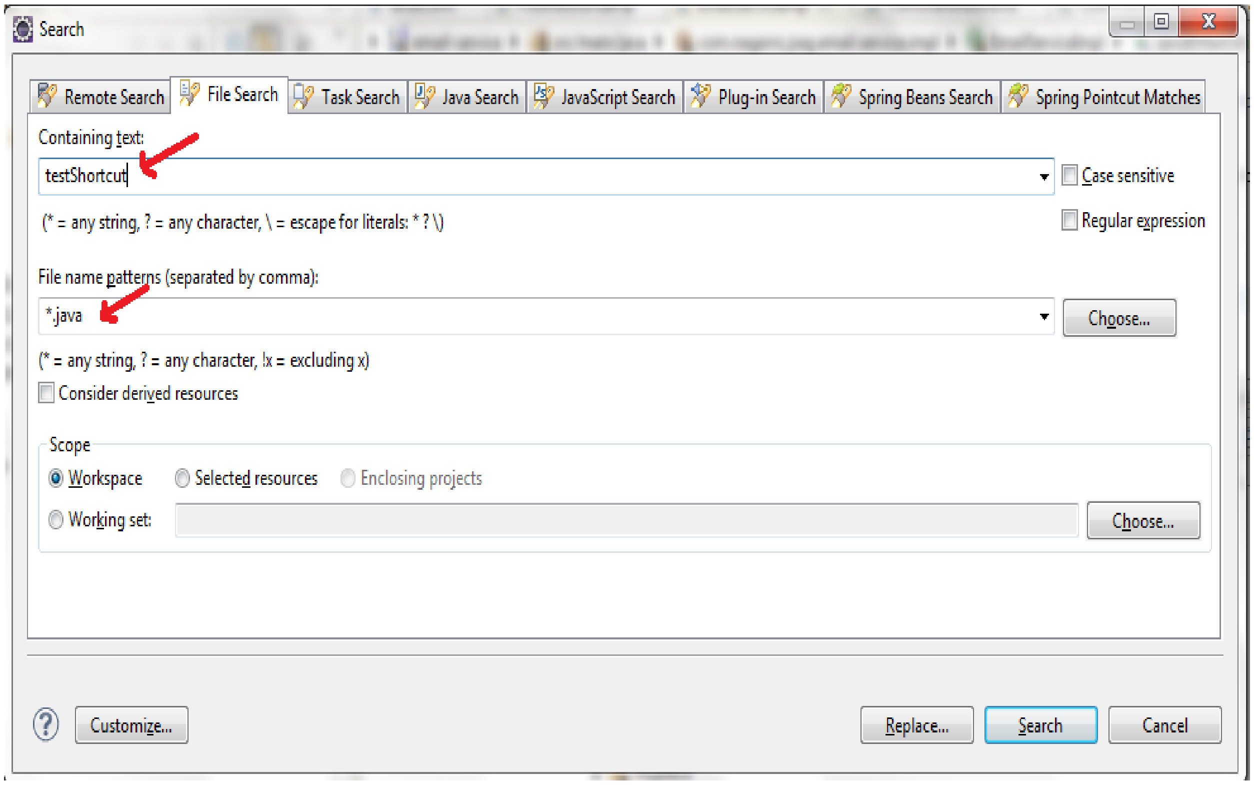
Most favorite of all. Type in something and press ctrl + space and chances are the eclipse will have some hint for you to generate the code.

For example,

- you type "main" and hit "ctrl+space", eclipse will show you the dialog to generate *main method*
- Write "for" and hit "ctrl+space" and it will show you the dialog to generate for loops etc.
- Write class name and use this combination, it will ask you to generate default constructor for the class.

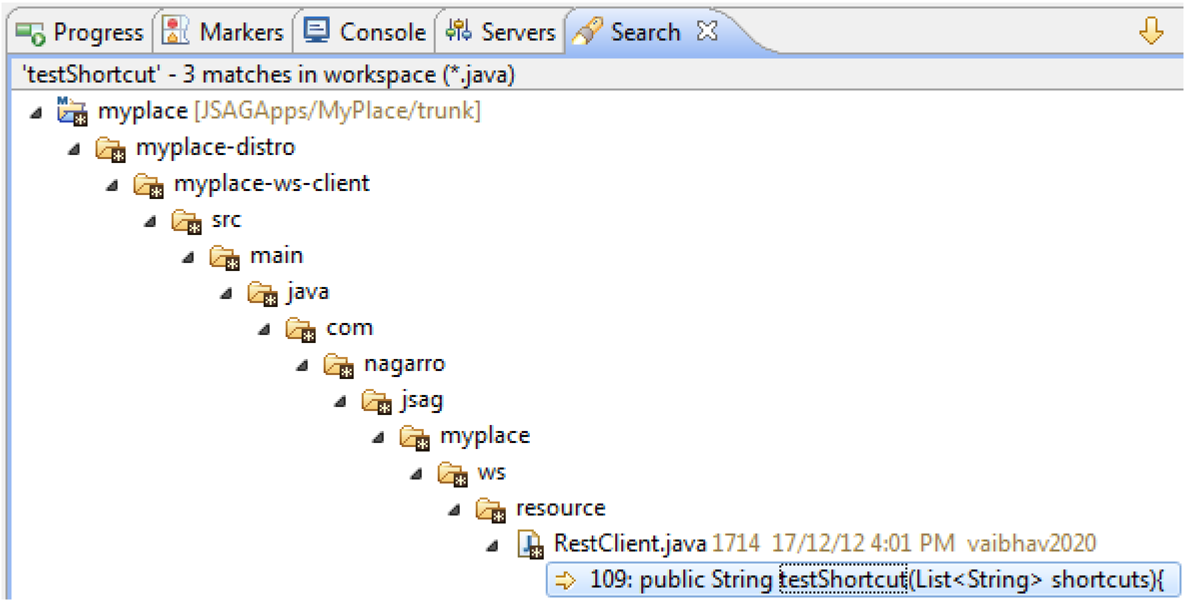
8) CTRL + H

Opens the search wizard. You can search for the text you need and also can specify the type of file where you need to search the text.
Suppose you are looking for the method name "testShortcut" which is in some java class, you can use this combination, specify the file type and can search for it.



Containing text:
testShortcut
☐ Case sensitive
(* = any string, ? = any character, \ = escape for literals: * ? \)
☐ Regular expression
File name patterns (separated by comma):
*.java
(* = any string, ? = any character, !x = excluding x)
☐ Consider derived resources
Scope
● Workspace ○ Selected resources ○ Enclosing projects
○ Working set:

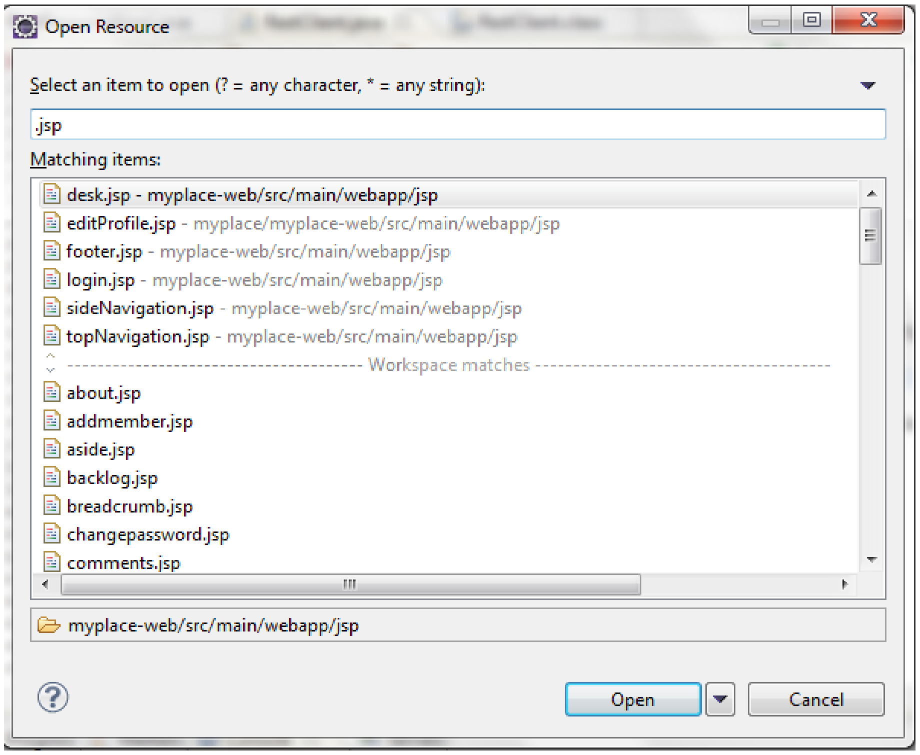
And your search result will be



'testShortcut' - 3 matches in workspace (*.java)
myplace [JSAGApps/MyPlace/trunk]
myplace-distro
myplace-ws-client
src
main
java
com
nagarro
jsag
myplace
ws
resource
RestClient.java 1714 17/12/12 4:01 PM vaibhav2020
109: public String testShortcut(List<String> shortcuts){

9) CTRL + SHIFT + R

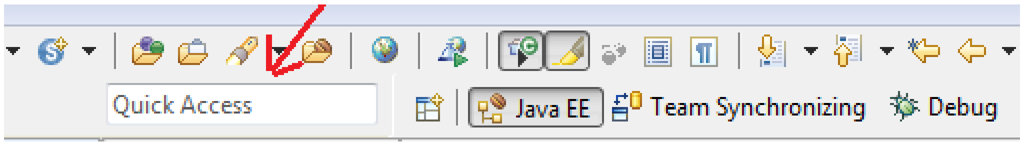
Similar to CTRL+SHIFT+T. Open resource wizard to help you look for any resource.



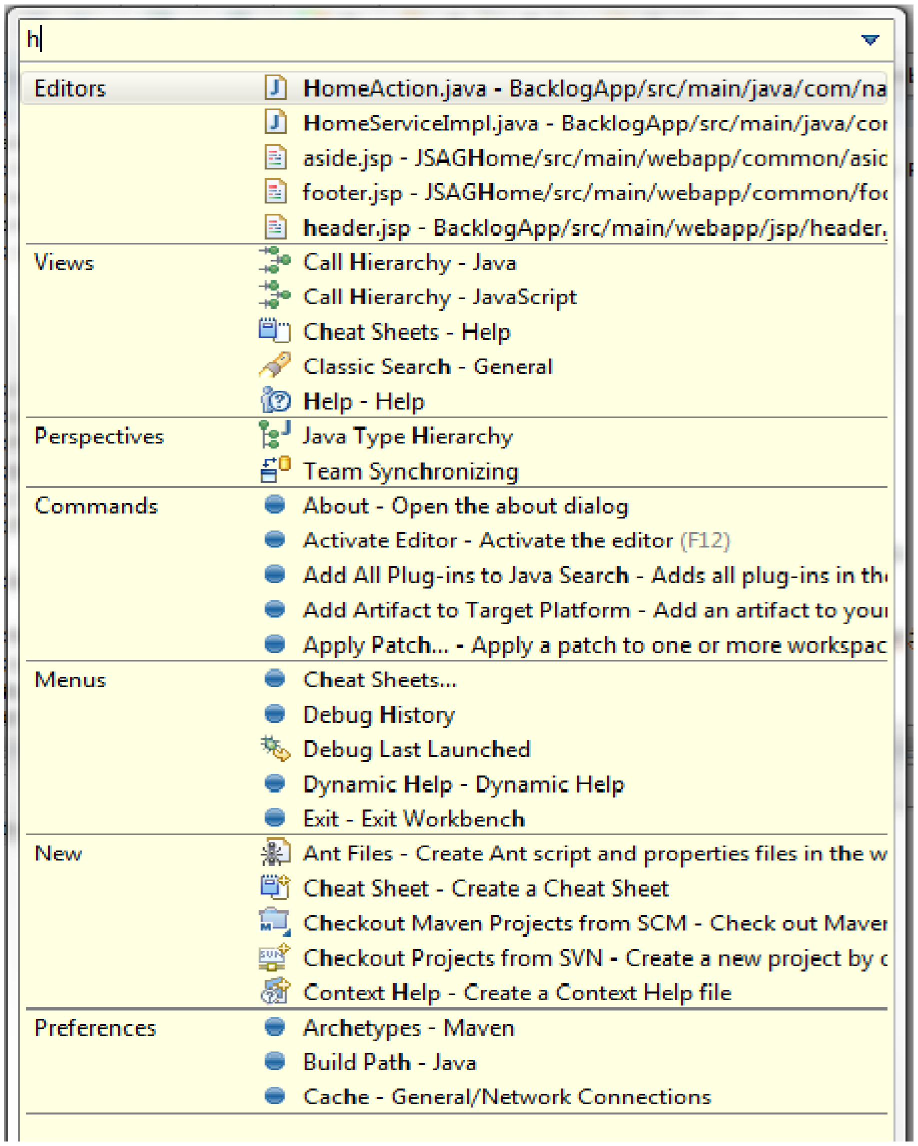
Select an item to open (? = any character, * = any string):
.jsp
Matching items:
desk.jsp - myplace-web/src/main/webapp/jsp
editProfile.jsp - myplace/myplace-web/src/main/webapp/jsp
footer.jsp - myplace-web/src/main/webapp/jsp
login.jsp - myplace-web/src/main/webapp/jsp
sideNavigation.jsp - myplace-web/src/main/webapp/jsp
topNavigation.jsp - myplace-web/src/main/webapp/jsp
------------------------------------- Workspace matches -------------------------------------
about.jsp
addmember.jsp
aside.jsp
backlog.jsp
breadcrumb.jsp
changepassword.jsp
comments.jsp
myplace-web/src/main/webapp/jsp

10) CTRL + 3

A quick access to almost everything in eclipse. The latest eclipse “JUNO” has a search bar named “quick access”



While in other versions of eclipse, this combination proves to be the “Entry Point” to almost everything in eclipse.



h

Editors
- HomeAction.java - BacklogApp/src/main/java/com/na
- HomeServiceImpl.java - BacklogApp/src/main/java/cor
- aside.jsp - JSAGHome/src/main/webapp/common/asic
- footer.jsp - JSAGHome/src/main/webapp/common/foc
- header.jsp - BacklogApp/src/main/webapp/jsp/header.

