



# Web services

Web service support for cab label printers  
How to create a Java service client with **NetBeans**

## Web Service support for cab Label Printers

valid for following printer types:

**A+ -Series**™

**XD4**™

**Mach 4**™

**PX -Print ModuleSeries**™

**and all printing systems based on the cab „X2“ board**

copyright © **cab Produkttechnik GmbH & Co KG**

all rights reserved

No parts of this manual may be copied, rewritten or used for anything else than for original cab printers.

This interdicts the usage of the manual for OEM products unless you have a written permission.

The cab printers command language is owned and copyrighted by

[cab Produkttechnik GmbH & Co KG](#)

cab Produkttechnik GmbH & Co KG

Wilhelm Schickard Str. 14

76131 Karlsruhe / Germany

Tel: +49 - 721-6626-0

Fax: +49 - 721-6626-239

Email: [support@cabgmbh.com](mailto:support@cabgmbh.com)

<http://www.cabgmbh.com>

All registered trademarks or product names are trademarks of their respective companies

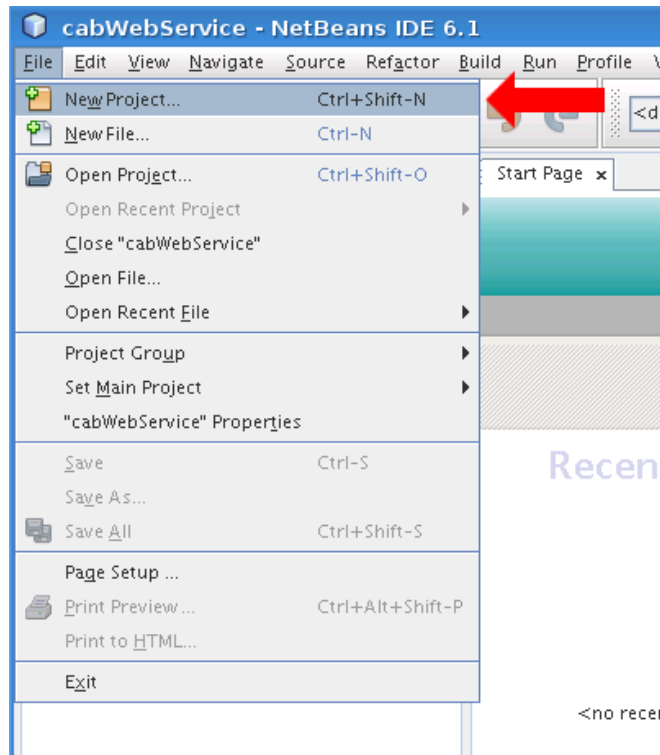
Swiss™ is registered Trademark of Bitstream Inc.

## How to create a Web Service client with Netbeans

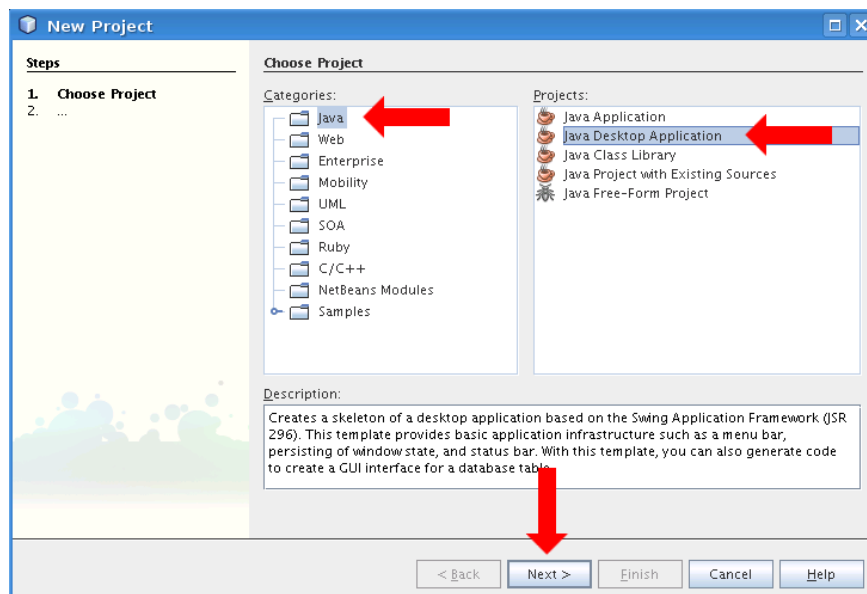


You have to download and install Netbeans (<http://www.netbeans.com>).

