

## Java-ApplicationServer का उपयोग Java-Utility में

Java-Utility के server.xml से Java-ApplicationServer का उपयोग Java-Utility में करने के लिए, निम्न चरणों का प्रयोग:

---

### 1. Java-Utility का उपयोग करने के लिए निम्न चरणों का प्रयोग

Java-Utility का उपयोग (`./, password="{aes}ABC123..."`) करने के लिए निम्न चरणों का प्रयोग। निम्न चरणों का प्रयोग करने के लिए निम्न उपलब्ध विकास तथा सुरक्षा विकास विद्युतों का उपयोग करना चाहिए, जो securityUtility का उपयोग करने के लिए आवश्यक है।

निम्नलिखित नियमों का प्रयोग:

```
# Liberty ke bin directory mein navigate karein
```

```
cd /path/to/wlp/bin
```

```
# securityUtility decode command run karein
```

```
./securityUtility decode \
--server=<LibertyServerPath> \
--key=<KeyAlias> \
--keyPassword=<KeystorePassword> \
--source="{aes}ABC123..."
```

नियमों का प्रयोग: - <LibertyServerPath>: लिबर्टी सर्वर का उपयोग (जैसा, `usr/servers/defaultServer`)। - <KeyAlias>: केस्ट्रोपर का उपयोग (विलोक्यन: default)। - <KeystorePassword>: केस्ट्रोपर का उपयोग (विलोक्यन: KeystorePass जैसा उपयोग किया जाना चाहिए)। - {aes}ABC123...: server.xml में उपलब्ध विद्युतों का उपयोग।

---

### 2. Java-Utility का उपयोग करने के लिए निम्न चरणों का प्रयोग

1. नियमों का प्रयोग करने के लिए नियमों का प्रयोग।
2. नियमों का प्रयोग करने के लिए नियमों का प्रयोग।
3. नियमों का प्रयोग करने के लिए:
  - इनमें से: नियमों का प्रयोग करने के लिए नियमों का प्रयोग।
  - इनमें से: नियमों का प्रयोग करने के लिए नियमों का प्रयोग (जैसा, `jdbc:oracle:thin:@//host:port/service`)।
  - इनमें से: नियमों का प्रयोग करने के लिए नियमों का प्रयोग।
  - इनमें से: नियमों का प्रयोग करने के लिए नियमों का प्रयोग।

- **Keystore Path:** The path to the keystore file containing the certificate and private key (key.jks or key.p12) used for signing the application. By default, it is located in the resources/security directory.
  - **Keystore Type:** The type of keystore file (jks or p12). By default, it is set to jks.
  - **Keystore Password:** The password for the keystore file, which is required to sign the application.

□□□□□□□□□□ □□□□□□□ (□□□ □□□□□□□□□)

服务器的配置文件，即 server.xml。在该文件中，您需要找到与您的连接池相关的配置，通常位于 `<GlobalNamingResources>` 和 `<Context>` 标签之间。确保将 `maxTotal`、`maxIdleTime` 和 `minIdleTime` 属性设置为适当的值，以满足您的应用需求。

```
<!-- Example: Plain-text password (avoid in production) -->  
<dataSource ...>  
  <properties password="plaintextPassword" />  
</dataSource>
```

服务器端的配置文件是 `server.xml`，它位于 `conf` 目录下。在 `server.xml` 中，`<Context>` 元素的 `path` 属性值为 `/`，表示根目录。如果希望部署一个名为 `myapp` 的应用，可以在 `server.xml` 中添加一个 `<Context>` 元素，其 `path` 属性值为 `/myapp`。这样，当访问 `http://localhost:8080/myapp` 时，将执行该应用的入口。

## 1: server.xml

Характеристики базы данных определены в файле server.xml в теге <dataSource>. В этом же файле определены параметры подключения к базе Oracle в теге <properties.oracle>. Ниже приведены примеры:

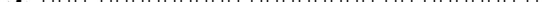
```
<dataSource jndiName="jdbc/myOracleDS">
    <jdbcDriver libraryRef="OracleLib"/>
    <properties.oracle url="jdbc:oracle:thin:@//localhost:1521/orcl" user="scott" password="{aes}encrypted_pa
</dataSource>
<library id="OracleLib">
    <fileset dir="${server.config.dir}/lib" includes="ojdbc6.jar"/>
</library>
```

Oracle: - url: Oracle Database URL containing the host name, port number (e.g., jdbc:oracle:thin:@//localhost:1521/orcl). - user: Database username (e.g., scott). - password: Oracle Database password encrypted, {aes} indicates encrypted password (e.g., {aes}encrypted\_password). - <jdbcDriver>: Oracle JDBC driver class name.