Views
- Call Hierarchy - Java
- Call Hierarchy - JavaScript
- Cheat Sheets - Help
- Classic Search - General
- Help - Help

Perspectives
- Java Type Hierarchy
- Team Synchronizing

Commands
- About - Open the about dialog
- Activate Editor - Activate the editor (F12)
- Add All Plug-ins to Java Search - Adds all plug-ins in th
- Add Artifact to Target Platform - Add an artifact to you
- Apply Patch... - Apply a patch to one or more workspac

Menus
- Cheat Sheets...
- Debug History
- Debug Last Launched
- Dynamic Help - Dynamic Help
- Exit - Exit Workbench

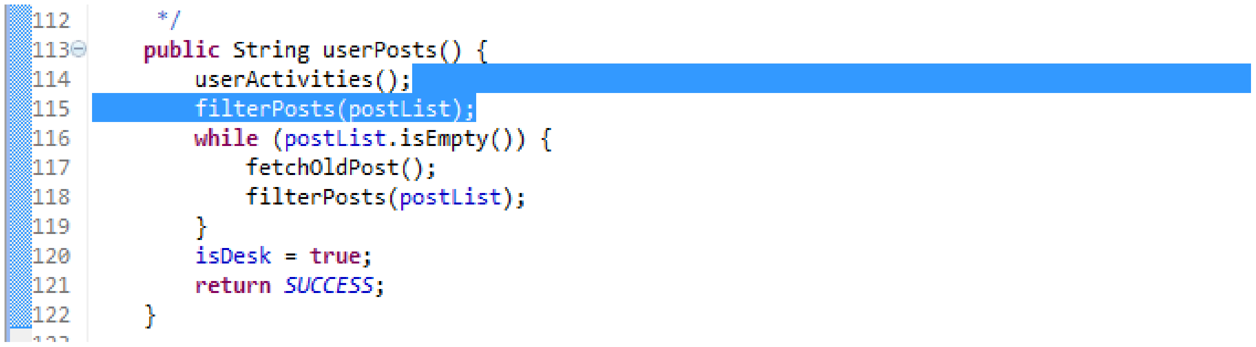
New
- Ant Files - Create Ant script and properties files in the w
- Cheat Sheet - Create a Cheat Sheet
- Checkout Maven Projects from SCM - Check out Maver
- Checkout Projects from SVN - Create a new project by c
- Context Help - Create a Context Help file

Preferences
- Archetypes - Maven
- Build Path - Java
- Cache - General/Network Connections

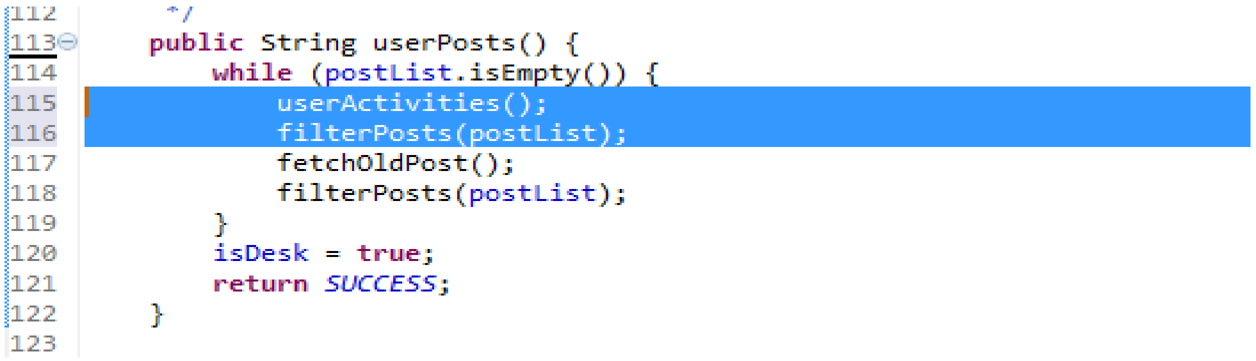
Other Shortcuts

11) ALT + Up/Down Arrow

Move the line or the selected code up or down. Very helpful while re-arranging the and aligning the code.



```
    */
   public String userPosts() {
       userActivities();
       filterPosts(postList);
       while (postList.isEmpty()) {
           fetchOldPost();
           filterPosts(postList);
       }
       isDesk = true;
       return SUCCESS;
   }
```



```
    */
   public String userPosts() {
       while (postList.isEmpty()) {
           userActivities();
           filterPosts(postList);
           fetchOldPost();
           filterPosts(postList);
       }
       isDesk = true;
       return SUCCESS;
   }
```

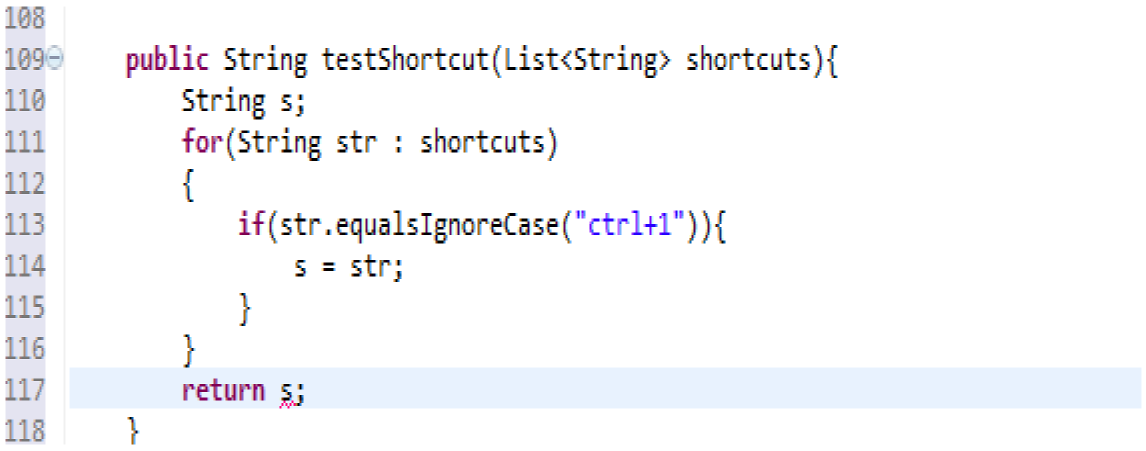
12) ALT + Left/Right Arrow

One of the most useful shortcuts for eclipse. Alt+left arrow will take you to the class you last edited while Alt+right arrow will bring you back to the current class you were working.

Suppose, you are working on class “Foo” and want to look for the changes you did in class “Bar” just before coming to Foo. For this, no need to look into your project explorer to find the class Bar and then scrolling to the line where you made the changes. Just press **alt+left arrow,** and it will take you to the exact line where you need to see and then **alt+right arrow** will bring you back.

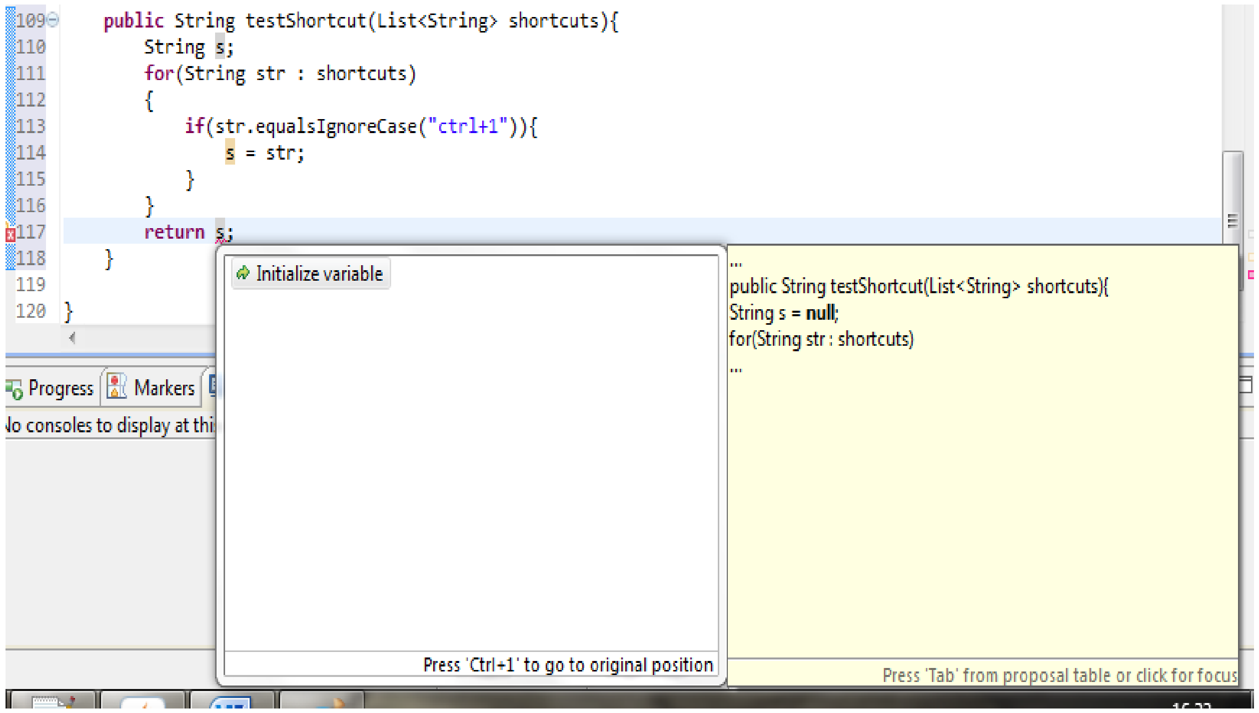
13) CTRL + 1

The most useful shortcut. This activates the quick fix. Reach to the line where you are getting some error and press this combination, this will fix the issue.
Suppose you create a local variable and have not initialized it before using. Then reach to the line where you are prompted for error, try this combination and it will activate the quick fix



```
    public String testShortcut(List<String> shortcuts){
        String s;
        for(String str : shortcuts)
        {
            if(str.equalsIgnoreCase("ctrl+1")){
                s = str;
            }
        }
        return s;
    }
```

Now when you press **ctrl + 1**, you get



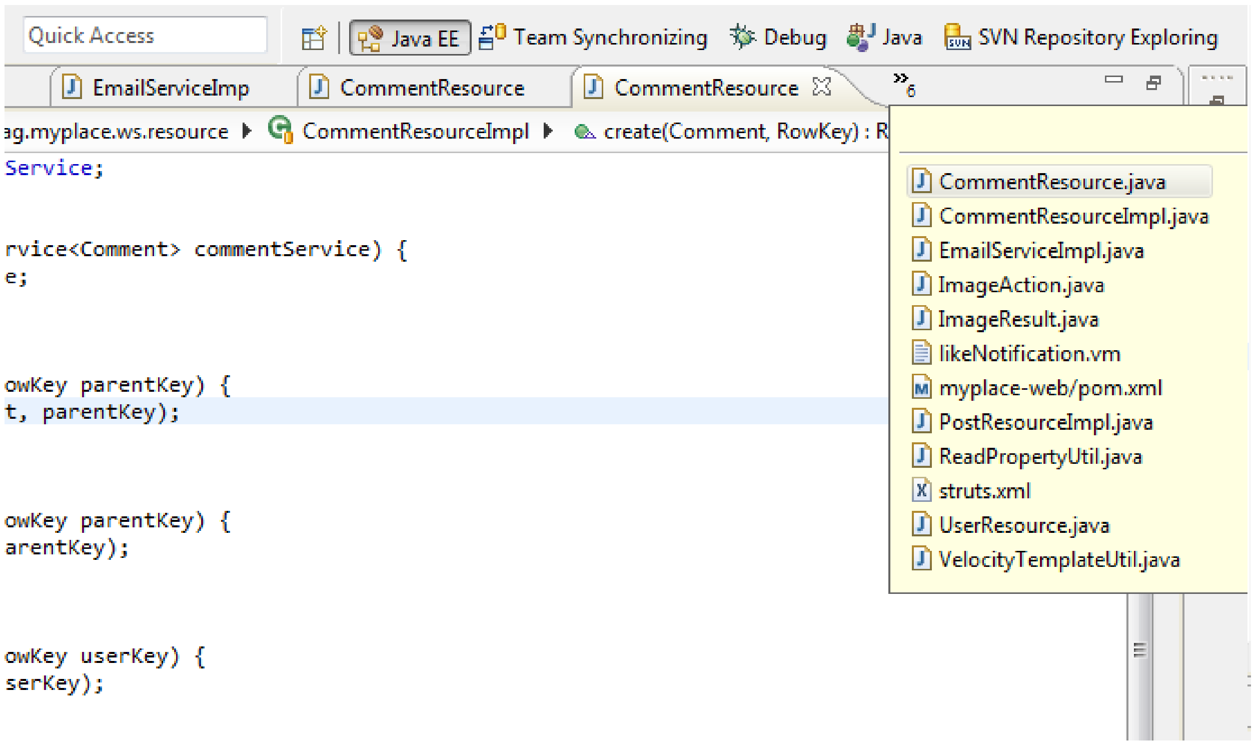
```
    public String testShortcut(List<String> shortcuts){
        String s;
        for(String str : shortcuts)
        {
            if(str.equalsIgnoreCase("ctrl+1")){
                s = str;
            }
        }
        return s;
    }

}
```

Initialize variable
Progress
Markers
No consoles to display at thi
Press 'Ctrl+1' to go to original position
Press 'Tab' from proposal table or click for focus

```
...
public String testShortcut(List<String> shortcuts){
String s = null;
for(String str : shortcuts)
...
```

14) CTRL + E

Shows you the list of all open editors.



```
Service;


rvice<Comment> commentService) {
e;



owKey parentKey) {
t, parentKey);



owKey parentKey) {
arentKey);



owKey userKey) {
serKey);
```

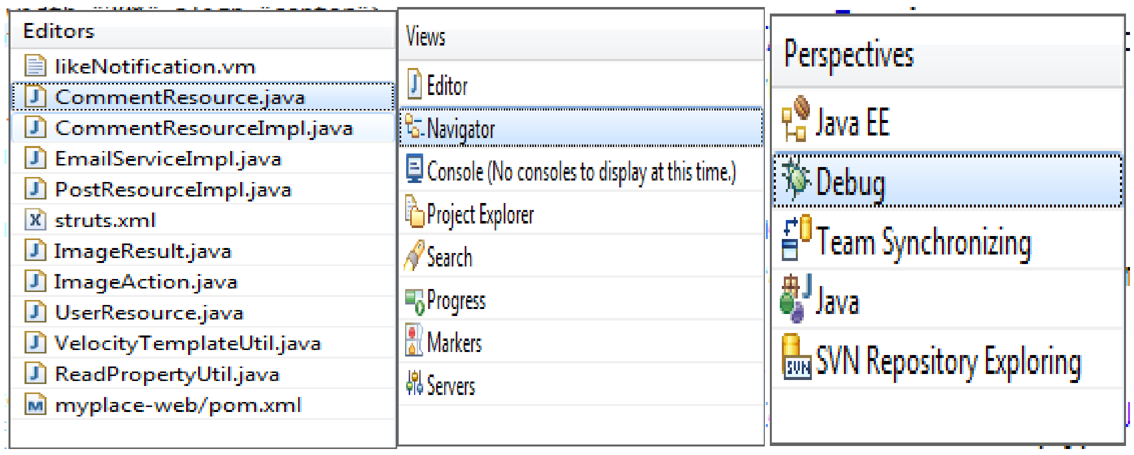
CommentResource.java
CommentResourceImpl.java
EmailServiceImpl.java
ImageAction.java
ImageResult.java
likeNotification.vm
myplace-web/pom.xml
PostResourceImpl.java
ReadPropertyUtil.java
struts.xml
UserResource.java
VelocityTemplateUtil.java