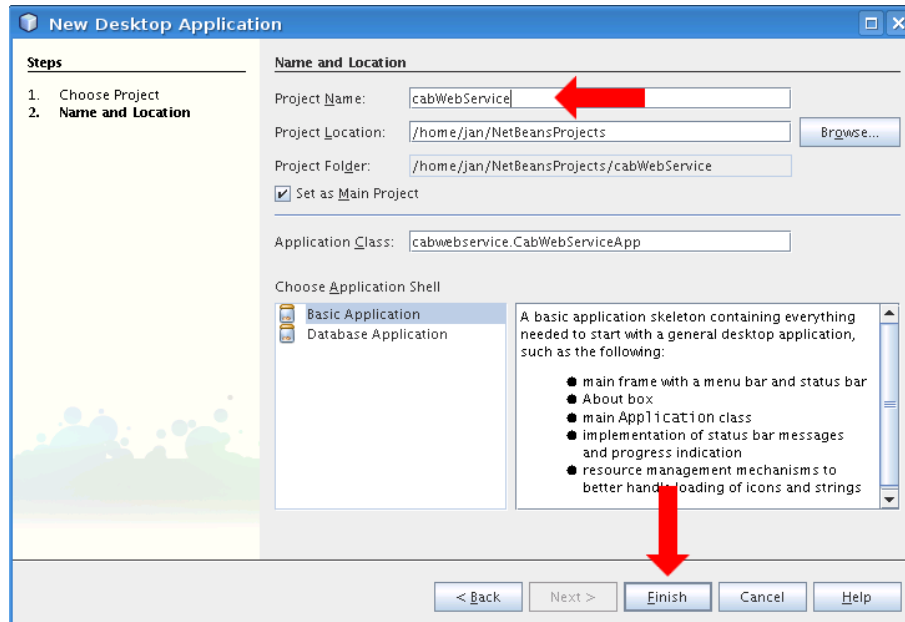
After you start Netbeans you have to create a new project.



In the opened dialog choose a 'Java Desktop Application' and then click on 'Next'.

