Chapter 1: Using NetBeans Projects
public class Paperback implements Book {
    @Override
    public int getNumberOfPages() {
        throw new UnsupportedOperationException();
    }
}

package com.davidsalter.cookbook.hovtodebug;

public class HowToDebug {
    public static void main(String[] args) {
        Person person1 = new Person();
        Person person2 = new Person();

        person1.setName("David");
        person1.setAge(21);
        person2.setName(null);
        person2.setAge(32);
    }
}
package com.davidsalter.cookbook.howtodebug;

public class HowToDebug {

    public static void main(String[] args) {
        Person person1 = new Person();
        Person person2 = new Person();

        person1.setName("David");
        person1.setAge(21);
        person2.setName(null);
        person2.setAge(32);
    }

    /**
     * @param age the age to set
     */
    public void setAge(int age) {
        this.age = age;
    }
}
public static void main(String[] args) {
    int total = 0;
    int maximumCount = 10;

    // <editor-fold desc="Basic summing algorithm.">
    for (int i = 0; i < maximumCount; i++) {
        total += i;
    }
    // </editor-fold>

    System.out.println("Total: "+total);
}

public static void main(String[] args) {
    int total = 0;
    int maximumCount = 10;

    Basic summing algorithm.

    System.out.println("Total: "+total);
}
public interface Book {
    void read();
}

Implement Interface
Chapter 3: NetBeans Productivity

Generate Constructor

Select fields to be initialized by constructor:
- numberOfSides : int
- area : float

Generate toString()

Select fields to be included in toString():
- id : int
- firstName : String
- lastName : String
Add Property

Name: string
Type: String
- private
- Generate getter and setter
- Generate javadoc

Preview:
```java
private String string;
/**
 * Get the value of string
 * @return the value of string
 */
```

Generate Override Methods

Select methods to override:
- Object
  - equals(Object obj) : boolean
- finalize()
- hashCode() : int
- toString() : String

Generate
Usages of EnvironmentPrinter

Usages of Vehicle [1 occurrence]

- Refactoring
  - Car.java
    - static void print(String environmentVariableName)

Rerun Safely Delete

Move Members

- Move members from: EnvironmentPrinter
- Target: Refactoring
- Location: Source Packages
- Package: com.davidebolter.cookbook.refactor
- To Class: Main

Members to be moved:

- print(String environmentVariableName)

Visibility:
- As is
- Private
- Protected
- Public
- <default>

JavaDoc:
- As is
- Update

Keep original method(s) and delegate to the moved method:

- Deprecate the old method(s)

Preview Refactor Cancel Help

Safely Delete

The following warnings and errors were found. You can continue only with warnings.

List of Errors

- References to selected elements were found. Remove all references to these elements in order to safely delete your class.
Chapter 4: Developing Desktop Applications with NetBeans
Set Form's iconImage property using: Custom code

```
Form.setIconImage(new javax.swing.ImageIcon(getClass().getResource(""))
```

Snap at the preferred distance from the container's left border.
Snap at the preferred distance from the container's upper border.
To change layout manager of a container in the menu from its context menu.
private void viewButtonActionPerformed(java.awt.event.ActionEvent ev) {
    // TODO add your handling code here:
}
```java
viewButton.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        viewButtonActionPerformed(evt);
    }
});
```
Manifest-Version: 1.0
Ant-Version: Apache Ant 1.9.2
Created-By: 1.7.0_51-b13 (Oracle Corporation)
Class-Path: lib/derby.jar lib/derbyclient.jar lib/derbynet.jar

Main-Class: com.davidsalter.cookbook.jarviewer.Main

exitMenuItem [JMenu] - accelerator

Set exitMenuItem’s accelerator property using: Key stroke editor

Virtual Key: VK_F4
Key Stroke: Alt+F4
Chapter 5: NetBeans Enterprise Application Development

## Downloads

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Description</th>
<th>License</th>
<th>Size</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>8.0.0.Final</td>
<td>2014-02-11</td>
<td>Java EE7 Certified Server</td>
<td>LGPL</td>
<td>148 MB</td>
<td>ZIP</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Minimalistic Core Distribution</td>
<td>LGPL</td>
<td>134 MB</td>
<td>TGZ</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Application Server Source Code</td>
<td>LGPL</td>
<td>15 MB</td>
<td>ZIP</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>29 MB</td>
<td>ZIP</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>17 MB</td>
<td>TGZ</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Release Notes</td>
<td></td>
<td></td>
<td>Notes</td>
</tr>
</tbody>
</table>