15) CTRL + F6 , CTRL + F7 , CTRL+F8

Ctrl+F6 - Move between open editors

Ctrl+F7 - Move between views

Ctrl+F8 - Move between perspectives



Editors
likeNotification.vm
CommentResource.java
CommentResourceImpl.java
EmailServiceImpl.java
PostResourceImpl.java
struts.xml
ImageResult.java
ImageAction.java
UserResource.java
VelocityTemplateUtil.java
ReadPropertyUtil.java
myplace-web/pom.xml

Views
Editor
Navigator
Console (No consoles to display at this time.)
Project Explorer
Search
Progress
Markers
Servers

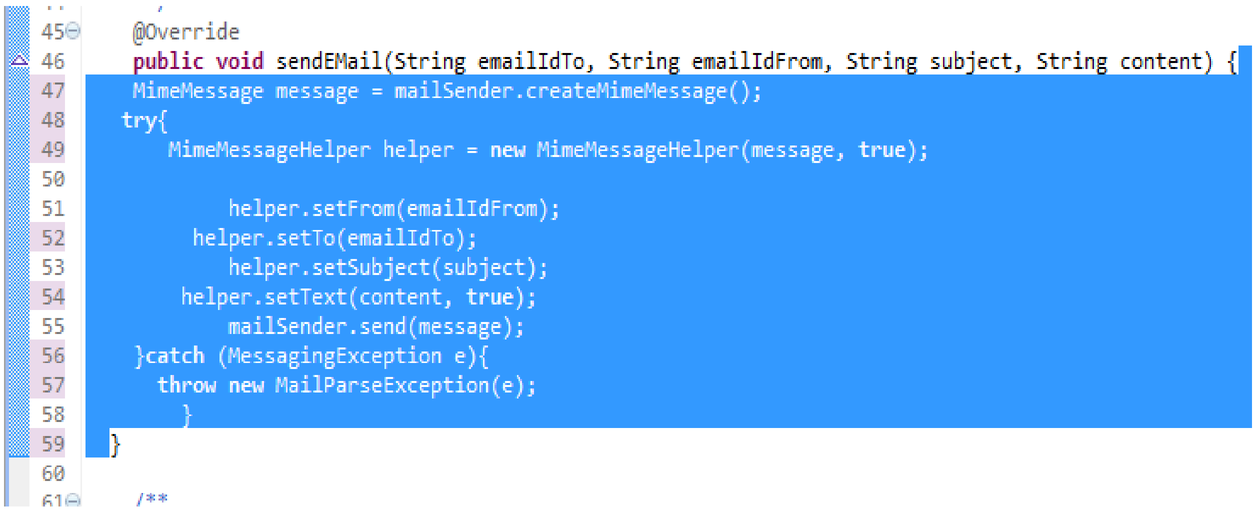
Perspectives
Java EE
Debug
Team Synchronizing
Java
SVN Repository Exploring

16) CTRL + M

Maximize or Un-maximize the current tab.

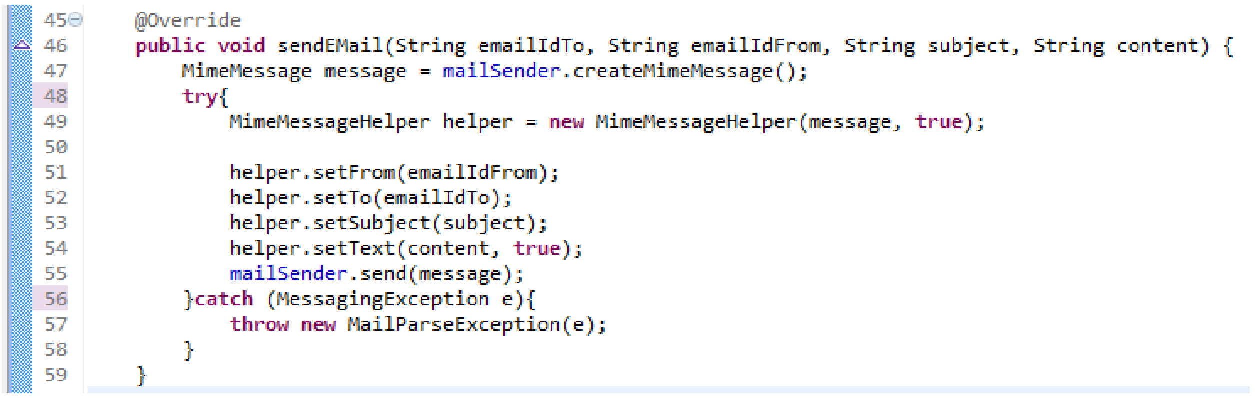
17) CTRL + I

Corrects indentation. Highlight the code, you want to correct the indentation for, and use this combination.



```
    @Override
    public void sendEMail(String emailIdTo, String emailIdFrom, String subject, String content) {
    MimeMessage message = mailSender.createMimeMessage();
   try{
        MimeMessageHelper helper = new MimeMessageHelper(message, true);

             helper.setFrom(emailIdFrom);
       helper.setTo(emailIdTo);
             helper.setSubject(subject);
      helper.setText(content, true);
             mailSender.send(message);
    }catch (MessagingException e){
      throw new MailParseException(e);
        }
  }

    /**
```

And after you use this combination you get



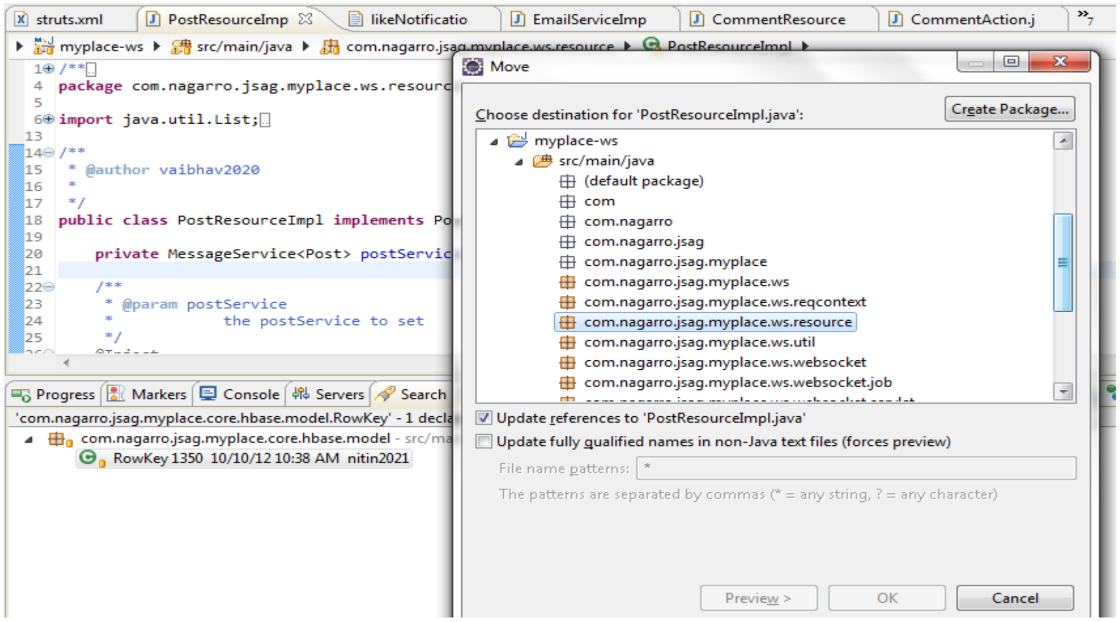
```
    @Override
    public void sendEMail(String emailIdTo, String emailIdFrom, String subject, String content) {
        MimeMessage message = mailSender.createMimeMessage();
        try{
            MimeMessageHelper helper = new MimeMessageHelper(message, true);

            helper.setFrom(emailIdFrom);
            helper.setTo(emailIdTo);
            helper.setSubject(subject);
            helper.setText(content, true);
            mailSender.send(message);
        }catch (MessagingException e){
            throw new MailParseException(e);
        }
    }
```

18) CTRL + G

Searches for the highlighted text in workspace.

19) ALT + SHIFT + V

Move your resource to correct destination.



```
/**
package com.nagarro.jsag.myplace.ws.resourc

import java.util.List;

/**
 * @author vaibhav2020
 *
 */
public class PostResourceImpl implements Po

    private MessageService<Post> postServic

    /**
     * @param postService
     *            the postService to set
     */
```

'com.nagarro.jsag.myplace.core.hbase.model.RowKey' - 1 decla
com.nagarro.jsag.myplace.core.hbase.model - src/ma
RowKey 1350 10/10/12 10:38 AM nitin2021

Choose destination for 'PostResourceImpl.java':

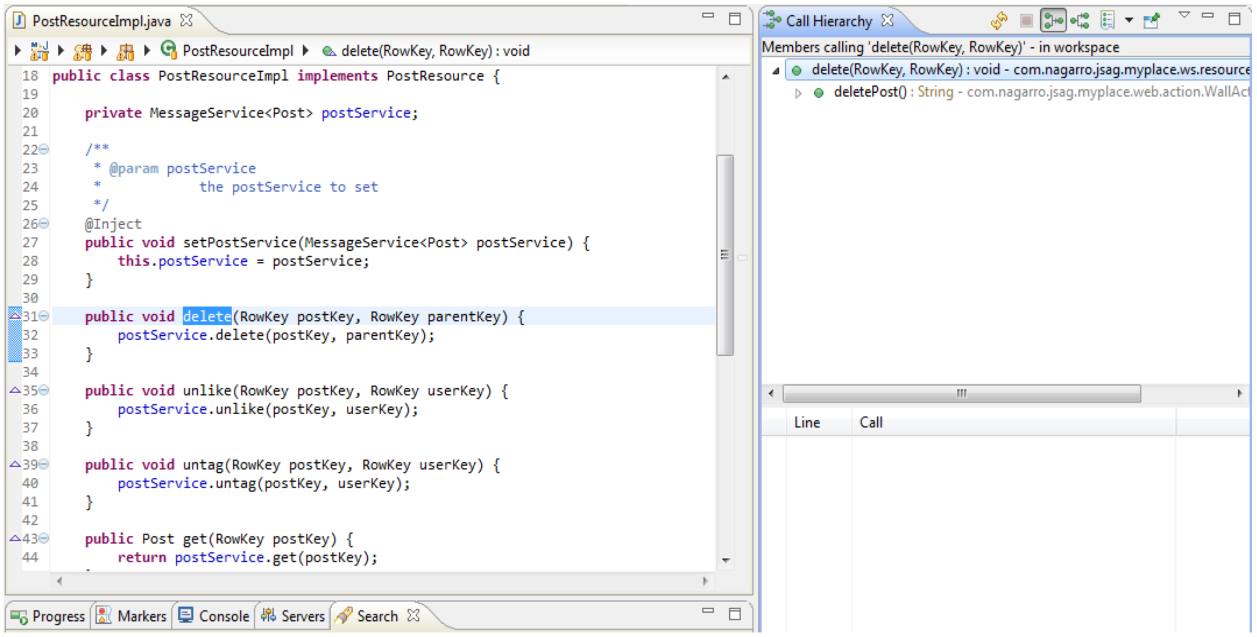
- myplace-ws
  - src/main/java
    - (default package)
    - com
    - com.nagarro
    - com.nagarro.jsag
    - com.nagarro.jsag.myplace
    - com.nagarro.jsag.myplace.ws
    - com.nagarro.jsag.myplace.ws.reqcontext
    - com.nagarro.jsag.myplace.ws.resource
    - com.nagarro.jsag.myplace.ws.util
    - com.nagarro.jsag.myplace.ws.websocket
    - com.nagarro.jsag.myplace.ws.websocket.job

[✓] Update references to 'PostResourceImpl.java'
[ ] Update fully qualified names in non-Java text files (forces preview)
File name patterns: *
The patterns are separated by commas (* = any string, ? = any character)

You can move your class to a new package, or any other existing package and eclipse will automatically update the references to your resource.

20) CTRL + ALT + H

Opens call hierarchy. Highlight the method and use this combination to find out where the method is called.

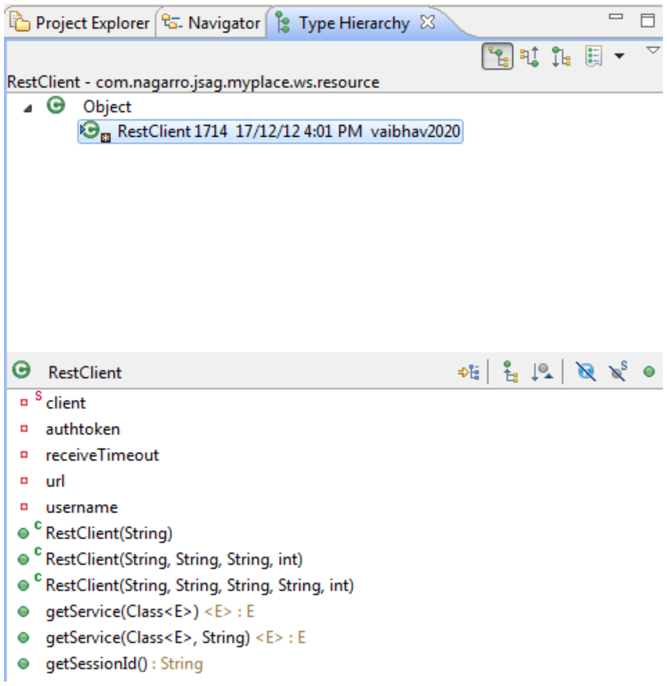


```
public class PostResourceImpl implements PostResource {

    private MessageService<Post> postService;

    /**
     * @param postService
     *            the postService to set
     */
    @Inject
    public void setPostService(MessageService<Post> postService) {
        this.postService = postService;
    }

    public void delete(RowKey postKey, RowKey parentKey) {
        postService.delete(postKey, parentKey);
    }

    public void unlike(RowKey postKey, RowKey userKey) {
        postService.unlike(postKey, userKey);
    }

    public void untag(RowKey postKey, RowKey userKey) {
        postService.untag(postKey, userKey);
    }

    public Post get(RowKey postKey) {
        return postService.get(postKey);
```

Members calling 'delete(RowKey, RowKey)' - in workspace
- delete(RowKey, RowKey) : void - com.nagarro.jsag.myplace.ws.resource
  - deletePost() : String - com.nagarro.jsag.myplace.web.action.WallAct

21) F4

Opens type hierarchy.