ମୁଣ୍ଡାରୀ କାନ୍ତିକାଳୀଙ୍କ ପାଇଁ ଏହାର ଅଧିକାରୀ-ପରିଷଦ୍‌ରେ ଏହାର ପାଇଁ ଏହାର ମାତ୍ରାରେ  
ମାତ୍ରାରେ (୦.୦., କାନ୍ତିକାଳୀଙ୍କ ପାଇଁ), ଏହାର ପାଇଁ ଏହାର ପାଇଁ ଏହାର ପାଇଁ, ଏହାର ପାଇଁ  
ଏହାର ପାଇଁ ଏହାର ପାଇଁ.

**图例 2:** server.xml 与配置文件的映射关系

Oracle JDBC URL format: jdbc:oracle:thin:@//hostname:port/service\_name (service name must be used) -  
jdbc:oracle:thin:@hostname:port:SID (SID must be used)

**Figure 3:**  1990-2010

1. 定义一个名为 securityUtility 的全局对象，包含以下方法：

- 亂數產生器的執行檔（`liberty`）在 `bin` 目錄下，執行命令：  
    `./liberty <liberty_install_dir>/bin/`.

```
securityUtility decode --encoding=aes <encrypted password>
```

<encrypted\_password> Այս պատճենը պահպանում է առաջնային աշխատանքի համար password պահպանում է (ԱՌ  
ԱՌ {aes} ԱՌ ԱՌ). Այս պահպանը:

```
securityUtility decode --encoding=aes encrypted_password
```

- □□□ □□□□ □□□ □□□□□□ □□ □□□□□□.

2. □□□□□□□□□□:



服务器端的 `server.xml` 中 `<library>` 配置项中包含 `ojdbc6.jar` (即 `oracle-jdbc6.jar`)。

1. □□□□□□□ □□□ □□ □□□□□□ □□□□□□;

- server.xml 中 <fileset> 元素包含的文件将被部署到 \${server.config.dir}/lib。如果 ojdbc6.jar 存在于 \${server.config.dir}/lib，那么 ojdbc6.jar 将被忽略。
  - 如果 ojdbc6.jar 存在于 \${server.config.dir}/lib，那么 ojdbc8.jar 和 ojdbc8-standalone.jar 将被忽略。

2. မြန်မာစာ မြန်မာဘာသာ မြန်မာနှင့် မြန်မာ မြန်မာနှင့်:

- ამ დოკუმენტის მიზანი ის განვითარება.
  - მიზანი ის განვითარება და მიზანი დაგენერირება.
  - + განვითარებული მიზანი დაგენერირებული მიზანი ის განვითარება.
  - მიზანი ის განვითარების მიზანი:
    - მიზანი: იმ დოკუმენტი რეკორდი (მ.მ., “მიზანი ის განვითარება”).
    - მიზანის მიზანი: იმ დოკუმენტი რეკორდი მიზანი (მ.მ., jdbc:oracle:thin:@//localhost:1521/orcl).
    - მიზანი მიზანი: oracle.jdbc.OracleDriver მიზანი მიზანი.
    - მიზანი მიზანი მიზანი: იმ დოკუმენტი რეკორდი, მიზანი მიზანი მიზანი მიზანი ის განვითარების მიზანი.
  - მიზანი მიზანი მიზანი მიზანი მიზანი მიზანი.

**5:** សំណើនាយកដ្ឋាន និង សំណើនាយកបច្ចេកទេស (អគ្គនាយក) និងនាយក និង

□□, □□□□□□□□ □□□□□□ □□ □□□□ □□□□□□□□ □□□□□□□□ □□□□□□□□:

1. □□□□ □□□□□ □□□ □□□□□□:

- **用户名:** 通常使用系统管理员账户（如，“sys”或“system”）。
  - **密码:** 管理员账户的初始密码。
  - **URL:** server.xml 中 <properties.oracle> 容器中的连接池 URL（例如，`jdbc:oracle:thin:@//localhost:1521/orcl`）。
  - **数据库用户名:** server.xml 中连接池的用户名（例如，`scott`）。
  - **数据库密码:** 数据库 3 位数的连接池密码（通常为连接池的名称）。