- appclient
- bin
- docs
- domain
- modules
- standalone
- welcome-content
- copyright.txt
- jboss-modules.jar
- LICENSE.txt
- README.txt
Update Center Customizer

Name: Dev Update Center

- check for updates automatically

URL: 'deadlock.netbeans.org/job/nbms-and-javadoc/fastStableBuild/artifact/hbbuild/nbms/updates.xml'
Servers

Server Names: Apache Tomcat 8.0.3.0
Server Type: Apache Tomcat or TomEE

Connection:
- Startup
- Platform
- Deployment
- Classes
- Sources
- Javadoc

Catalina Home: C:\DevTools\Apache Tomcat 8.0.3
Catalina Base: C:\Users\david\AppData\Roaming\NetBeans\8.0\apache-tomcat-8.0.3.0_base

Credentials of an existing user in the "manager-script" role
Username: ide
Password: *********

Server Port: 8084
Shutdown Port: 8025

Enable HTTP Monitor

Note: Changes will take effect the next time you start the server

Add Server Instance

Steps
1. Choose Server
2. ...

Choose Server

Servers:
- Apache Tomcat or TomEE
- GlassFish Server
- JBoss Application Server
- Oracle WebLogic Server
- WildFly Application Server

Name: WildFly Application Server

< Back Next > Finish Cancel Help
Apache TomEE 1.7.0  August 2014

Web Profile
Apache TomEE Web Profile delivers Servlets, JSP, JSF, JTA, JPA, CDI, Bean Validation and EJB Lite

Java EE6 Certified

- [apache-tomee-1.7.0-webprofile.tar.gz (asc - md5)]
- [apache-tomee-1.7.0-webprofile.zip (asc - md5)]

Comparison »
Hello World!
Always catch exceptions.
New Message-Driven Bean

Steps
1. Choose File Type
2. Name and Location
3. Activation Config Properties

Name and Location
- EJB Name: CookbookQueueListenerBean
- Project: MessageBeans
- Location: Source Packages
- Package: com.davidalott.cookbook.messagebeans

○ Project Destinations: [Input Field]
○ Server Destinations: cookbookQueue

New File

Steps
1. Choose File Type
2. Name and Location
3. Activation Config Properties

Activation Config Properties
<table>
<thead>
<tr>
<th>Property Name</th>
<th>Property Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>acknowledgeMode</td>
<td>AUTO_ACKNOWLEDGE</td>
</tr>
<tr>
<td>clientId</td>
<td></td>
</tr>
<tr>
<td>connectionFactoryLookup</td>
<td></td>
</tr>
<tr>
<td>destinationType</td>
<td>QUEUE</td>
</tr>
<tr>
<td>destinationLookup</td>
<td></td>
</tr>
<tr>
<td>messageSelector</td>
<td></td>
</tr>
<tr>
<td>subscriptionDurability</td>
<td>NON_DURABLE</td>
</tr>
<tr>
<td>subscriptionName</td>
<td></td>
</tr>
</tbody>
</table>

SendMessageServlet

localhost:8080/MessageBeans/SendMessageServlet

Message sent at: Fri Apr 04 20:51:24 BST 2014
Send JMS Message

- **Server Destinations:** cookbookQueue
- **Connection Factory:** java:comp:DefaultJMSConnectionFactory
- **Service Locator Strategy:**
  - Generate Inline Lookup Code
  - Existing Class

Output:

```
21:41:04,642 INFO [ocs.jboss.as.server] (management-handler-thread - 8) JPA018942: Redeployed "Timers.war"
21:41:10,003 INFO [stdout] (EJB default - 6) Timer event: Fri Apr 04 21:41:10 BST 2014
21:41:30,001 INFO [stdout] (EJB default - 7) Timer event: Fri Apr 04 21:41:30 BST 2014
```
New RESTful Web Services from Patterns

Steps
1. Choose File Type
2. Select Pattern
3. Specify Resource Classes

Select Pattern
Select a RESTful web service design pattern:
- Simple Root Resource
- Container-Item
- Client-Controlled Container-Item

Description:
Create a RESTful root resource class with GET and PUT methods using Java API for RESTful Web Service (JSR-311). This pattern is useful for creating a simple HelloWorld service and wrapper services for invoking WSDL-based web services.