RestClient - com.nagarro.jsag.myplace.ws.resource
Object
RestClient 1714 17/12/12 4:01 PM vaibhav2020

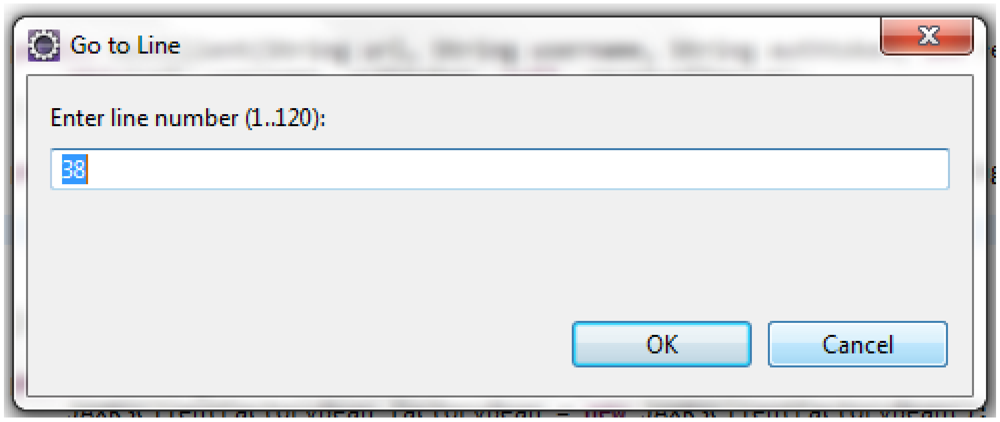
RestClient
client
authtoken
receiveTimeout
url
username
RestClient(String)
RestClient(String, String, String, int)
RestClient(String, String, String, String, int)
getService(Class<E>) <E> : E
getService(Class<E>, String) <E> : E
getSessionId() : String

22) F3

Takes you to the declaration of the method, variable or class. Use this while the cursor is on the method, variable or class name.

23) CTRL + L

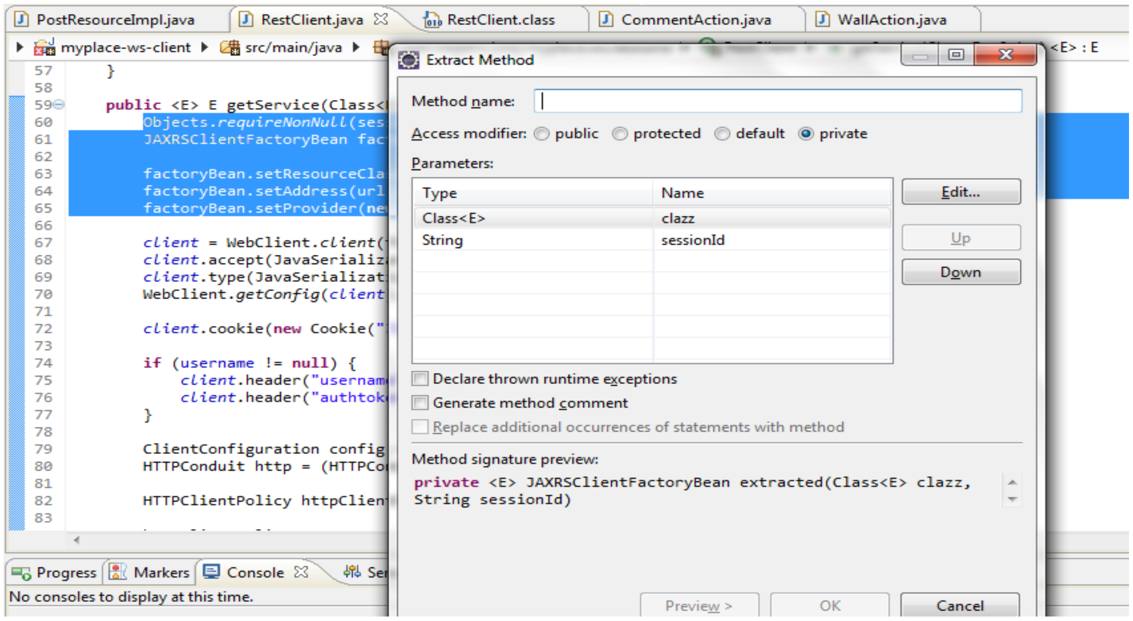
Make you reach to a line number.



Enter line number (1..120):
38

24) ALT + SHIFT + M

Most wonderful shortcuts of all. Helps you extract the highlighted code as a *separate method.*



```
    }

    public <E> E getService(Class<
        Objects.requireNonNull(ses
        JAXRSClientFactoryBean fac

        factoryBean.setResourceCla
        factoryBean.setAddress(url
        factoryBean.setProvider(ne

        client = WebClient.client(
        client.accept(JavaSerializ
        client.type(JavaSerializat
        WebClient.getConfig(client

        client.cookie(new Cookie("

        if (username != null) {
            client.header("usernam
            client.header("authtok
        }

        ClientConfiguration config
        HTTPConduit http = (HTTPCo

        HTTPClientPolicy httpClien
```

No consoles to display at this time.

Extract Method

Method name:

Access modifier: public protected default [✓ private]

Parameters:

Type	Name
Class<E>	clazz
String	sessionId

Edit... Up Down

☐ Declare thrown runtime exceptions
☐ Generate method comment
☐ Replace additional occurrences of statements with method

Method signature preview:

```
private <E> JAXRSClientFactoryBean extracted(Class<E> clazz,
String sessionId)
```

Preview > OK Cancel

It breaks up that monolithic code, takes the selection and tries to make it into a method. Will optionally find duplicate code blocks and use the new method there as well.

25) CTRL + W

Use this combination to close the current editor.

26) CTRL + SHIFT+ W

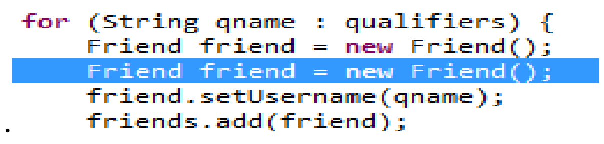
This combination will close all the open editors in eclipse.

27) CTRL + SHIFT+ C

To comment out the current line where the cursor is. This combination can also be used to un-comment any commented line

28) CTRL + ALT + UP

Copy current line above



```
for (String qname : qualifiers) {
    Friend friend = new Friend();
    Friend friend = new Friend();
    friend.setUsername(qname);
    friends.add(friend);
```

.

29) CTRL + ALT + DOWN

Copy the line below the current line.

30) CTRL + SHIFT + DELETE

Delete the from the cursor position till the end.

31) CTRL + SHIFT + P

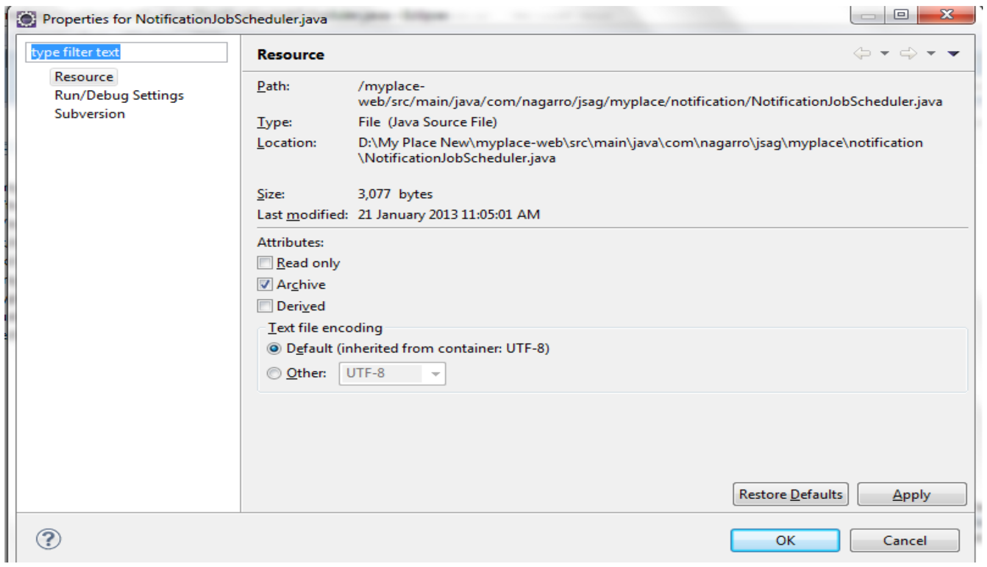
A good shortcut to use. It matches the correct brace. From opening brace, when you use this shortcut, it takes you to the matching closing bracket and vice-versa.

32) CTRL+SHIFT+ UP/DOWN

Move among class members. Be it variables or method. Using these shortcuts, you can walk through class members sequentially up or down by pressing the corresponding arrow keys.

33) ALT+ENTER

Most useful shortcut. It opens the properties tab of the class.



Properties for NotificationJobScheduler.java

type filter text

Resource
Run/Debug Settings
Subversion

Resource

Path: /myplace-web/src/main/java/com/nagarro/jsag/myplace/notification/NotificationJobScheduler.java
Type: File (Java Source File)
Location: D:\My Place New\myplace-web\src\main\java\com\nagarro\jsag\myplace\notification\NotificationJobScheduler.java

Size: 3,077 bytes
Last modified: 21 January 2013 11:05:01 AM

Attributes:
- [ ] Read only
- [✓] Archive
- [ ] Derived

Text file encoding
- [✓ Default (inherited from container: UTF-8)]
- Other: UTF-8

34) CTRL + DELETE

Delete the next word of the line from cursor position.

35) CTRL + BACKSPACE

Delete the previous word of the line.

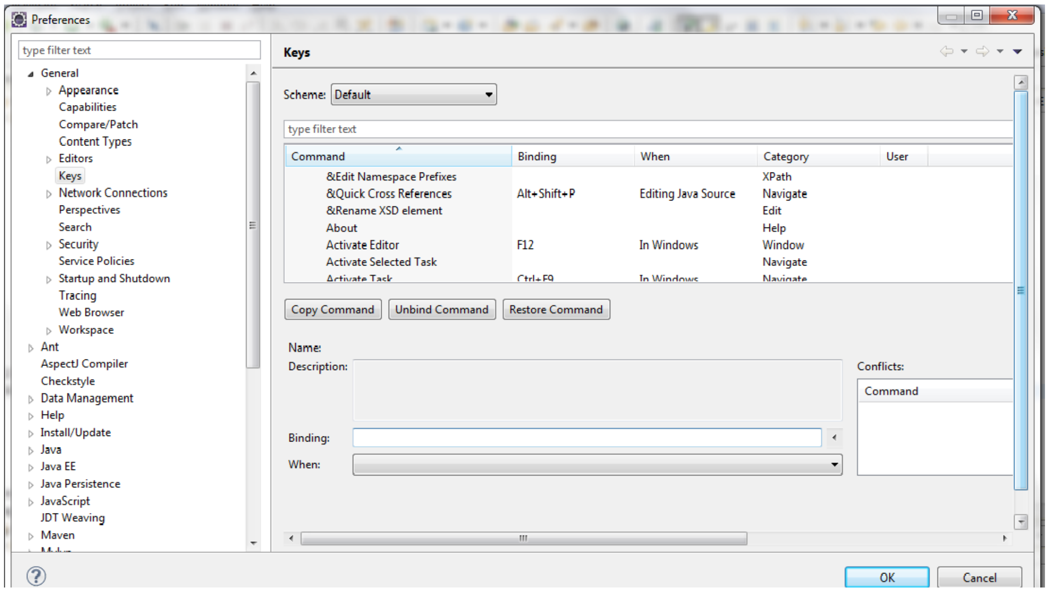
36) ALT + UP/DOWN

Shifts the current line up or down.

Customizing Shortcuts

You can always change the bindings to match them to your preference.

Open ***Windows->Preferences->General->Keys***. Now you can use the filter to find your shortcut and change its binding.



Scheme: Default

type filter text

Command	Binding	When	Category	User
&Edit Namespace Prefixes			XPath	
&Quick Cross References	Alt+Shift+P	Editing Java Source	Navigate	
&Rename XSD element			Edit	
About			Help	
Activate Editor	F12	In Windows	Window	
Activate Selected Task			Navigate	
Activate Task	Ctrl+F9	In Windows	Navigate	

Name:
Description:
Binding:
When:
Conflicts:
Command

Code Templates

Code templates are certain user defined (some are available by default) templates which can assist in rapid code writing. Not only code templates increase your code writing speed, but it also adds consistency across the code.

Code template can be generated by writing few characters and pressing "***CTRL + SPACE***". For example, you write "tr" and press ctrl+space and you'll have a try-catch block.

Top 10 Templates

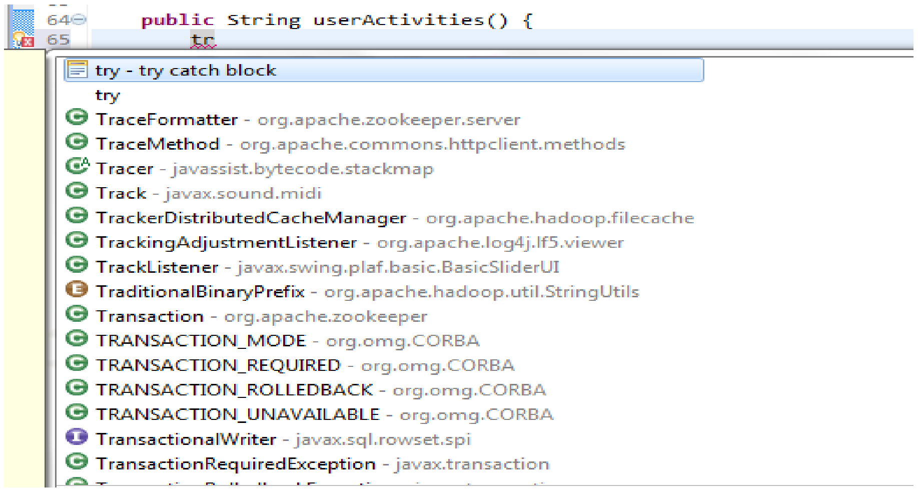
Most widely used templates are

1) syso

This will generate a print statement i.e “System.out.println()”. No need write it completely.