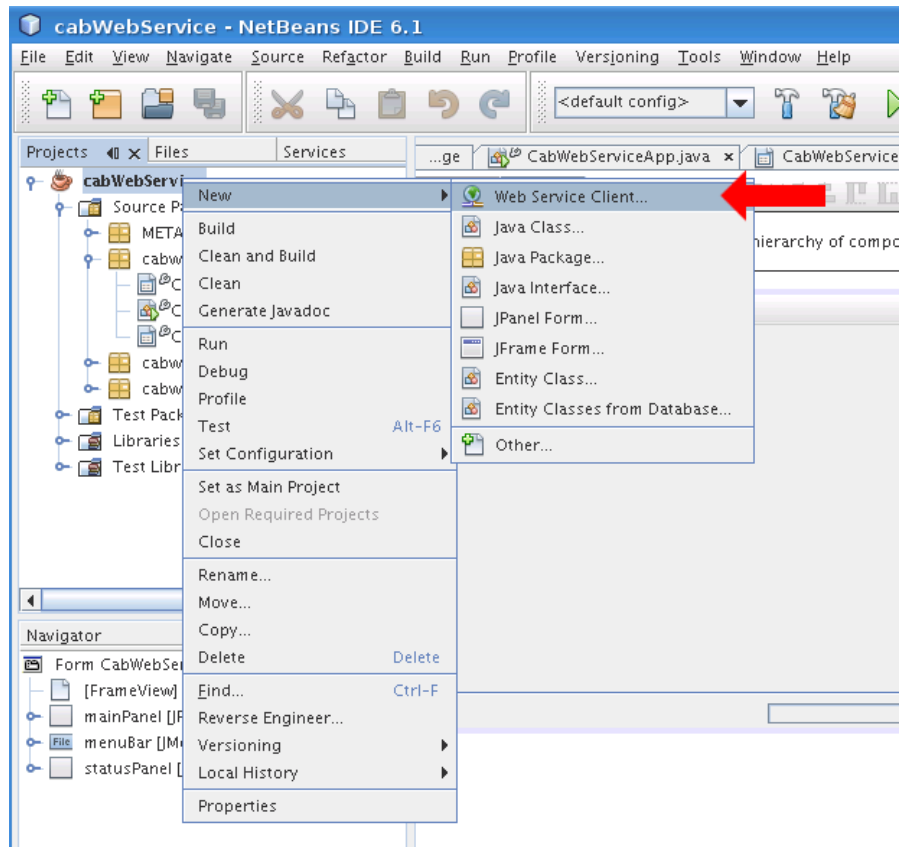


Next give your application a name and click on 'Finish'.

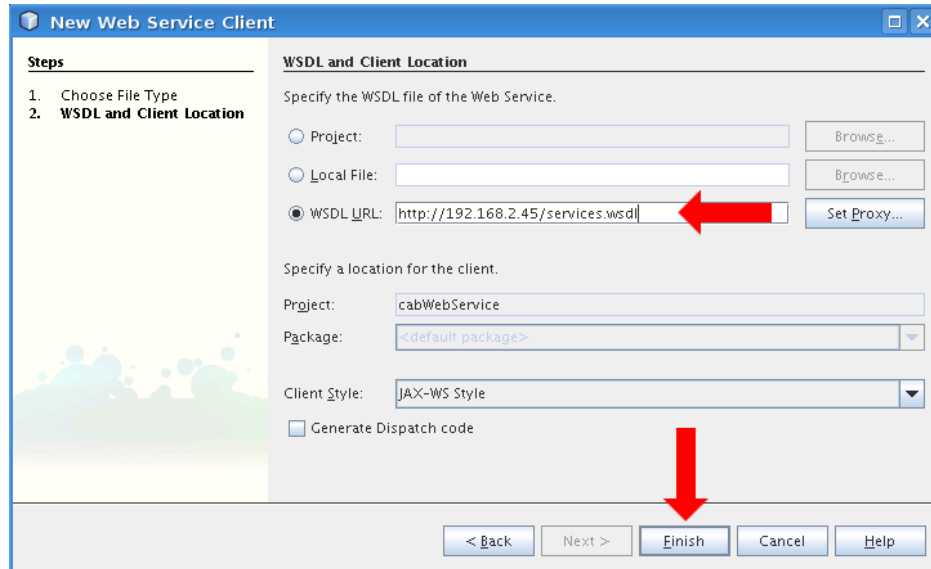


Next step is to create the Web Service Proxy.

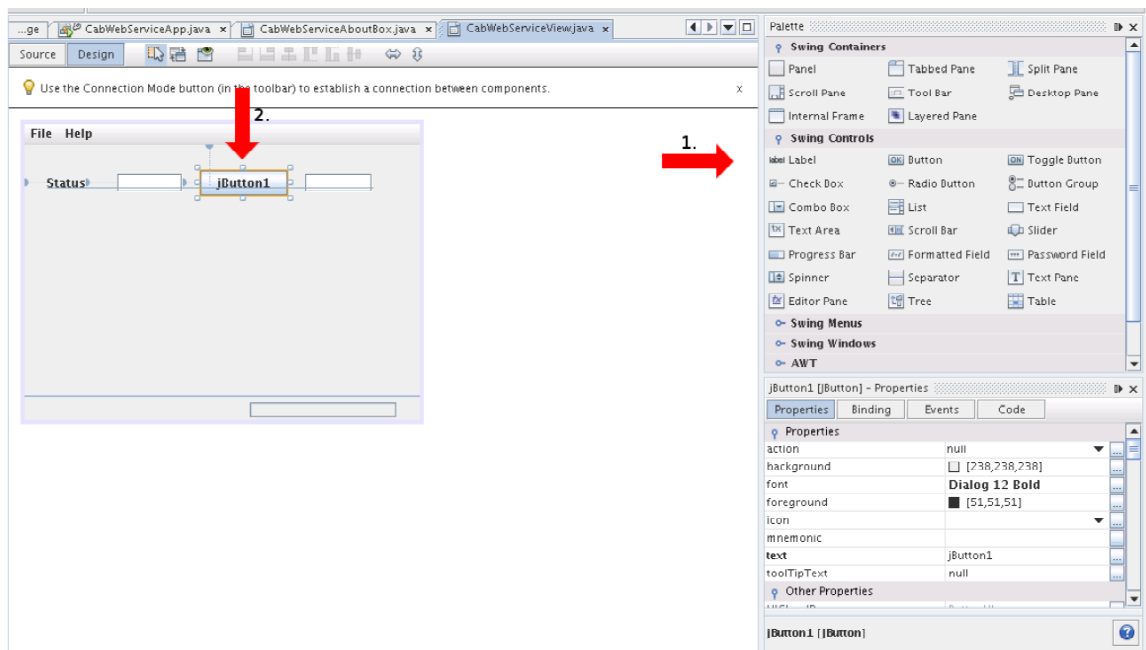
In the project view right click the root element from your project and go to 'New->Web Service Client...'