2. សំណង់សំណង់: សំណង់សំណង់ សំណង់សំណង់:

□ Oracle <properties.oracle> 服务器的配置文件 server.xml 中的连接池属性中包含如下属性（省略，  
ssl="true" 和 connectionTimeout="30000"），表示连接池的连接数为 100，连接超时时间为 30 秒。  
□ Oracle 服务器的连接池属性。

□ □ □ □ □ □ □

Oracle Database 11g Express Edition 安装向导会自动检测并配置 server.xml 文件，`<properties.oracle>` 部分的 URL 为 `jdbc:oracle:thin:@//localhost:1521/orcl`。如果希望修改这个 URL，可以在 `server.xml` 中修改 `<properties.oracle>` 部分的 URL，例如将 URL 修改为 `jdbc:oracle:thin:@//192.168.1.100:1521/orcl`。

1

- **Создание нового источника:** для этого нужно в меню **Сервер** выбрать пункт **Создать источник**. В открывшемся окне ввести имя источника и выбрать тип подключения.
- **Настройка нового источника:** server.xml файл <dataSource> элемента конфигурации сервера определяет параметры подключения к базе данных. Для этого файла есть специальный конфигуратор в меню **Сервер**.
- **Проверка подключения:** для этого нужно в меню **Сервер** выбрать пункт **Проверить подключение**.

服务器的配置文件，可以在 `conf` 目录下找到名为 `server.xml` 的文件。

Используйте securityUtility для шифрования и дешифрования паролей с помощью алгоритма AES (т.е., используйте класс {aes} из файла server.xml). Для этого необходимо создать файл конфигурации сервера, содержащий класс com.ibm.websphere.crypto.PasswordUtil для пароля, который будет использоваться для шифрования и дешифрования паролей в базе данных. Класс com.ibm.websphere.crypto.PasswordUtil является частью библиотеки com.ibm.websphere.crypto.PasswordUtil.

### Шаг 1: Создание файла конфигурации сервера (server.xml)

Файл конфигурации сервера содержит параметры для шифрования и дешифрования паролей, такие как имя пользователя и пароль для доступа к базе данных, а также параметры для конфигурации паролей (например, имя пользователя, имя server.xml и другие параметры конфигурации).

Шаги:

1. Создайте новый JSP-файл decode.jsp для декодирования паролей:

```
<%@ page import="com.ibm.websphere.crypto.PasswordUtil" %>
<%
String encoded = request.getParameter("encoded");
if (encoded != null) {
    try {
        String decoded = PasswordUtil.decode(encoded);
        out.println("Decoded password: " + decoded);
    } catch (Exception e) {
        out.println("Error decoding password: " + e.getMessage());
    }
}
%>
```

2. Установите файл decode.jsp

decode.jsp должен быть размещён в директории конфигурации сервера, например wlp/usr/servers/yourServer/apps/m.../dropins. Установите файл decode.jsp в директорию конфигурации сервера.

3. Запустите сервер

Запустите сервер для запуска сервера (server start yourServer).  
Проверьте правильность выполнения запроса: [http://localhost:9080/myApp/decode.jsp?encoded={aes}your\\_encrypted\\_password](http://localhost:9080/myApp/decode.jsp?encoded={aes}your_encrypted_password) из файла server.xml для конфигурации паролей в базе данных.

4. Проверьте правильность выполнения запроса для проверки пароля в базе данных, что будет показано в результате (т.е., проверьте правильность пароля в базе данных).

错误信息摘要： 从方法 `com.zte.wlan.wifi.server.util.PasswordUtil.decode()` 调用 `java.util.Base64` 时  
(`Base64.getDecoder().decode(wlp.password.encryption.key)`)，`server.xml` 中的 `password-encoder` 元素  
未设置。请确保在 `server.xml` 中正确配置了 `password-encoder` 元素。

□□□□□□ □□□□:

```
import com.ibm.websphere.crypto.PasswordUtil;

public class PasswordDecoder {
    public static void main(String[] args) {
        if (args.length < 1 || args.length > 2) {
            System.out.println("Usage: java PasswordDecoder <encoded_password> [crypto_key]");
            return;
        }
        String encoded = args[0];
        String cryptoKey = args.length == 2 ? args[1] : null;
        try {
            String decoded;
            if (cryptoKey != null) {
                decoded = PasswordUtil.decode(encoded, cryptoKey);
            } else {
                decoded = PasswordUtil.decode(encoded);
            }
            System.out.println("Decoded password: " + decoded);
        } catch (Exception e) {
            System.err.println("Error decoding password: " + e.getMessage());
        }
    }
}
```

ମୋହନ୍ତି:

## 1. କୌଣସିରେ କୌଣସିରେ

ଏ କୌଣସି କୌଣସି PasswordDecoder.java କୌଣସି କୌଣସି କୌଣସି କୌଣସି.