2) tr

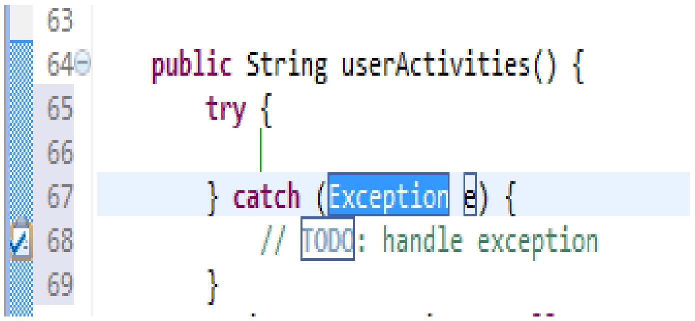
This will generate the try-catch block. The cursor will be at the Exception so that you can edit the Exception as per your requirement.



```
    public String userActivities() {
        tr
```

```
try - try catch block
try
TraceFormatter - org.apache.zookeeper.server
TraceMethod - org.apache.commons.httpclient.methods
Tracer - javassist.bytecode.stackmap
Track - javax.sound.midi
TrackerDistributedCacheManager - org.apache.hadoop.filecache
TrackingAdjustmentListener - org.apache.log4j.lf5.viewer
TrackListener - javax.swing.plaf.basic.BasicSliderUI
TraditionalBinaryPrefix - org.apache.hadoop.util.StringUtils
Transaction - org.apache.zookeeper
TRANSACTION_MODE - org.omg.CORBA
TRANSACTION_REQUIRED - org.omg.CORBA
TRANSACTION_ROLLEDBACK - org.omg.CORBA
TRANSACTION_UNAVAILABLE - org.omg.CORBA
TransactionalWriter - javax.sql.rowset.spi
TransactionRequiredException - javax.transaction
```

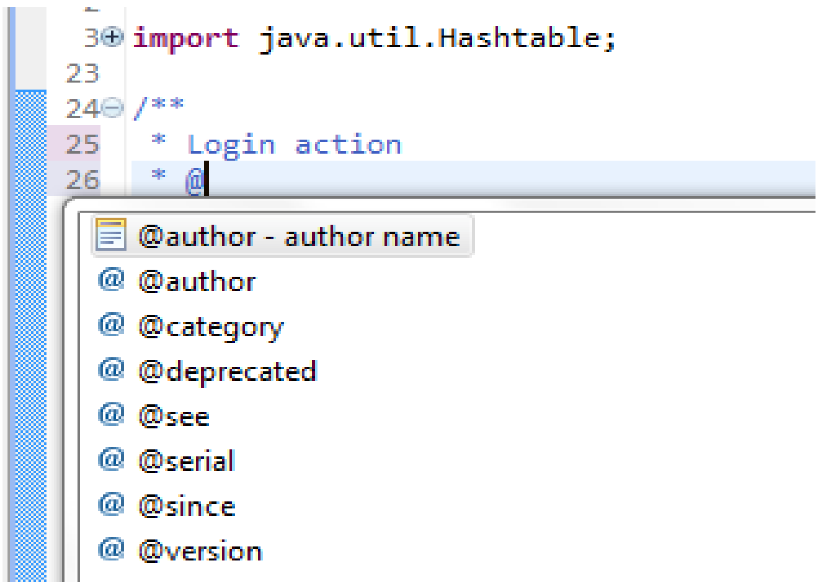
and after pressing enter you get



```
    public String userActivities() {
        try {

        } catch (Exception e) {
            // TODO: handle exception
        }
```

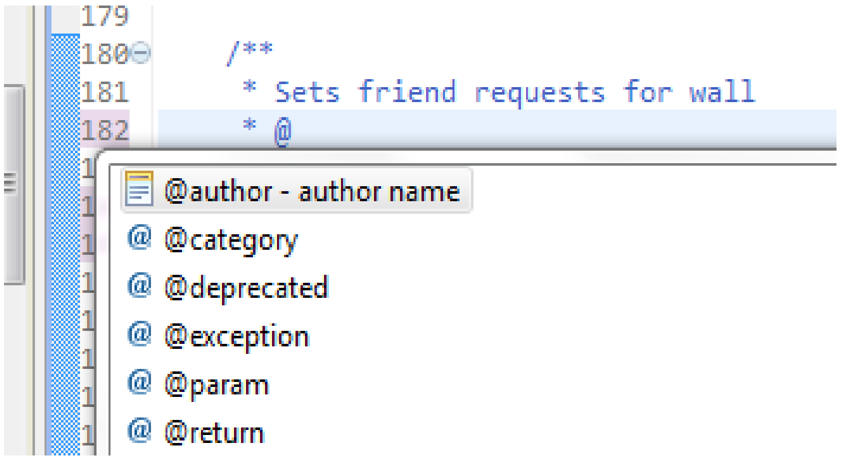
3) @

Very useful code template. Use this on class comment block to set author name.



```
import java.util.Hashtable;

/**
 * Login action
 * @
```

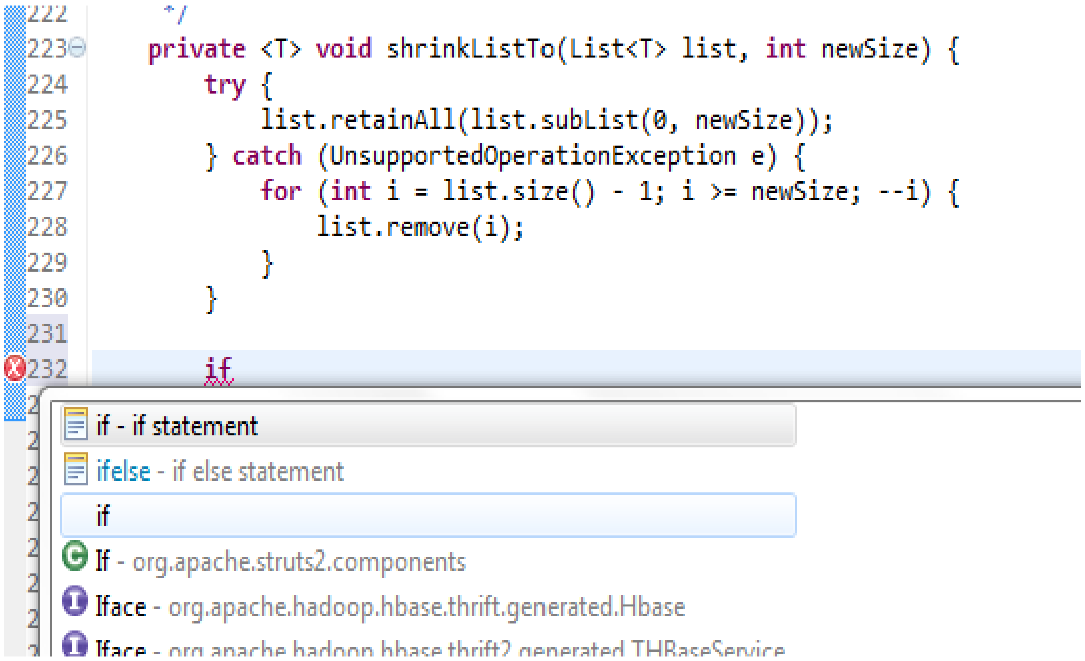
Provides various annotations such as author, version etc. Using this over the method's comment block will allow you to add 'param', 'return', 'Exception' annotations as well



```
    /**
     * Sets friend requests for wall
     * @
```

4) If, If/Else

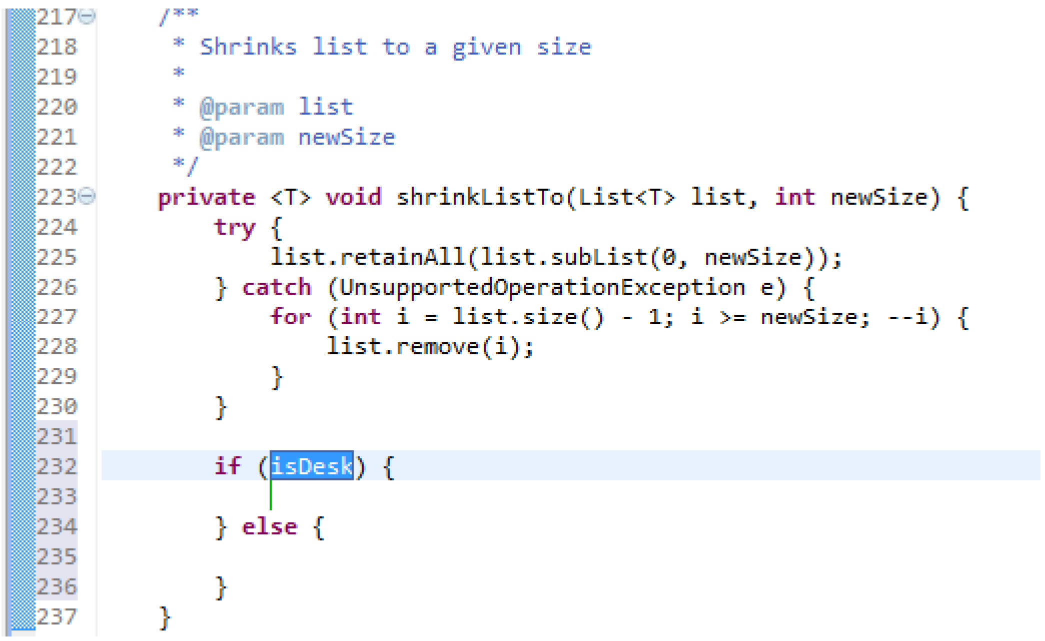
This will allow you to generate If/else block in your code.



```
     */
    private <T> void shrinkListTo(List<T> list, int newSize) {
        try {
            list.retainAll(list.subList(0, newSize));
        } catch (UnsupportedOperationException e) {
            for (int i = list.size() - 1; i >= newSize; --i) {
                list.remove(i);
            }
        }

        if
```

if - if statement
ifelse - if else statement
if
If - org.apache.struts2.components
Iface - org.apache.hadoop.hbase.thrift.generated.Hbase
Iface - org.apache.hadoop.hbase.thrift2.generated.THBaseService

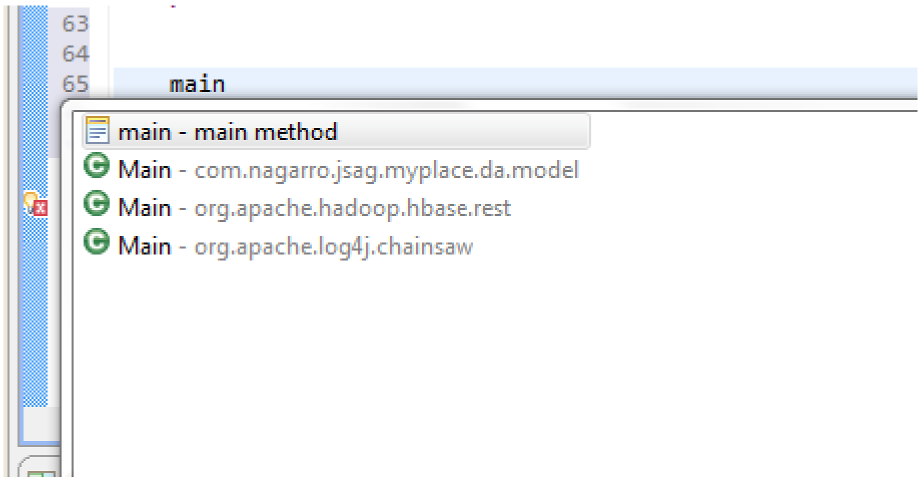
After generation, the cursor will point to the if condition to allow you to edit the condition as per your requirement.



```
    /**
     * Shrinks list to a given size
     *
     * @param list
     * @param newSize
     */
    private <T> void shrinkListTo(List<T> list, int newSize) {
        try {
            list.retainAll(list.subList(0, newSize));
        } catch (UnsupportedOperationException e) {
            for (int i = list.size() - 1; i >= newSize; --i) {
                list.remove(i);
            }
        }

        if (isDesk) {

        } else {

        }
    }
```

5) main

This code template will generate the main method.



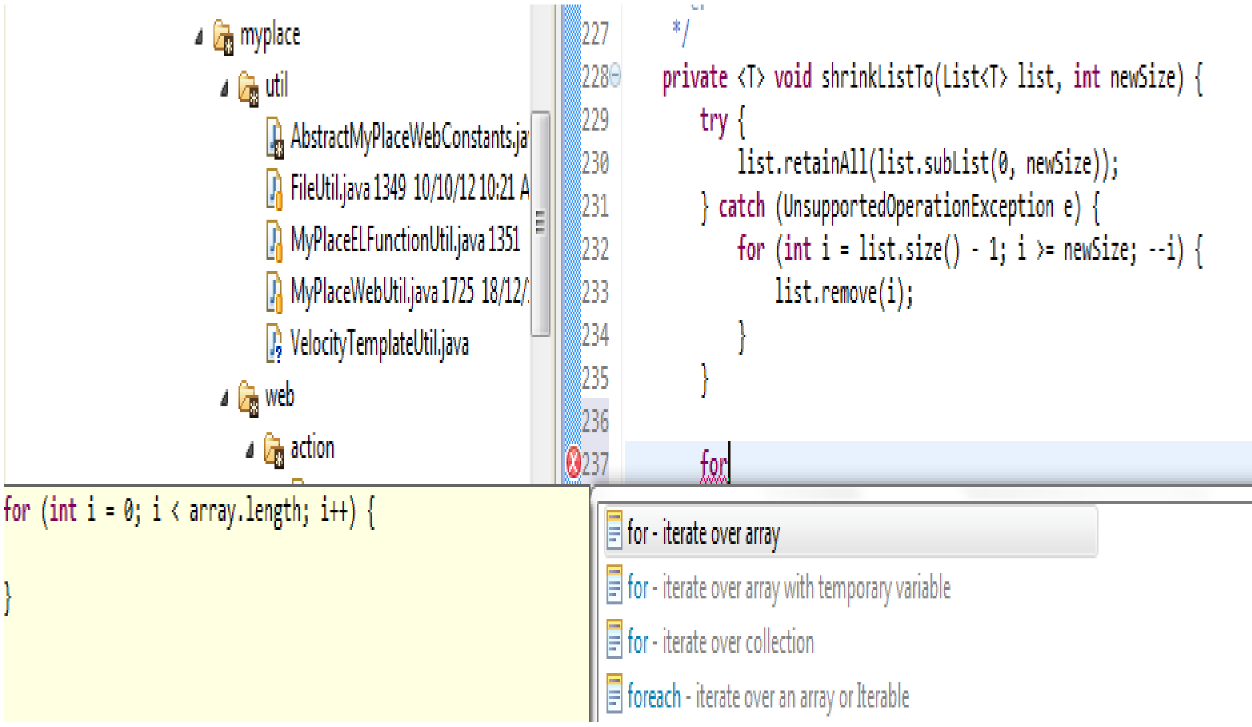
```
main
```

main - main method
Main - com.nagarro.jsag.myplace.da.model
Main - org.apache.hadoop.hbase.rest
Main - org.apache.log4j.chainsaw

6) loop statements

a. **for**

This will allow you to generate the “for loop” statement, and will give you the option whether you need to iterate over an array or over a collection and will generate the code accordingly.