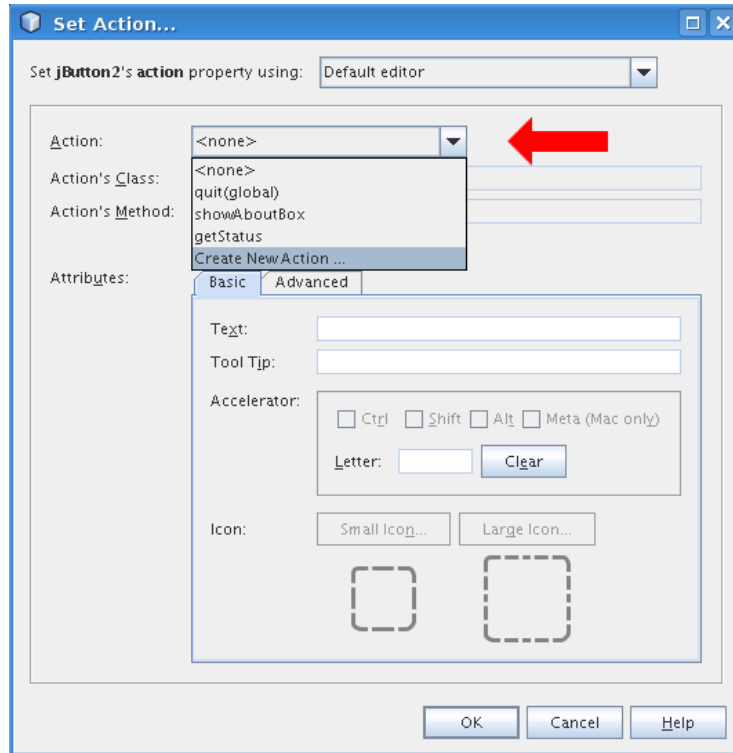
In the opened dialog choose 'WSDL URL' and type in the IP-address or DNS-name from your cab printer and add '/services.wsdl'. This is the location of the WSDL. Close the dialog by pressing the 'Finish' button.



