

# Property File Reading and Writing

## 1. Write to a config.properties file using File Output Stream

```
App.java
package com.mkyong.properties;

import java.io.FileOutputStream;
import java.io.IOException;
import java.io.OutputStream;
import java.util.Properties;

public class App {
    public static void main(String[] args) {

        Properties prop = new Properties();
        OutputStream output = null;

        try {

            output = new FileOutputStream("config.properties");

            // set the properties value
            prop.setProperty("database", "localhost");
            prop.setProperty("dbuser", "mkyong");
            prop.setProperty("dbpassword", "password");

            // save properties to project root folder
            prop.store(output, null);

        } catch (IOException io) {
            io.printStackTrace();
        } finally {
            if (output != null) {
                try {
                    output.close();
                } catch (IOException e) {
                    e.printStackTrace();
                }
            }
        }
    }
}
```

### Output:-

```
config.properties
#Fri Jan 17 22:37:45 MYT 2014
dbpassword=password
database=localhost
dbuser=mkyong
```

## 2. Read from config.properties file using file input stream

```
App.java
package com.mkyong.properties;
import java.io.FileInputStream;
import java.io.IOException;
import java.io.InputStream;
import java.util.Properties;

public class App {
    public static void main(String[] args) {

        Properties prop = new Properties();
        InputStream input = null;

        try {

            input = new FileInputStream("config.properties");

            // load a properties file
            prop.load(input);

            // get the property value and print it out
            System.out.println(prop.getProperty("database"));
            System.out.println(prop.getProperty("dbuser"));
            System.out.println(prop.getProperty("dbpassword"));

        } catch (IOException ex) {
            ex.printStackTrace();
        } finally {
            if (input != null) {
                try {
                    input.close();
                } catch (IOException e) {
                    e.printStackTrace();
                }
            }
        }
    }
}
```

Output:-

```
localhost
mkyong
password
Copy
```

## 3. Load a properties file from classpath

```
package com.mkyong.properties;

import java.io.IOException;
```

```

import java.io.InputStream;
import java.util.Properties;

public class App {

    public static void main( String[] args ){

        Properties prop = new Properties();
        InputStream input = null;

        try {

            String filename = "config.properties";
            input = App3.class.getClassLoader().getResourceAsStream(filename);
            if(input==null){
                System.out.println("Sorry, unable to find " + filename);
                return;
            }

            //load a properties file from class path, inside static method
            prop.load(input);

            //get the property value and print it out
            System.out.println(prop.getProperty("database"));
            System.out.println(prop.getProperty("dbuser"));
            System.out.println(prop.getProperty("dbpassword"));

        } catch (IOException ex) {
            ex.printStackTrace();
        } finally{
            if(input!=null){
                try {
                    input.close();
                } catch (IOException e) {
                    e.printStackTrace();
                }
            }
        }
    }
}

```

#### Output:-

```

localhost
mkyong
password

```

#### 4. Load a properties file config.properties from project classpath, and get the keys and values.

```
App.java
package com.mkyong.properties;

import java.io.IOException;
import java.io.InputStream;
import java.util.Enumeration;
import java.util.Properties;

public class App {

    public static void main(String[] args) {
        App app = new App();
        app.printThemAll();
    }

    private void printThemAll() {

        Properties prop = new Properties();
        InputStream input = null;

        try {

            String filename = "config.properties";
            input = getClass().getClassLoader().getResourceAsStream(filename);
            if (input == null) {
                System.out.println("Sorry, unable to find " + filename);
                return;
            }

            prop.load(input);

            Enumeration<?> e = prop.propertyNames();
            while (e.hasMoreElements()) {
                String key = (String) e.nextElement();
                String value = prop.getProperty(key);
                System.out.println("Key : " + key + ", Value : " + value);
            }

        } catch (IOException ex) {
            ex.printStackTrace();
        } finally {
            if (input != null) {
                try {
                    input.close();
                } catch (IOException e) {
                    e.printStackTrace();
                }
            }
        }
    }
}
```

```
}
```

**Output:-**

Key : dbpassword, Value : password

Key : database, Value : localhost

Key : dbuser, Value : mkyong

Copy

References

Java : GetResourceAsStream In Static Method

Java Properties JavaDoc

java properties file