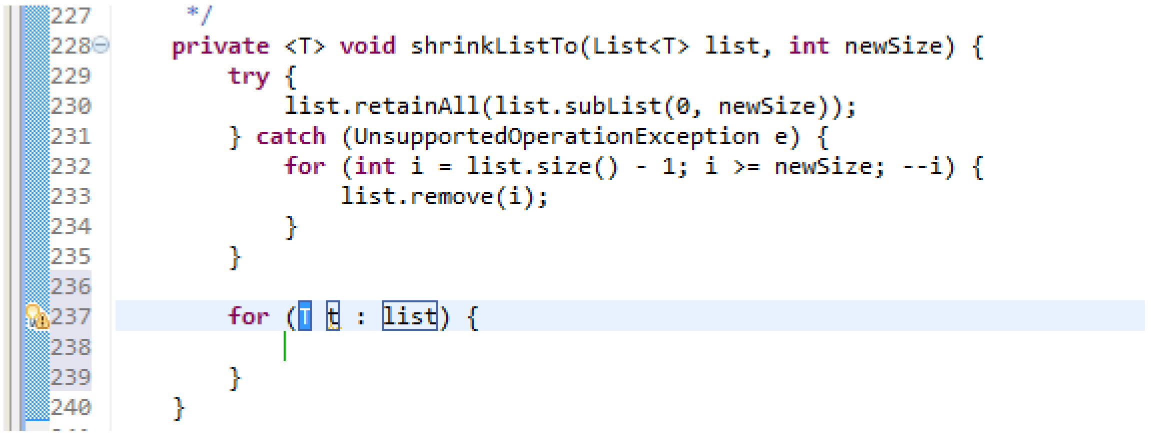
myplace
util
AbstractMyPlaceWebConstants.ja
FileUtil.java 1349 10/10/12 10:21 A
MyPlaceELFunctionUtil.java 1351
MyPlaceWebUtil.java 1725 18/12/
VelocityTemplateUtil.java
web
action

```
 */
private <T> void shrinkListTo(List<T> list, int newSize) {
    try {
        list.retainAll(list.subList(0, newSize));
    } catch (UnsupportedOperationException e) {
        for (int i = list.size() - 1; i >= newSize; --i) {
            list.remove(i);
        }
    }

    for
```

```
for (int i = 0; i < array.length; i++) {

}
```

for - iterate over array
for - iterate over array with temporary variable
for - iterate over collection
foreach - iterate over an array or Iterable

If you select option “foreach”, you’ll have something like

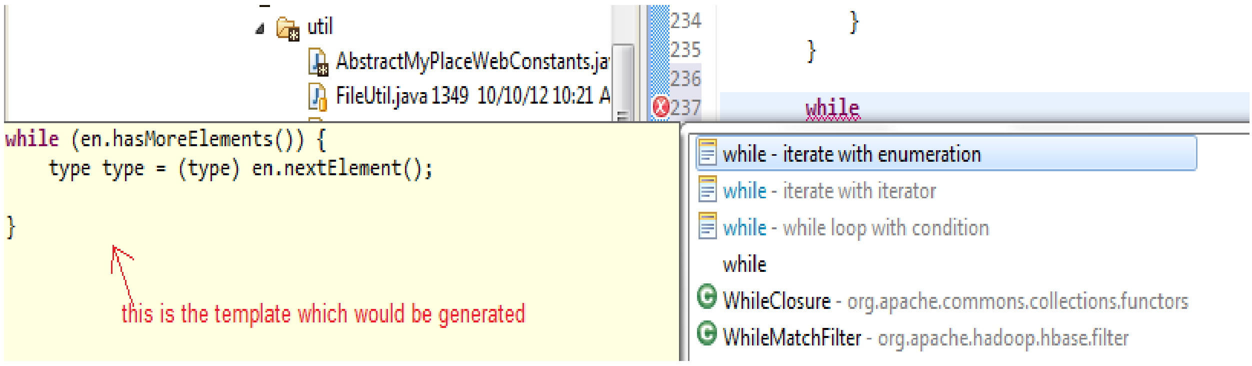


```
 */
private <T> void shrinkListTo(List<T> list, int newSize) {
    try {
        list.retainAll(list.subList(0, newSize));
    } catch (UnsupportedOperationException e) {
        for (int i = list.size() - 1; i >= newSize; --i) {
            list.remove(i);
        }
    }

    for (T t : list) {

    }
}
```

with cursor over the type to allow you to edit according to your requirement.

b. **while**

This will provide you with the options of iterating with an enum, or iterating over an iterator or iterating with a while loop condition.

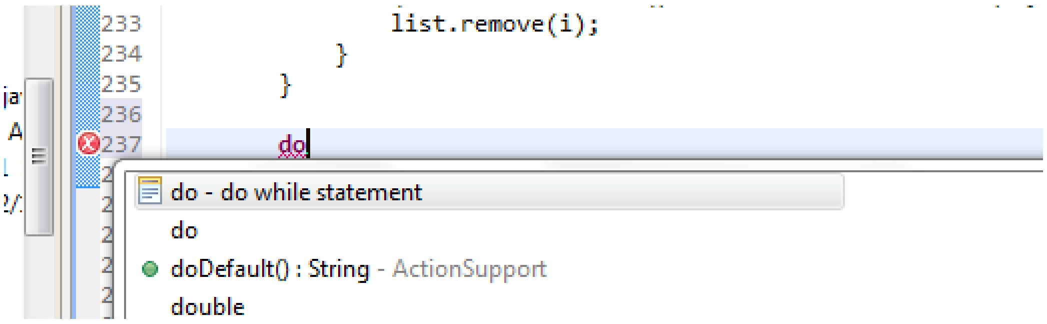


```
while (en.hasMoreElements()) {
    type type = (type) en.nextElement();

}
```

this is the template which would be generated
util
AbstractMyPlaceWebConstants.ja
FileUtil.java 1349 10/10/12 10:21 A
while
while - iterate with enumeration
while - iterate with iterator
while - while loop with condition
while
WhileClosure - org.apache.commons.collections.functors
WhileMatchFilter - org.apache.hadoop.hbase.filter

c. **do-while**

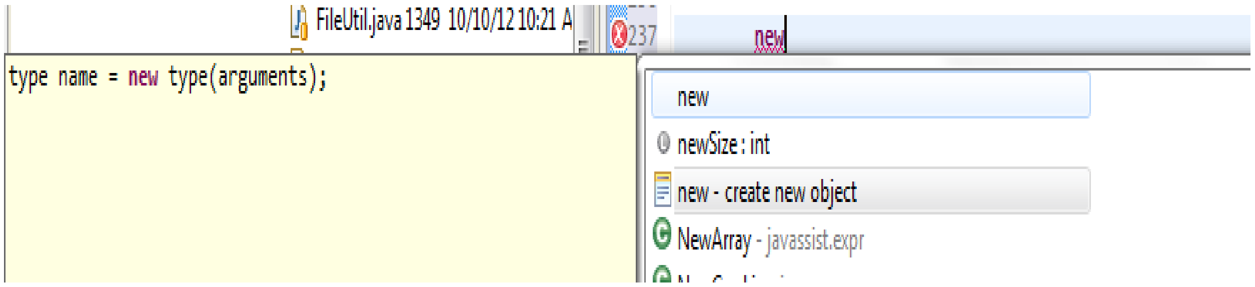
This will generate a do-while loop statement.



```
            list.remove(i);
        }
    }
    do
```

7) new

Allows you to create a new object of your type.

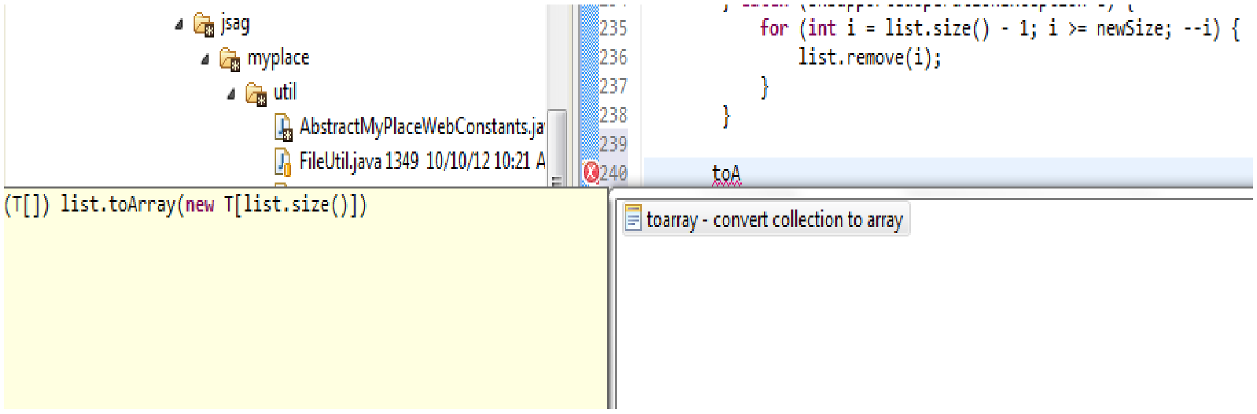


```
type name = new type(arguments);
```

FileUtil.java 1349 10/10/12 10:21 A
new
new
newSize : int
new - create new object
NewArray - javassist.expr

8) toArray

If you require to convert your collection to an array, you simply need to type these two or three characters and hit “ctrl+space”, this will generate a toArray statement for your collection.



```
for (int i = list.size() - 1; i >= newSize; --i) {
    list.remove(i);
}
}

toA
```

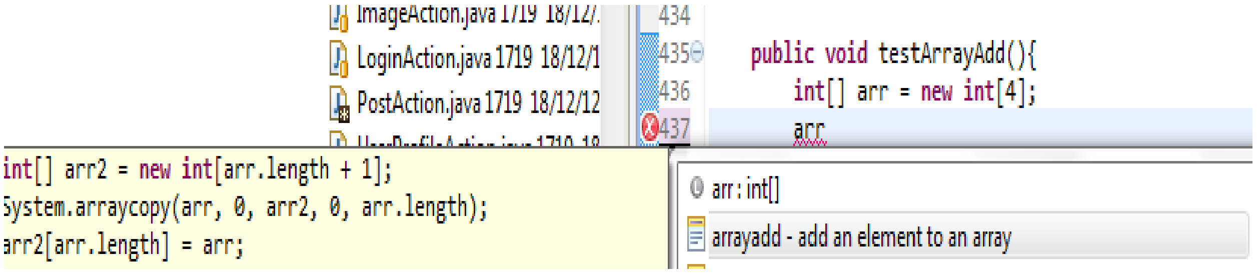
```
(T[]) list.toArray(new T[list.size()])
```

jsag
myplace
util
AbstractMyPlaceWebConstants.ja
FileUtil.java 1349 10/10/12 10:21 A
toarray - convert collection to array

With cursor over the type T which you need to edit according to your need.

9) arrayadd

Allows you to add an element to your array. Type arr and "ctrl+space" will show you the template



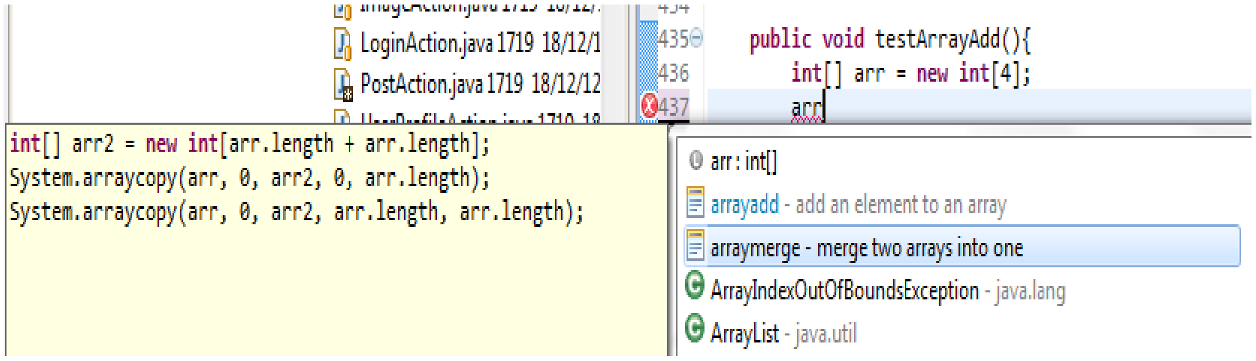
```
public void testArrayAdd(){
    int[] arr = new int[4];
    arr
```

```
int[] arr2 = new int[arr.length + 1];
System.arraycopy(arr, 0, arr2, 0, arr.length);
arr2[arr.length] = arr;
```

ImageAction.java 1719 18/12/
LoginAction.java 1719 18/12/1
PostAction.java 1719 18/12/12
arr : int[]
arrayadd - add an element to an array

10) arraymerge

You'll also find the option to merge merge two arrays into one.



```
public void testArrayAdd(){
    int[] arr = new int[4];
    arr
```

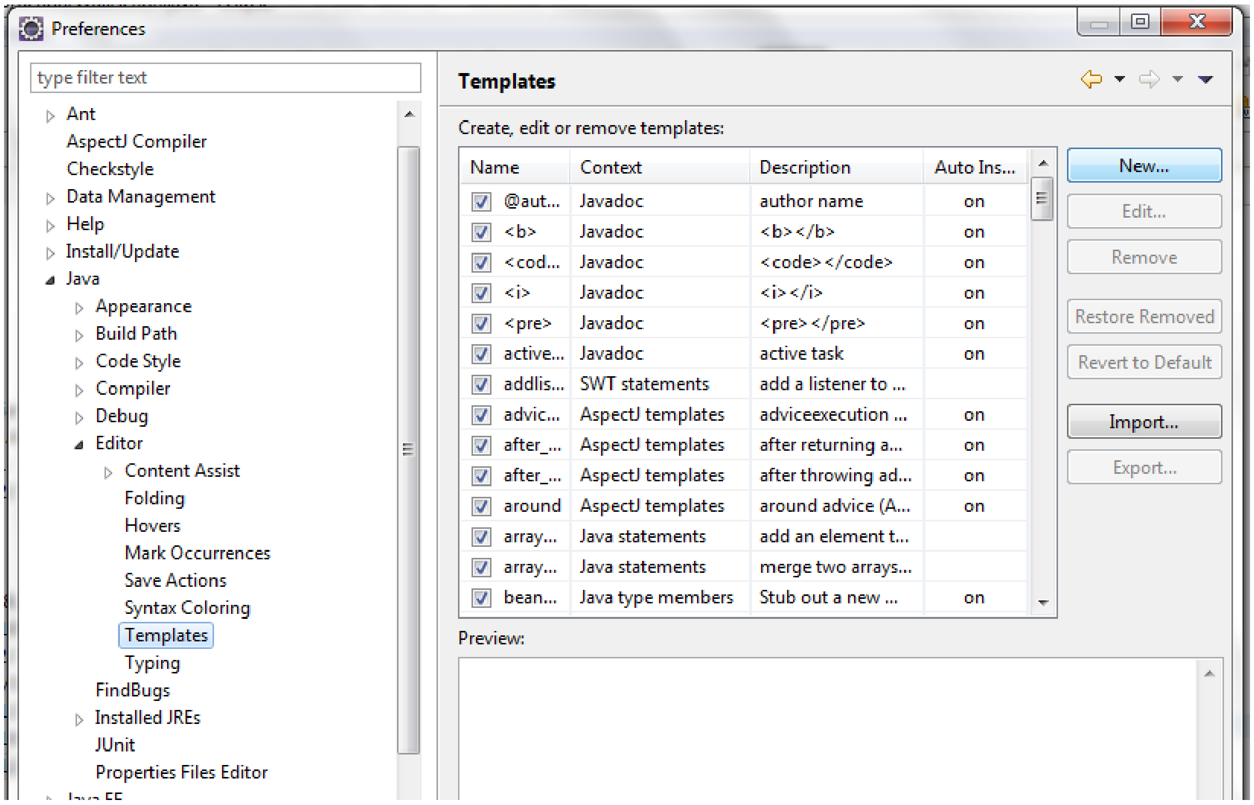
```
int[] arr2 = new int[arr.length + arr.length];
System.arraycopy(arr, 0, arr2, 0, arr.length);
System.arraycopy(arr, 0, arr2, arr.length, arr.length);
```

LoginAction.java 1719 18/12/1
PostAction.java 1719 18/12/12
arr : int[]
arrayadd - add an element to an array
arraymerge - merge two arrays into one
ArrayIndexOutOfBoundsException - java.lang
ArrayList - java.util

Customizing Templates

Apart from the default template provided in eclipse, you can also write your custom template in order to generate it easily using "*ctrl+space*". Here is how you can do it.