On the next page you will be specifying class name, URL, and representation type of the resource.

localhost:8080/BookService

[ json]
[
  - {
      name: "Moby Dick",
      author: "Herman Melville",
      isbn: "1"
  },
  - {
      name: "A Princess of Mars",
      author: "Edgar Rice Burroughs",
      isbn: "2"
  }
]

[0]
### Browser

**With NetBeans Connector**

- Embedded WebKit Browser
- Chrome

**Mobile Device Browser**

- Android Device (Chrome)
- Android Device (Default Browser)
- Android Emulator (Default Browser)

---

### Install Chrome Extension

NetBeans Connector Chrome Extension is required.

To run your project in Chrome with NetBeans Connector, you must first install the NetBeans Connector extension in the Chrome Web Store.

Go to Chrome Web Store

[Not connected?]
The NetBeans Connector provides deep integration between the award-winning NetBeans IDE and the Google Chrome browser.

With the release of NetBeans 7.3 comes a new feature for exciting and interesting with HTML, JavaScript, and CSS-based applications. This connector allows for tight integration with Google Chrome.

With this integration, comes the ability to:
- Refresh on Save
- Live DOM navigation from within the IDE itself
- Bi-directional element inspection. Click in the browser, see it in the IDE, and vice-versa
- JavaScript debugging of the application using remote WebKit APIs
- Visual CSS Style editing of all page elements, including JavaScript generated elements

Install Chrome Extension

Re-run your project after the extension is installed.

Re-Run Project

Unable to install the extension?

Cancel

To do...

Here's your tasks to complete.

Wow, you've got no tasks to complete!

Create New Task

(c) David Salter 2014
Forename

size must be between 1 and 25

New JSF Composite Component

File Name: inputWithLabel
Project: CompositeComp
Location: CompositeComp - Web Pages
Folder: resources\cookbook
Created File: c:users\NetBeansProjects\CompositeComp\web\resources\cookbook\inputWithLabel.xhtml
Prefix:

Implementation Section:

resources
cookbook
inputWithLabel.xhtml

Forename

size must be between 1 and 25

Last Name

size must be between 1 and 50

Submit
Chapter 6: Managing Databases with NetBeans
Customize Connection

Driver Name: PostgreSQL
Host: 127.0.0.1
Database: cookbook
User Name: postgres
Password: ********

JDBC URL: jdbc:postgresql://127.0.0.1:5432/cookbook

Connection Properties  Test Connection

< Back  Next >  Finish  Cancel  Help
Create Table

Table name: TASKS

<table>
<thead>
<tr>
<th>Key</th>
<th>Index</th>
<th>Null</th>
<th>Unique</th>
<th>Column name</th>
<th>Data type</th>
<th>Size</th>
<th>Scale</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td>TASK_ID</td>
<td>INTEGER</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>TASK_NAME</td>
<td>VARCHAR</td>
<td>25</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>TASK_COMPLETED</td>
<td>BOOLEAN</td>
<td>0</td>
<td>0</td>
<td>false</td>
</tr>
<tr>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td>TASK_DUE</td>
<td>DATE</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

OK  Cancel  Help

SQL: [jdbc:derby://localhost:1527/cookbook [APP on APP]]

1. insert into TASKS (TASK_ID, TASK_NAME) values (1, 'Read Cookbook');
2. insert into TASKS (TASK_ID, TASK_NAME) values (2, 'Eat sandwich');
3. insert into TASKS (TASK_ID, TASK_NAME, TASK_DUE) values (3, 'Do some programming', '2015-03-01');

SQL: [jdbc:derby://localhost:1527/cookbook [APP on APP]]

```sql
select * from APP.TASKS;
```

select * from APP.TASKS

<table>
<thead>
<tr>
<th>#</th>
<th>TASK_ID</th>
<th>TASK_NAME</th>
<th>TASK_COMPLETED</th>
<th>TASK_DUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1Read Cookbook</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>2Eat sandwich</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>3Do some programming</td>
<td></td>
<td>2015-03-01</td>
</tr>
</tbody>
</table>
Documents

- _id: 53f09e2b250c638cc523247
- name: "NetBeans Cookbook"
- author: "David Salter"

Mongo Shell

Path to the mongo executable file:

C:\Program Files\MongoDB 2.6 Standard\bin\mongo.exe
Chapter 7: NetBeans JavaFX

Hello World!