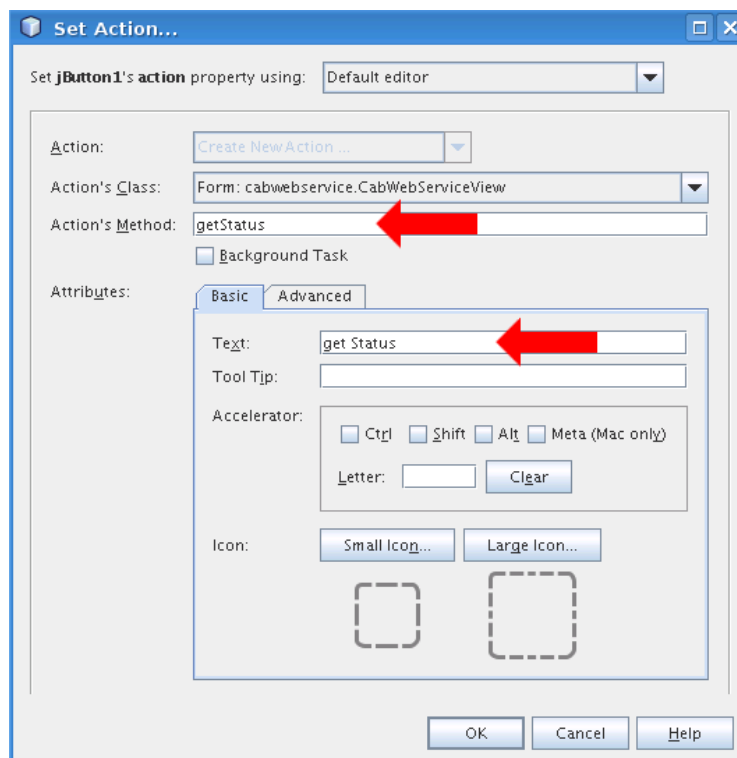
In the next step create your user interface. You can place the interface objects by drag and drop from the right side to the preview. In our case it's a simple status query, with an input text box, a button and an output text box. To add a click action to the button you have to double click on it.



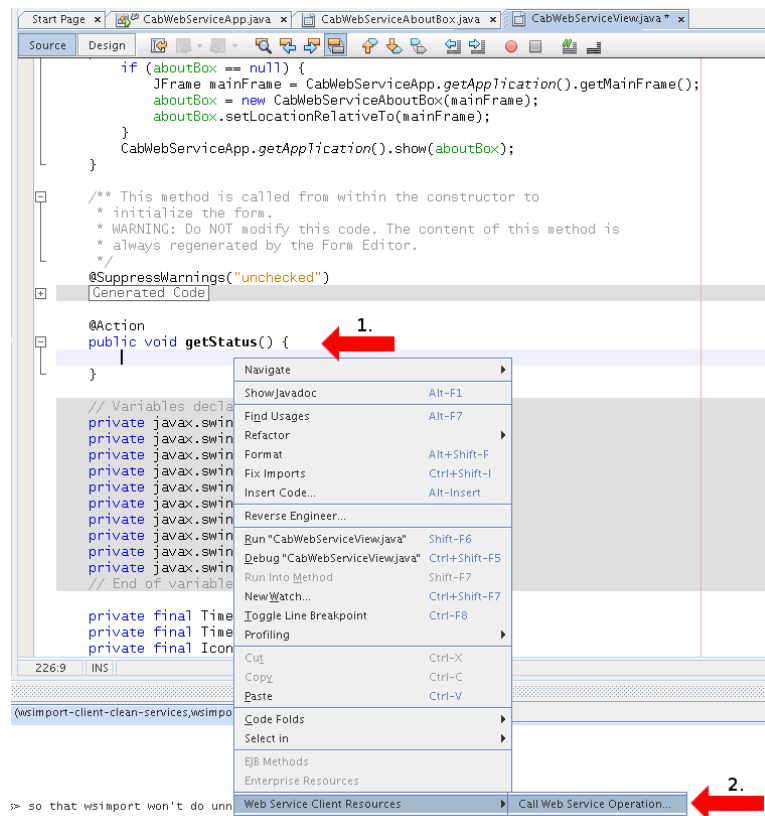
In the recent opened dialog choose in the 'Action' pull down menu the 'Create New Action..' Item.



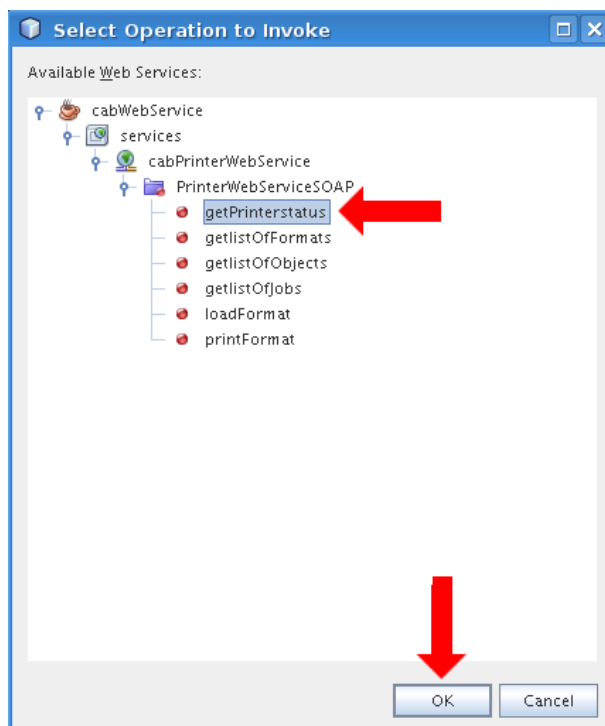
Then you give your new action a method and a button name.