Just go to **Windows > Preferences > Java > Editor > Templates**.

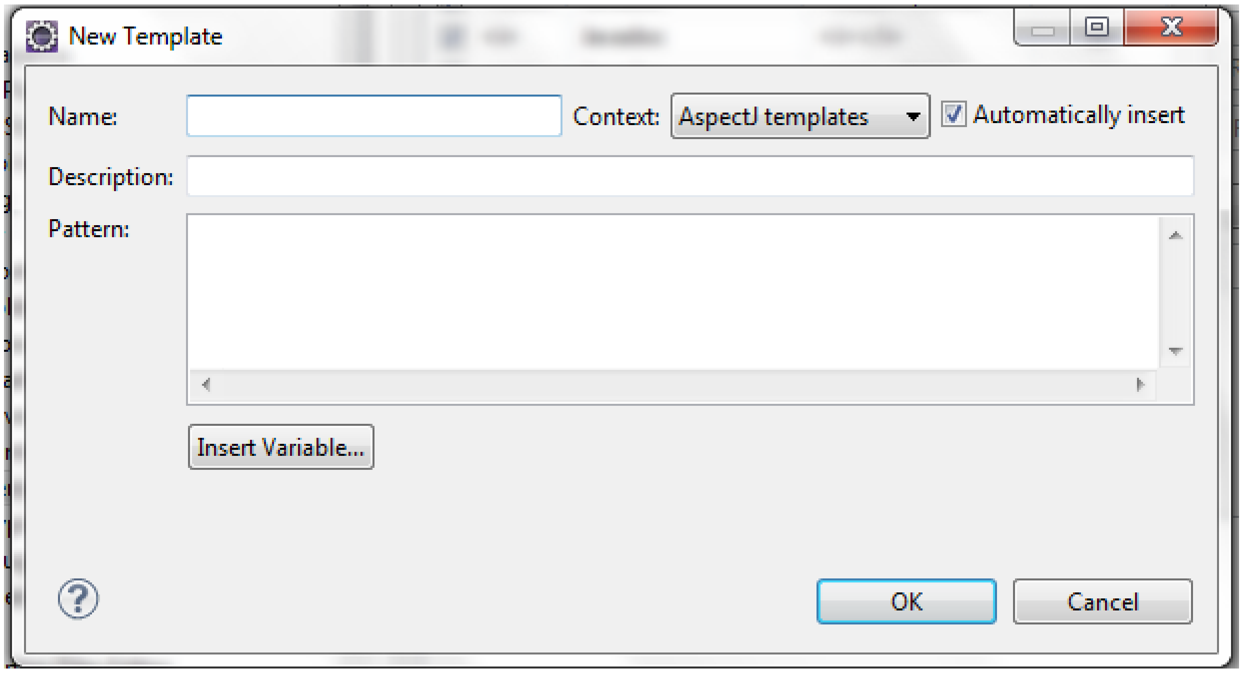


Create, edit or remove templates:

Name	Context	Description	Auto Ins...
✓ @aut...	Javadoc	author name	on
✓ <b>	Javadoc	<b></b>	on
✓ <cod...	Javadoc	<code></code>	on
✓ <i>	Javadoc	<i></i>	on
✓ <pre>	Javadoc	<pre></pre>	on
✓ active...	Javadoc	active task	on
✓ addlis...	SWT statements	add a listener to ...	
✓ advic...	AspectJ templates	adviceexecution ...	on
✓ after_...	AspectJ templates	after returning a...	on
✓ after_...	AspectJ templates	after throwing ad...	on
✓ around	AspectJ templates	around advice (A...	on
✓ array...	Java statements	add an element t...	
✓ array...	Java statements	merge two arrays...	
✓ bean...	Java type members	Stub out a new ...	on

We will create a template which will check if a local variable is null or not. If it is null, then throw "NullPointerException".

In Template window, click on new to add the code template and description.

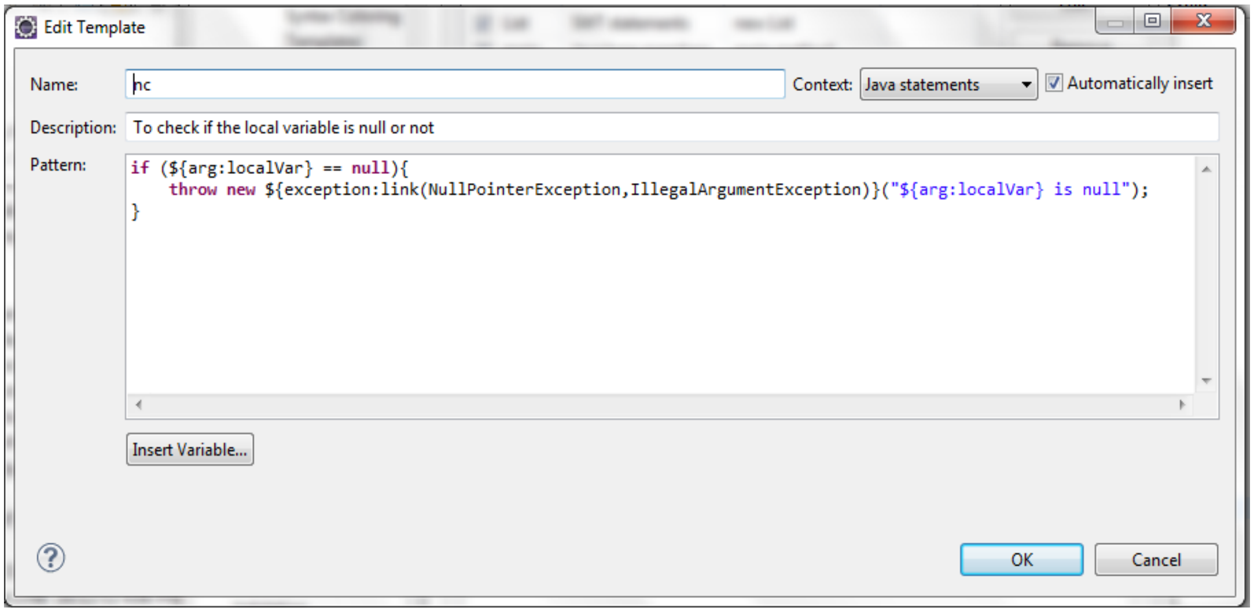


Fill up the details as following

Name: nc (let's call it 'null check')

Description: To check if the local variable is null or not.

Pattern : Enter as shown below



Name: nc

Context: Java statements

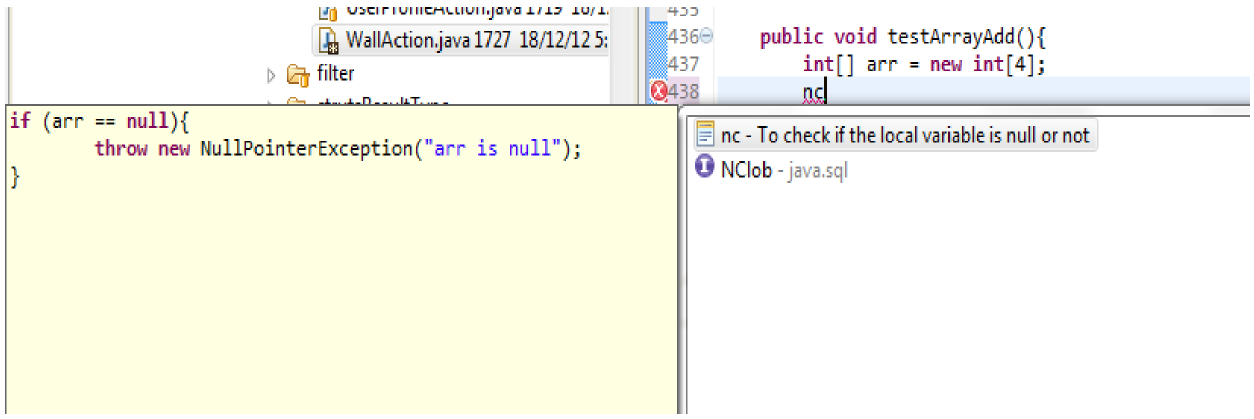
[✓] Automatically insert

Description: To check if the local variable is null or not

Pattern:

```
if (${arg:localVar} == null){
    throw new ${exception:link(NullPointerException,IllegalArgumentException)}("${arg:localVar} is null");
}
```

Click Ok to save your template. Now, go to any editor and type nc and hit "ctrl+space", you'll have something like



```
public void testArrayAdd(){
    int[] arr = new int[4];
    nc
```

```
if (arr == null){
        throw new NullPointerException("arr is null");
}
```

WallAction.java 1727 18/12/12 5:
filter
nc - To check if the local variable is null or not
NClob - java.sql

This way, you can add your own code templates to ease your work.

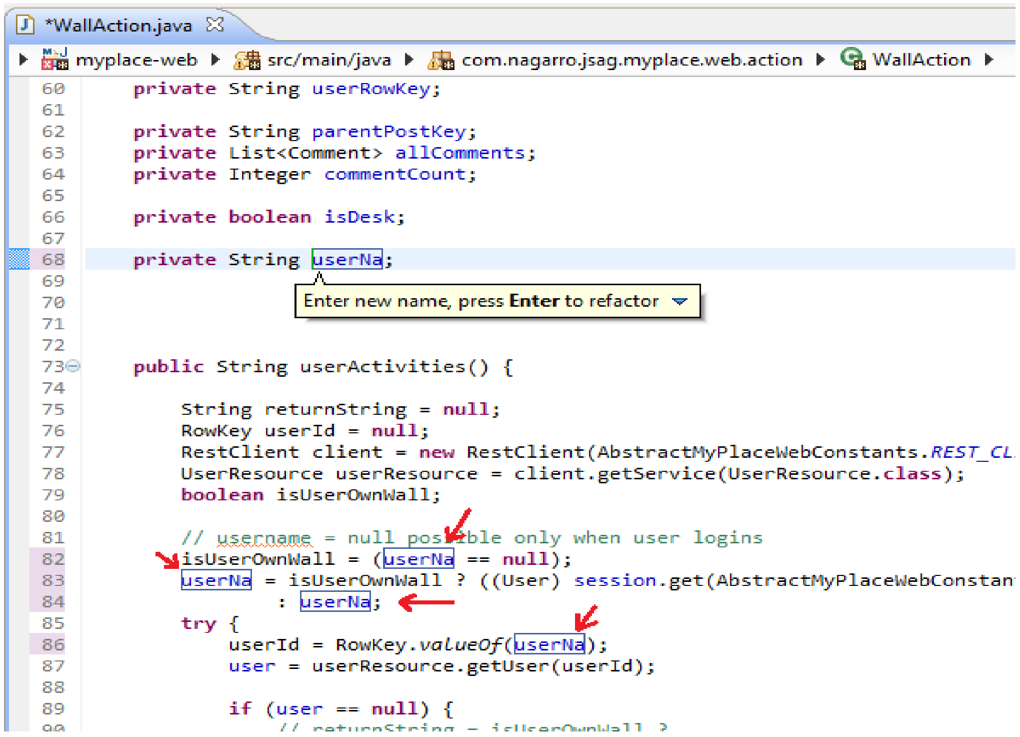
Usage Tips

Refactoring

Many a times it happens that we need to refactor our code or rename method or variable or even class. For this, there is an easy shortcut to do so.

ALT+SHIFT+R

Just highlight the thing you need to refactor, and use this combination, it will refactor all the references in the class or entire project as well.