Say 'Hello World'

Hello

What's your name?

Say Hello

New Empty FXML

Steps
1. Choose File Type
2. FXML Name and Location
3. Controller Class
4. Cascading Style Sheet

FXML Controller Class Name and Location

Use Java Controller

Create New

Controller Name: GreetingsController
Location: Source Packages
Package: com.davidsalter.cookbook.welcometofx

Use Existing

Controller Class: 
Browse...

File Location: \WelcomeToF\proj\davidsalter\cookbook\welcometofx\GreetingsController.jrxml
Hi
David

Hello
What's your name?  
Say Hello

What is your name?  

What is your name?  David  
Say 'Hello World'
### Environment Variables

#### User variables for david

<table>
<thead>
<tr>
<th>Variable</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>ANT_HOME</td>
<td>C:\DevTools\apache-ant-1.9.3</td>
</tr>
<tr>
<td>JAVA_HOME</td>
<td>c:\dev\tools\jdk1.8.0</td>
</tr>
<tr>
<td>MAVEN_OPTS</td>
<td>Xmx512m -XX:MaxPermSize=128m</td>
</tr>
<tr>
<td>MVN_HOME</td>
<td>C:\DevTools\apache-maven-3.2.1</td>
</tr>
</tbody>
</table>

#### System variables

<table>
<thead>
<tr>
<th>Variable</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>NUMBER_OF_PROCESSORS</td>
<td>8</td>
</tr>
<tr>
<td>OS</td>
<td>Windows_NT</td>
</tr>
<tr>
<td>Path</td>
<td>c:\app\oracle\product\12.1.0\dbhome ...</td>
</tr>
<tr>
<td>PATH</td>
<td>.COM;.EXE;.BAT;CMD;VBS;VBE;JS;...</td>
</tr>
</tbody>
</table>

### Edit System Variable

**Variable name:** Path

**Variable value:** \bin;C:\Program Files (x86)\Inno Setup 5
WelcomeToFX

Package as

jfx-build-native:

BUILD SUCCESSFUL (total time: 4 seconds)
David Salter

Welcome To JavaFx  NEW

Icons and Splash Image

- WebStart Icon:
- Splash Image:
- Native Package Icon:

(Leave empty for defaults)

[OK] [Cancel]
Chapter 8: NetBeans Mobile Development
Steps
1. Select platform type
2. Platform Folders
3. Detected Platforms

Detected Platforms
Select Platforms to Install:
- C:\DevTools\Java_ME_platform_SDK_3.4

Description
- Javadocs
- Sources

Platform Name: Oracle Java(TM) Platform Micro Edition SDK 3.4
Platform Details:
- Devices: JavaMEPhone1, ClamshellJavaMEPhone1, IMPINGDevice1, IMPINGDevice2, JavaMEPhone2, Qualcomm_Jav ME_Device
- Profiles: MIDP, MIDP-2.0, MIDP-2.1
- Configurations: CLDC-1.0, CLDC-1.1
- Optional APIs:
  - javax.accessibility.*
  - javax.jws.*
  - javax.jfr.*
  - javax.management.*
  - javax.microedition.*
  - javax.security.*
  - javax.sound.*
  - javax.sip.*
  - javax.telephony.*
  - javax.xml.*
  - javax.xml.bind.*
  - javax.xml.soap.*
  - javax.xml.transform.*
  - javax.xml.ws.*
  - javax.xml.xpath.*
  - javax.xml.transform.dom.*
  - javax.xml.transform.stax.*

Java Platform Manager
Use the Javadoc tab to register the API documentation for your JDK in the IDE.
Click Add Platform to register other Java platform versions.