After clicking the 'OK' button you see the source code of your new method. To create a web service query, right click in the method and click on 'Web Service Client Resources->Call Web Service Operation'.



In the opened dialog choose the operation in our case 'getPrinterStatus' and click the 'OK' button.



Now you see the automatic generated source code for calling the cab printer status web service.

```

    @Action
    public void getStatus() {
try { // Call Web Service Operation
de.cab.wsschema.CabPrinterWebService service = new de.cab.wsschema.CabPrinterWebService();
de.cab.wsschema.CabPrinterSOAP port = service.getPrinterWebServiceSOAP();
// TODO initialize WS operation arguments here
java.lang.String parameters = "";
// TODO process result here
java.lang.String result = port.getPrinterstatus(parameters);
System.out.println("Result = "+result);
} catch (Exception ex) {
// TODO handle custom exceptions here
}
}
    
```

Modify the code to adapt it in your created user interface.

You need the following mandatory methods to send a web service query to the cab printer.

- create the service (Arrow 1)
- open the SOAP binding (Arrow 2)
- call the status web service (Arrow 3)

At arrow 4 is the documentation of the method.

The status web service need a String with the query parameter(s) and returns a string with the answer. The two strings originate from the text boxes in the user interface.

```

    @Action
    public void getStatus() {
try {
// initialize Web Service
de.cab.wsschema.CabPrinterWebService service = new de.cab.wsschema.CabPrinterWebService();
de.cab.wsschema.CabPrinterSOAP port = service.getPrinterWebServiceSOAP();

// call Status Web Service
this.tStatusOUT.setText(port.);
} catch (Exception ex) {
// TODO handle custom exceptions here
}
}
    
```

**Method Call Popup:**

- Method: `getPrinterstatus(String parameters)` returns `String`
- Documentation: `Printer Status. Input Values: ESCs, ESCz, ESCi, ESCj, ESCa, ESCj, ESC7, Generic, Label Output Values: Generic: Output a generic Printer Status, Label: Output the loaded Label, Other: see Programming manual`
- Parameters: `parameters`
- Returns: `returns java.lang.String`