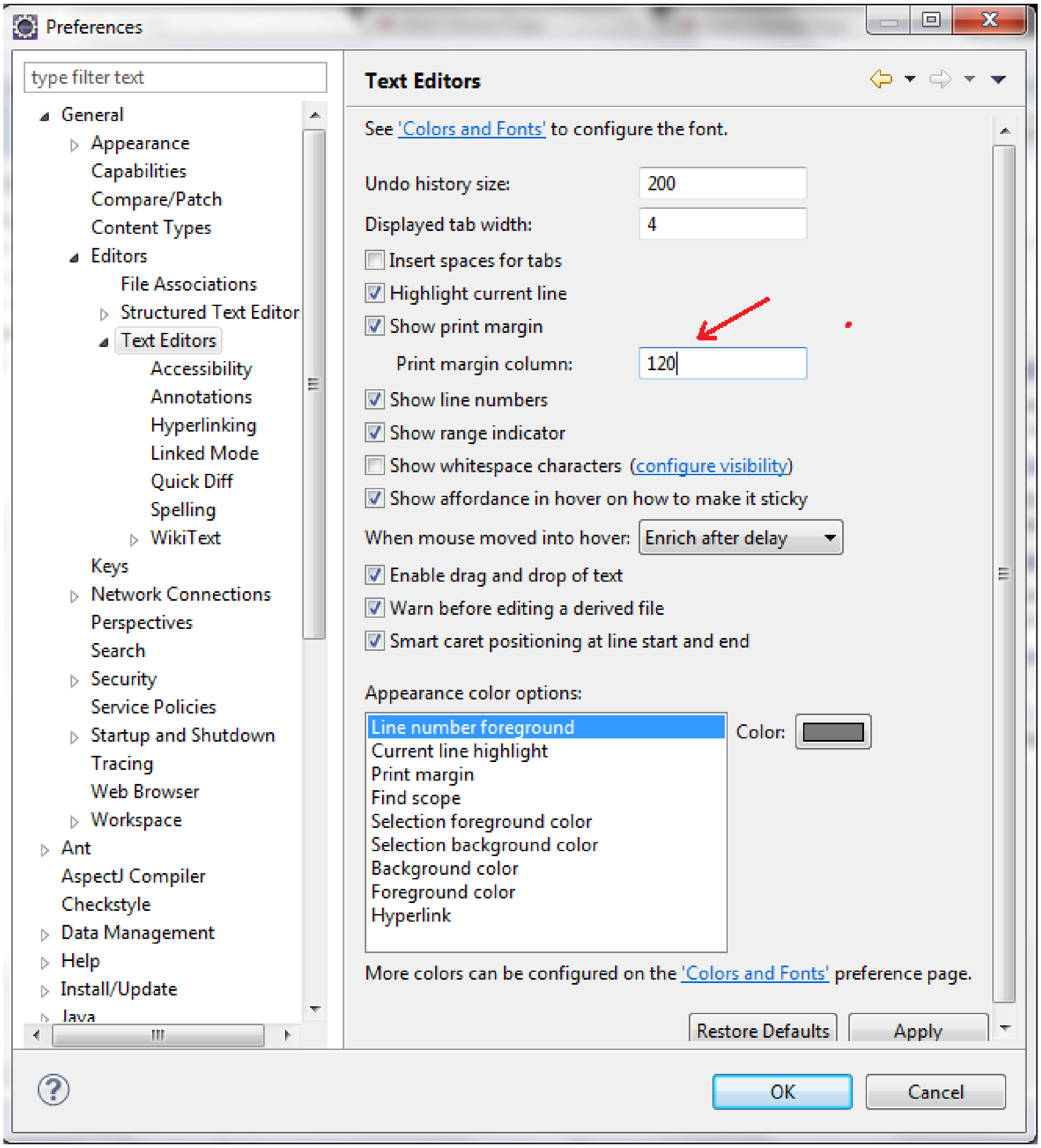
*WallAction.java
myplace-web ▸ src/main/java ▸ com.nagarro.jsag.myplace.web.action ▸ WallAction ▸
Enter new name, press Enter to refactor

```
    private String userRowKey;

    private String parentPostKey;
    private List<Comment> allComments;
    private Integer commentCount;

    private boolean isDesk;

    private String userNa;




    public String userActivities() {

        String returnString = null;
        RowKey userId = null;
        RestClient client = new RestClient(AbstractMyPlaceWebConstants.REST_CL
        UserResource userResource = client.getService(UserResource.class);
        boolean isUserOwnWall;

        // username = null possible only when user logins
        isUserOwnWall = (userNa == null);
        userNa = isUserOwnWall ? ((User) session.get(AbstractMyPlaceWebConstan
                : userNa;
        try {
            userId = RowKey.valueOf(userNa);
            user = userResource.getUser(userId);

            if (user == null) {
```

Show Print Margin

Now, you can write code without worrying about the standard '***120 character line***' margin. Under preferences, we can make setting to show margin after 120 characters and by that we can easily take care that we do not violate this standard coding guideline.

Go to **windows->preferences->general->editors->Text Editors**



Text Editors

See 'Colors and Fonts' to configure the font.

Undo history size: 200

Displayed tab width: 4

- [ ] Insert spaces for tabs
- [x] Highlight current line
- [x] Show print margin

Print margin column: 120

- [x] Show line numbers
- [x] Show range indicator
- [ ] Show whitespace characters (configure visibility)
- [x] Show affordance in hover on how to make it sticky

When mouse moved into hover: Enrich after delay

- [x] Enable drag and drop of text
- [x] Warn before editing a derived file
- [x] Smart caret positioning at line start and end

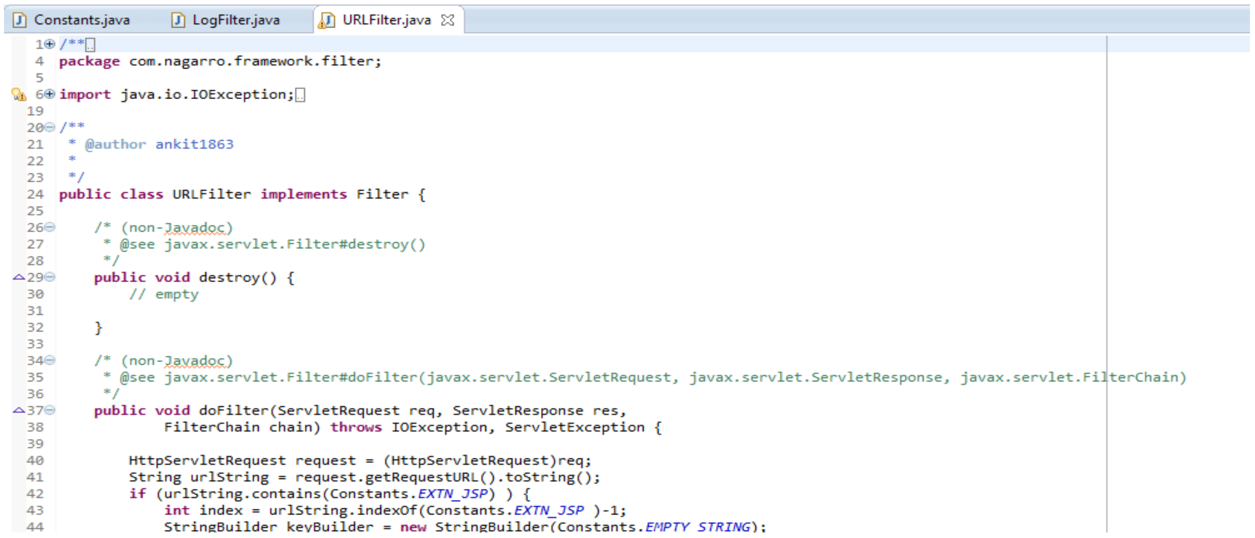
Appearance color options:

Line number foreground
Current line highlight
Print margin
Find scope
Selection foreground color
Selection background color
Background color
Foreground color
Hyperlink

Color:

More colors can be configured on the 'Colors and Fonts' preference page.

Click ok and you will see the margin after 120 characters, like below



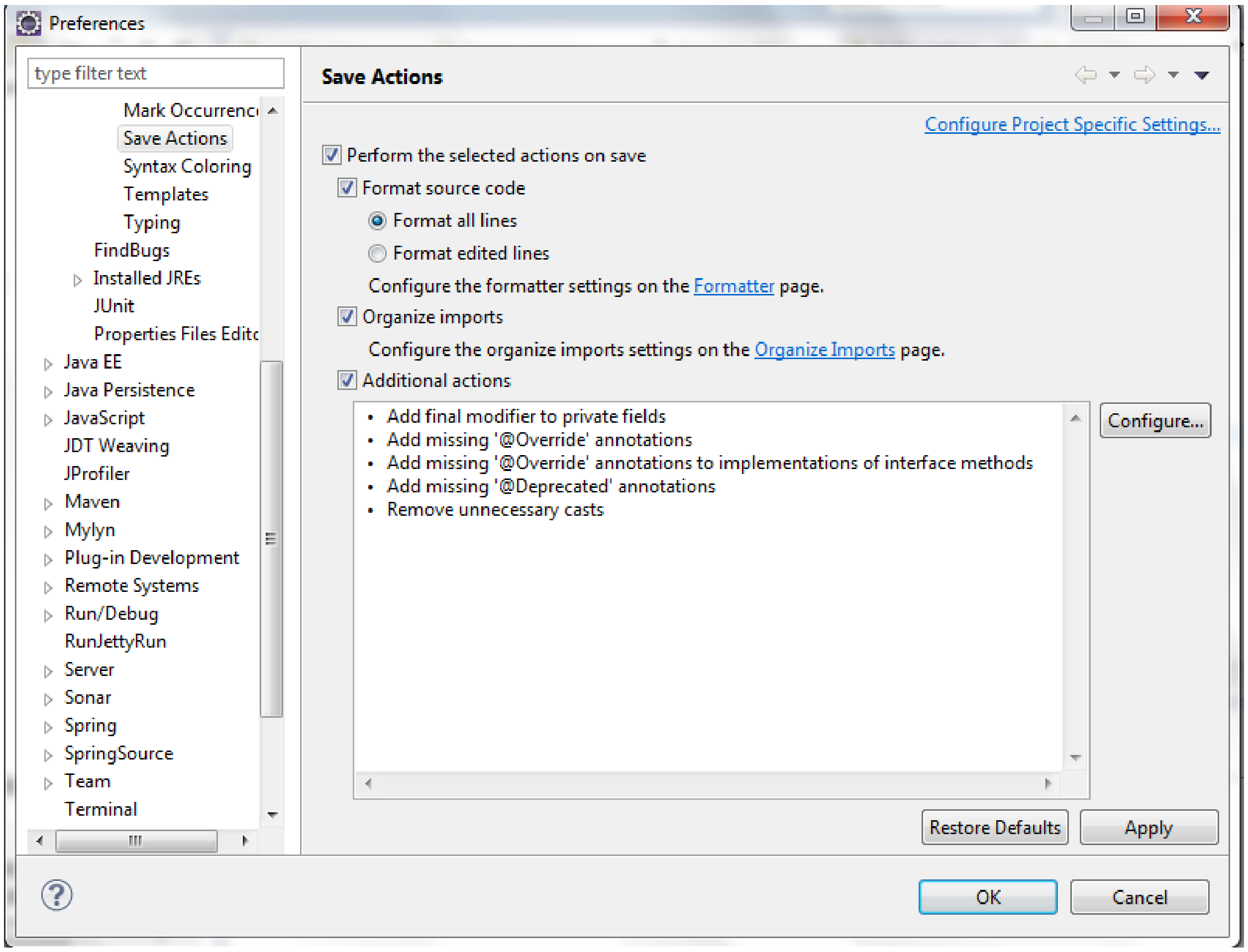
```
/**
package com.nagarro.framework.filter;

import java.io.IOException;

/**
 * @author ankit1863
 *
 */
public class URLFilter implements Filter {

    /* (non-Javadoc)
     * @see javax.servlet.Filter#destroy()
     */
    public void destroy() {
        // empty

    }

    /* (non-Javadoc)
     * @see javax.servlet.Filter#doFilter(javax.servlet.ServletRequest, javax.servlet.ServletResponse, javax.servlet.FilterChain)
     */
    public void doFilter(ServletRequest req, ServletResponse res,
            FilterChain chain) throws IOException, ServletException {

        HttpServletRequest request = (HttpServletRequest)req;
        String urlString = request.getRequestURL().toString();
        if (urlString.contains(Constants.EXTN_JSP) ) {
            int index = urlString.indexOf(Constants.EXTN_JSP )-1;
            StringBuilder keyBuilder = new StringBuilder(Constants.EMPTY_STRING);
```

Automatic Code cleanup and formatting

Eclipse also provides a feature called **Save Actions** which allows you to format your code every time when you save. You don't have to press different shortcuts for formatting, organizing imports and other clean ups, just enable the Save Actions and eclipse will do it for you when you save your file.

Below are the steps to enable auto code formatting:

1. Go to **Window > Preferences > Java > Editor > Save Actions**.
2. Select **Perform** the **selected actions on save**.
3. Select **Format Source code**. Make sure Format all lines is selected.
4. Make sure **Organize imports** is selected.
5. Select **Additional actions**. You can also configure additional actions to set up according to your need.
6. Click Ok, edit some code, save it and watch Eclipse format it automatically.



Save Actions

[✓] Perform the selected actions on save

[✓] Format source code

(●) Format all lines

( ) Format edited lines

Configure the formatter settings on the Formatter page.

[✓] Organize imports

Configure the organize imports settings on the Organize Imports page.

[✓] Additional actions

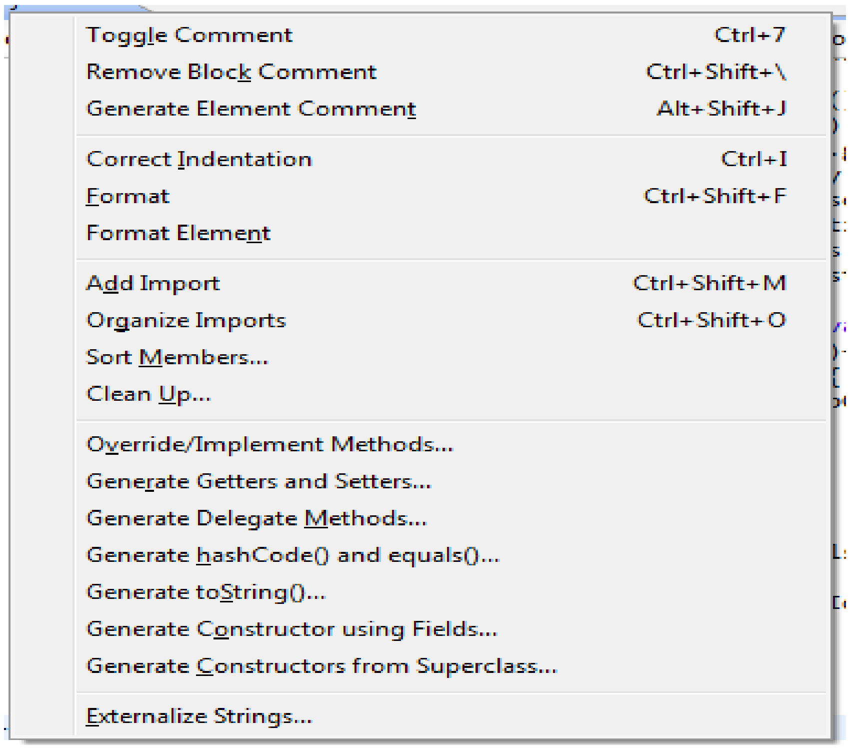
- Add final modifier to private fields
- Add missing '@Override' annotations
- Add missing '@Override' annotations to implementations of interface methods
- Add missing '@Deprecated' annotations
- Remove unnecessary casts

Code generation

There is a shortcut which will open a source window which provide few instant functionalities like generating getters and setters for instance fields, generating hashcode and equals method, generate toString method etc etc.

Alt + SHIFT + S

By using this combination, you'll see something like below where you can select functions of your choice.


Toggle Comment Ctrl+7
Remove Block Comment Ctrl+Shift+\
Generate Element Comment Alt+Shift+J
Correct Indentation Ctrl+I
Format Ctrl+Shift+F
Format Element
Add Import Ctrl+Shift+M
Organize Imports Ctrl+Shift+O
Sort Members...
Clean Up...
Override/Implement Methods...
Generate Getters and Setters...
Generate Delegate Methods...
Generate hashCode() and equals()...
Generate toString()...
Generate Constructor using Fields...
Generate Constructors from Superclass...
Externalize Strings...

References

You can find more stuff using these references

1) http://stackoverflow.com/questions/1028858/useful-eclipse-java-code-templates
2) http://www.vogella.com/articles/EclipseShortcuts/article.html
3) http://www.shortcutworld.com/en/win/Eclipse.html
4) http://eclipseone.wordpress.com/2009/12/13/automatically-format-and-cleanup-code-every-time-you-save/