Platforms:
- CLOC
  - Oracle Java(TM) Platform Micro Edition SDK 3.4
- Java SE
  - JDK 1.8 (default)

Platform Name: Oracle Java(TM) Platform Micro Edition SDK 3.4
Platform Folder: C:\DevTools\Java_ME_platform_SDK_3.4
Devices:
- JavaMEPhone1
- ClamshellJavaMEPhone1
- IMPINGDevice1
- IMPINGDevice2
- JavaMEPhone2
- Qualcomm_Jav ME_Device

Tools & Extensions

Add Platform... Remove

Close Help
MyMIDPApp

Project Configuration: Default Configuration

Category:
- General
- Platform
- APIs
- Application Descriptor
- Build
- Libraries & Resources
- Obfuscating
- Creating JAR
- Signing
- Generating Javadoc
- Running
- Deploying

Use Values from "DefaultConfiguration"

Select Platform Type: CLDC

Emulator Platform: Oracle Java(TM) Platform Micro Edition SDK 3.4

Device:
- JavaMEPhone1
- PalmPDAppliance1
- IMPDAppliance2

Device Profile: JavaMEPhone1

Optional Packages:
- Advanced Multimedia Supplements API 1.0
- Contactless Communication API 1.0
- ContentHandler API 1.0
- File Connection and PIM Optional Packages 1.0
- J2ME(TM) Web Services 1.0, Java ME Web Services 1.0
- Java APIs for Bluetooth Wireless Technology 1.1, Object Exchange APIs 1.0
- Java(TM) Binding for OpenGL ES API 1.0
- Location Based APIs 1.0
- Mobile 3D Graphics Optional Package 1.1

Quick Project Run

Current distribution Jar will be used for execution on selected platform. Project will not be built.

Select Platform: Oracle Java(TM) Platform Micro Edition SDK 3.4

Select Device: JavaMEPhone1

OK Cancel Help
<table>
<thead>
<tr>
<th>Name</th>
<th>API</th>
<th>Rev.</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Android SDK Tools</td>
<td>23</td>
<td></td>
<td>Installed</td>
</tr>
<tr>
<td>Android SDK Platform-tools</td>
<td>20</td>
<td></td>
<td>Installed</td>
</tr>
<tr>
<td>Android SDK Build-tools</td>
<td>19.0.3</td>
<td></td>
<td>Installed</td>
</tr>
<tr>
<td>Android 4.4.2 (API 19)</td>
<td>19</td>
<td></td>
<td>Installed</td>
</tr>
<tr>
<td>ARM EABI v7a System Image</td>
<td>19</td>
<td>2</td>
<td>Installed</td>
</tr>
<tr>
<td>Android 4.3 (API 18)</td>
<td>18</td>
<td>2</td>
<td>Installed</td>
</tr>
<tr>
<td>Extras</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Chapter 9: Version Control
Clone Repository

Steps
1. Remote Repository
2. Remote Branches
3. Destination Directory

Remote Repository
Specify Git Repository Location:
Repository URL: https://github.com/choose/cookbook-jarviewer.git
http[s]://host.wz[org]/path/to/repo.git/
User: dobbie (leave blank for anonymous access)
Password: ************
Save Password

Proxy Configuration...

Specify Destination Folder:
Clone into: C:/Users/david/Documents/NetBeansProjects cooker-jarviewer
(Leave empty to specify the destination later)

Clone Repository

Steps
1. Remote Repository
2. Remote Branches
3. Destination Directory

Destination Directory
Specify the Parent Directory and Name for this Clone:
Parent Directory: C:/Users/david/Documents/NetBeansProjects
Clone Name: cookbook-jarviewer

Checkout Branch: master
Remote Name: origin

Scan for NetBeans Projects after Clone

< Back Next > Finish Cancel Help
Checkout

Subversion Repository

Specify the location of Subversion repository.
- Repository URL: https://cookbook-jarviewer-svn.googlecode.com/svn/trunk
  https://hostname/repository_path[@REV]
- User: 
- Password: 
- Save Username and Password
- Client Certificate File: Browse...
- Passphrase:

Proxy Configuration...

Folders to Checkout

Specify the folder(s) to checkout from Subversion repository.
- Repository Folder(s): trunk
- Repository Revision: HEAD
  Search... Browse...

- Skip "trunk" and checkout only its content
- Export a clean directory tree from the repository

Specify the local folder to checkout folders into.
- Local Folder: C:\Users\david\Documents\NetBeansProjects
  Browse...
- Working Copy: C:\Users\david\Documents\NetBeansProjects\trunk
  Prefer Old Subversion 1.6 Format

- Scan for NetBeans Projects after Checkout
David Salter <david@developjava.com> 07/06/14 11:22

**Added file header and author information to the files.**

David Salter <david@developjava.com> 07/06/14 10:18

**Initial commit.**
public static void main(String[] args) {
    LOG.info("Starting application: "+args.length+" arguments.");
    java.awt.EventQueue.invokeLater(new Runnable() {

    });
# This patch file was generated by NetBeans IDE
# It uses platform neutral UTF-8 encoding and \n newlines.
--- Remotely Modified (Based On HEAD)
+++ Locally Modified (Based On LOCAL)
@@ -16,17 +16,20 @@
package com.davidsalter.cookbook.jarviewer;

Create Branch

Branch Name: Bug911

Revision: master

Commit ID: master (0cf4e85b9e080091c4aeadd44d286160385d8467)
Author: David Salter <david@developingjava.com>
Message: Added file header and author information to the files.