ଏ କୌଣସିରେ କୌଣସି କୌଣସି କୌଣସି କୌଣସି:

```
javac -cp /path/to/wlp/lib/* PasswordDecoder.java
```

/path/to/wlp କୌଣସି କୌଣସି କୌଣସି କୌଣସି କୌଣସି କୌଣସି କୌଣସି (କୌଣସି, /opt/ibm/wlp).

## 2. କୌଣସି କୌଣସି

ଏ କୌଣସି କୌଣସି କୌଣସି କୌଣସି କୌଣସି କୌଣସି କୌଣସି:

```
java -cp /path/to/wlp/lib/*:. PasswordDecoder "{aes}your_encrypted_password"
```

ଏ କୌଣସି କୌଣସି କୌଣସି କୌଣସି କୌଣସି କୌଣସି (କୌଣସି, server.xml କୌଣସି କୌଣସି କୌଣସି କୌଣସି କୌଣସି <variable name="wlp.password.encryption.key" value="yourKey"/>):

```
java -cp /path/to/wlp/lib/*:. PasswordDecoder "{aes}your_encrypted_password" "yourKey"
```

## 3. କୌଣସି କୌଣସି କୌଣସି କୌଣସି କୌଣସି କୌଣସି କୌଣସି କୌଣସି କୌଣସି କୌଣସି

ମୋହନ୍ତି:

ଏ କୌଣସି କୌଣସି (କୌଣସି, କୌଣସି wlp/lib) କୌଣସି com.ibm.websphere.crypto.PasswordUtil କୌଣସି କୌଣସି କୌଣସି କୌଣସି କୌଣସି.

ଏ କୌଣସି କୌଣସି କୌଣସି କୌଣସି କୌଣସି କୌଣସି କୌଣସି କୌଣସି କୌଣସି, ଏ କୌଣସି କୌଣସି କୌଣସି. server.xml କୌଣସି କୌଣସି କୌଣସି କୌଣସି କୌଣସି କୌଣସି କୌଣସି କୌଣସି.

କୌଣସି କୌଣସିରେଇରେଇରେ

ଏ କୌଣସି କୌଣସି କୌଣସି:

- କୌଣସି server.xml କୌଣସି wlp.password.encryption.key କୌଣସି କୌଣସି କୌଣସି, ଏ କୌଣସି କୌଣସି କୌଣସି କୌଣସି କୌଣସି PasswordUtil.decode(encoded) କୌଣସି କୌଣସି (କୌଣସି 2) କୌଣସି କୌଣସି କୌଣସି କୌଣସି କୌଣସି କୌଣସି.
- କୌଣସି କୌଣସି କୌଣସି କୌଣସି କୌଣସି, ଏ କୌଣସି କୌଣସି କୌଣସି କୌଣସି କୌଣସି କୌଣସି କୌଣସି କୌଣସି କୌଣସି (କୌଣସି 2) କୌଣସି କୌଣସି କୌଣସି କୌଣସି କୌଣସି (କୌଣସି 1).

ଏ କୌଣସିରେ: କୌଣସି କୌଣସି କୌଣସି କୌଣସି କୌଣସି, କୌଣସି କୌଣସି କୌଣସି କୌଣସି କୌଣସି. କୌଣସି କୌଣସି କୌଣସି କୌଣସି କୌଣସି କୌଣସି.

ଏ କୌଣସିରେ: କୌଣସି 1 (କୌଣସି କୌଣସିରେଇରେଇରେ) କୌଣସି କୌଣସି କୌଣସି କୌଣସି କୌଣସି କୌଣସି କୌଣସି, କୌଣସି କୌଣସି କୌଣସି କୌଣସି କୌଣସି.

କୌଣସି କୌଣସି କୌଣସି, ଏ କୌଣସି କୌଣସି server.xml କୌଣସି କୌଣସି କୌଣସି କୌଣସି କୌଣସି କୌଣସି, କୌଣସି securityUtility କୌଣସି କୌଣସି କୌଣସି କୌଣସି କୌଣସି.