You see the final code below.

```

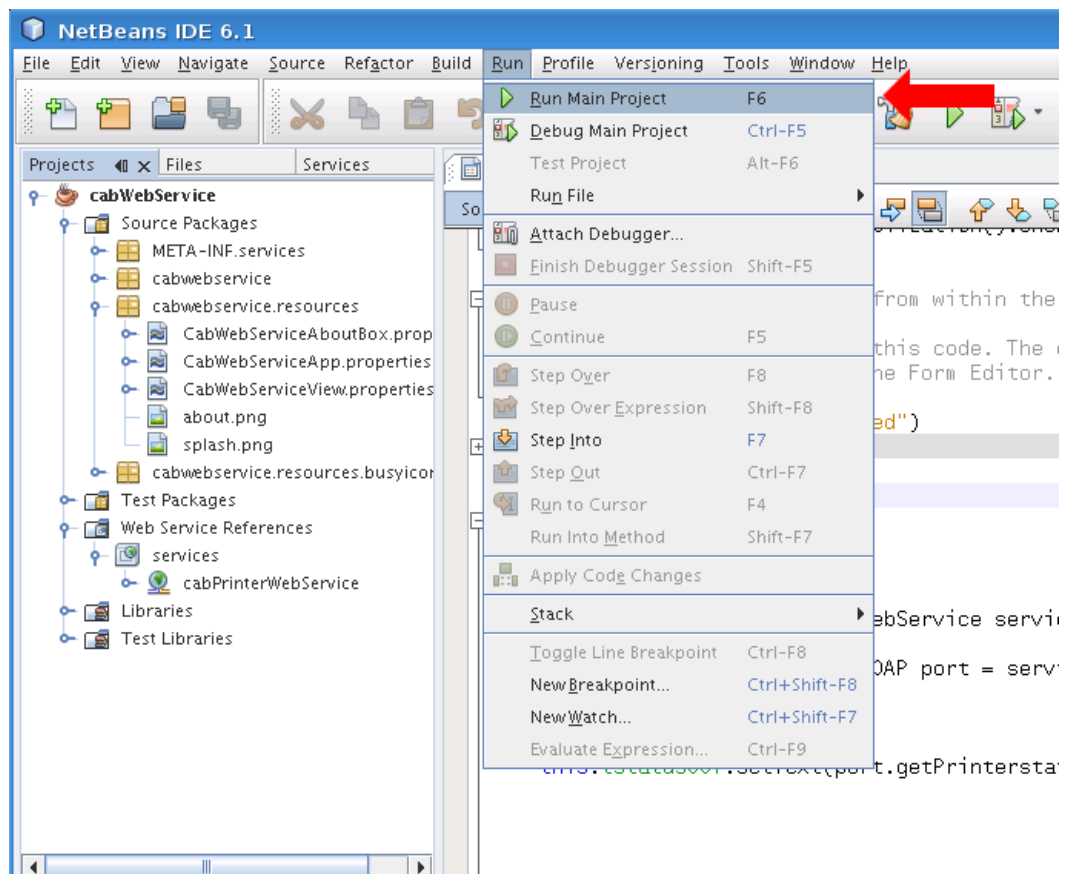
    @Action
    public void getStatus() {
try {
    // initialize Web Service
    de.cab.wsschema.CabPrinterWebService service = new de.cab.wsschema.CabPrinterWebService();
    de.cab.wsschema.CabPrinterSOAP port = service.getPrinterWebServiceSOAP();

    // call Status Web Service
    this.tStatusOUT.setText(port.getPrinterstatus(this.tStatusIN.getText()));

} catch (Exception ex) {
    // TODO handle custom exceptions here
}
}

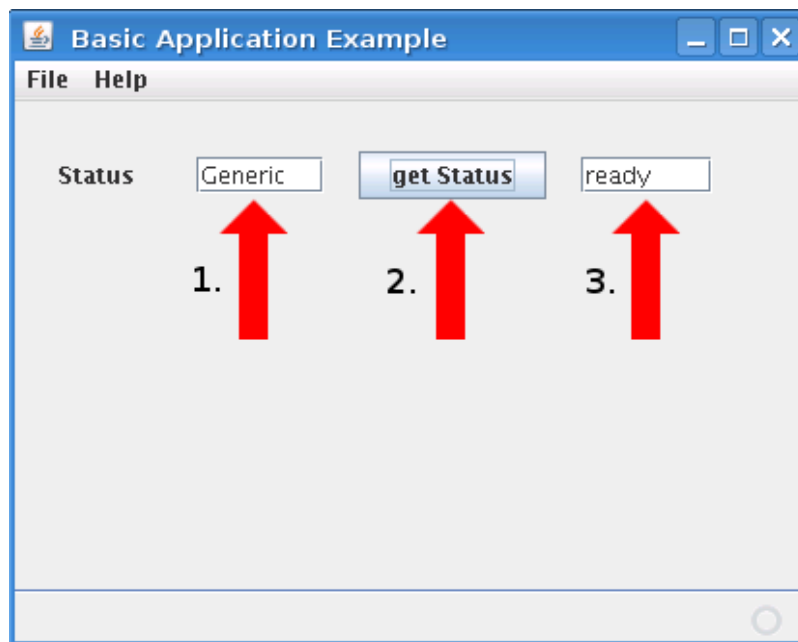
```

You can test your application now. In the main bar on the top of Netbeans go to 'Run->Run Main Project'.



You see now your created user interface.

Enter the query message, like 'Generic' in the input text box. Click the button and wait for the response.



An enhanced sample code, including MD5 authorisation is available in the download area of cab's website