- [x] Checkout Created Branch

Create  Cancel  Help
Chapter 10: NetBeans Testing and Profiling
The existing test class will be updated.

<table>
<thead>
<tr>
<th>Method Access Levels</th>
<th>Generated Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Public</td>
<td>Test Initializer</td>
</tr>
<tr>
<td>Protected</td>
<td>Test Finalizer</td>
</tr>
<tr>
<td>Package Private</td>
<td>Test Class Initializer</td>
</tr>
<tr>
<td></td>
<td>Test Class Finalizer</td>
</tr>
<tr>
<td></td>
<td>Default Method Bodies</td>
</tr>
</tbody>
</table>

Generated Comments:
- Javadoc Comments
- Source Code Hints
Chapter 11: Using External Web Services

![Diagram of Web Services with Amazon and Delicious nodes]

Customize GET Saas Service

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>tag</td>
<td>java.lang.String</td>
<td>Java</td>
</tr>
<tr>
<td>url</td>
<td>java.lang.String</td>
<td>&lt;not set&gt;</td>
</tr>
<tr>
<td>hashes</td>
<td>java.lang.String</td>
<td>&lt;not set&gt;</td>
</tr>
<tr>
<td>meta</td>
<td>java.lang.String</td>
<td>&lt;not set&gt;</td>
</tr>
</tbody>
</table>

Output - DeliciousBookMarks (run)

run:
Develop In Java Newsletter #2
BUILD SUCCESSFUL (total time: 1 minute 27 seconds)
By authorizing this link, you'll allow **FlickrPhotos** to:

- **Access** your Flickr account (including private content)
- **Upload, Edit, and Replace** photos and videos in your account
- **Interact** with other members' photos and videos (comment, add notes, favorite)
- **Delete** photos and videos from your account

[OK, I'LL AUTHORIZE IT] [NO THANKS]

```java
private static boolean readOnly = true;
```
Public API access
Use of this key does not require any user action or consent, does not grant access to any account information, and is not used for authorization.
Learn more

Create new Key

Key for browser applications

| API KEY | [black] |
| REFERERS | Any referer allowed |
| ACTIVATION DATE | [black] |
| ACTIVATED BY | [black] |

Edit allowed referers  Regenerate key  Delete

Web Services
- Amazon
- Delicious
- Flickr
- Google
- Accounts Service
- AdSenseForContent Service
- AdSenseForSearch Service
- Geocoding Service
- [geo]
  - geocode

Customize GET Saas Service

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>q</td>
<td>java.lang.String</td>
<td>[trafalgar square, london]</td>
</tr>
<tr>
<td>output</td>
<td>java.lang.String</td>
<td>[json]</td>
</tr>
</tbody>
</table>

OK  Cancel

# To change this template, choose Tools | Templates
# and open the template in the editor.

api_key="my api key"
"short_name": "WC2N 6DN",
"types": [ "postal_code" ]
},
"formatted_address": "Trafalgar Square, London WC2N 6DN, UK",
"geometry": {
"location": {
"lat": 51.508039,
"lng": -0.123069
},
"location_type": "APPROXIMATE",
"viewport": {
"northeast": {
"lat": 51.50938798029149,
"lng": -0.126720019708493
},
"southwest": {
"lat": 51.5066900197088,
"lng": -0.129417980291502
}
}
},
"types": [ "point_of_interest", "establishment" ]
}
,"status": "OK"
}

BUILD SUCCESSFUL (total time: 2 seconds)
java.lang.String email = "cookbookreader@gmail.com";
Test Web Service Method

**Test Web Service Method:**
Enter the values for the parameters and press the 'Submit' button.

<table>
<thead>
<tr>
<th>Type</th>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>java.lang.String</td>
<td>unregisteredUserEmail</td>
<td></td>
</tr>
<tr>
<td>java.lang.String</td>
<td>userID</td>
<td></td>
</tr>
<tr>
<td>java.lang.String</td>
<td>password</td>
<td></td>
</tr>
<tr>
<td>java.lang.String</td>
<td>email</td>
<td></td>
</tr>
<tr>
<td>int</td>
<td>timeout</td>
<td>0</td>
</tr>
<tr>
<td>java.xml.ws.Holder&lt;com.strikeVerifyEmailResult&gt;</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Submit

Results:

Add Web Service

Specify the web service, or REST resources, descriptor file (WSDL or WADL file):

- **Local File:**
- **URL:** http://www.webservicex.net/stockquote.asmx?WSDL
- **Package Name:** <click to override>

Use the Tools->Plugins manager to install the JAX-RPC library for J2EE 1.4 compatibility.

OK  Cancel  Help
Source Packages
- com.davidalter.cookbook.delicious
- org.netbeans.saas
  - RestConnection.java
  - RestResponse.java
- org.netbeans.saas.delicious
  - DeliciousBookmarkingService.java
  - DeliciousBookmarkingServiceAuthenticator.java
  - deliciousbookmarkingserviceauthenticator.properties

Output - Geocoding (run)

```java
java.io.IOException: <html><head><meta http-equiv="content-type" content="text/html; charset=utf-8" /></head><title>Sorry...</title>
  <style>body { font-family: verdana, arial, sans-serif; background-color: #ffffff; color: #000000; user-select: none; }
</style></head>

```
Chapter 12: Extending NetBeans

New Module

Steps
1. Choose Project
2. Name and Location
3. Basic Module Configuration

Name and Location
- Project Name: AddUnitTest
- Project Location: C:\Users\david\Documents\NetBeansProjects\AddUnitTest
- Project Folder: C:\Users\david\Documents\NetBeansProjects\AddUnitTest

Standalone Module
- NetBeans Platform: Development IDE
- Add to Module Suite
- Module Suite:

Basic Module Configuration
- Code Name Basis: com.davidalter.cookbook.codegenerator
  [e.g. "org.yourorg.module.name"]
- Module Display Name: Add Unit Test
- Localizing Bundle: com.davidalter/cookbook/codegenerator/Bundle.properties
- Generate OSGi Bundle

Projects × Files
Add Unit Test
public String getDisplayName() {
    return "JUnit Test...";
}
Add Module Dependency

Filter: (try class or package names; JAR paths; display or code names)
Cluster: All clusters

Show Non-API Modules
Show Excl. Modules
Match Case

Module:
Java Lexer
Java Platform
Java Project Support
Java Refactoring
JavaServer Faces 1.2 Library
JavaServer Faces 2.2 Library
Java Source
Java Source UI

Description:
This module provides Java Source API and its implementation.

For the list of included fixed bugs please check http://wiki.netbeans.org/NetBeans8.0PatchesInfo.

Show Javadoc

OK Cancel Help

Module Dependencies:
Editor Library 2
Java API Wrapper
Java Source
Lookup API
MIME Lookup API
Utilities API

Generate
JUnit Test...
Constructor...
Logger...
toString...
Override Method...
Add Property...
Call Web Service Operation...
Generate REST Client...
@Test
public void testAbc() throws IOException {
    fail("Test not written yet");
}

@MimeRegistration|mimeType = "text/x-java", service = CodeGenerator.Factory.class
public static class Factory implements CodeGenerator.Factory {
    public List<? extends CodeGenerator> create(Lookup context) {
        return Collections.singletonList(new AddUnitTest(context));
    }
}

public String getDisplayName() {
    return "JUnit Test...";
}

public void invoke() {
    try {
        CancellableTask task = new CancellableTask<WorkingCopy>() {

            @Override
            public void cancel() {...2 lines }

            @Override
            public void run(WorkingCopy workingCopy) throws Exception {...29 lines }
        };
        Document doc = textComp.getDocument();
        JavaSource javaSource = JavaSource.forDocument(doc);
        ModificationResult result = javaSource.runModificationTask(task);
        result.commit();
    } catch (Exception e) {
        Exceptions.printStackTrace(e);
    }
}
Welcome to the NetBeans Plugin Portal

Download, comment, and rate plugins provided by community members and third-party companies, or post your own contributions!

Publish Your Own Plugin