

MỤC TIÊU:

Kết thúc bài thực hành này bạn có khả năng

- Sử dụng BeanUtils để đọc tham số
- Gửi email
- Sử dụng Cookie
- Sử dụng Session

PHẦN I: SỬ DỤNG BEANUTILS VÀ GỬI EMAIL

BÀI 1: SỬ DỤNG BEANUTILS ĐỂ ĐỌC THAM SỐ

Hãy viết servlet sử dụng BeanUtils để đọc dữ liệu của form có mã như sau:

```
<c:url value="/save" var="url"/>
<form action="${url}" method="post">
  <input name="fullname"> <br>
  <input name="birthday"> <br>

  <input type="radio" name="gender" value="true"> Male
  <input type="radio" name="gender" value="false"> Female <br>

  <input type="checkbox" name="hobbies" value="R"> Reading
  <input type="checkbox" name="hobbies" value="Traveling"> Traveling
  <input type="checkbox" name="hobbies" value="M"> Music <br>

  <select name="country">
    <option value="VN">Việt Nam</option>
    <option value="US" selected>United States</option>
  </select> <br>

  <input name="salary"> <br>

  <hr>
  <button>Submit</button>
</form>
```

Tạo lớp java bean Staff như sau

```
public class Staff {  
    String fullname;  
    Date birthday;  
    boolean gender;  
    String[] hobbies;  
    String country;  
    double salary;  
    getters/setters  
}
```

Tạo Servlet @WebServlet("/save") doGet() chuyển sang jsp để hiển thị form, doPost() để đọc dữ liệu form code như sau:

```
Staff bean = new Staff(); // dùng để chứa dữ liệu form  
try {  
    // Khai báo định dạng ngày  
    DateTimeConverter dtc = new DateConverter(new Date());  
    dtc.setPattern("MM/dd/yyyy");  
    ConvertUtils.register(dtc, Date.class);  
    // đọc và chuyển đổi tham số vào bean  
    BeanUtils.populate(bean, req.getParameterMap());  
    System.out.println(bean.getFullname());  
} catch (Exception e) {  
    e.printStackTrace();  
}
```

Kiểm tra giá trị các tham số đọc được qua console.

Chú ý: khai báo thư viện phụ thuộc trong file pom.xml trước khi viết code

```
<dependency>  
    <groupId>commons-beanutils</groupId>  
    <artifactId>commons-beanutils</artifactId>  
    <version>1.9.3</version>  
</dependency>
```

BÀI 2: GỬI EMAIL

Xây dựng thư viện tiện ích chức năng gửi email với các tham số đơn giản là

- From: email người gửi
- To: email người nhận

- Subject: Tiêu đề mail
- Body: Nội dung mail

Tạo lớp tiện ích

```
public class Mailer {  
    public static void send(String from, String to, String subject, String  
body) {  
        Code send mail đặt ở đây  
    }  
}
```

Viết mã cho hàm tiện ích send

```
// Thông số kết nối GMail  
Properties props = new Properties();  
props.setProperty("mail.smtp.auth", "true");  
props.setProperty("mail.smtp.starttls.enable", "true");  
props.setProperty("mail.smtp.host", "smtp.gmail.com");  
props.setProperty("mail.smtp.port", "587");  
// Đăng nhập GMail  
Session session = Session.getInstance(props, new Authenticator() {  
    @Override  
    protected PasswordAuthentication getPasswordAuthentication() {  
        String username = "user@gmail.com";  
        String password = "*****";  
        return new PasswordAuthentication(username, password);  
    }  
});  
try {  
    // Tạo mail  
    MimeMessage mail = new MimeMessage(session);  
    mail.setFrom(new InternetAddress(from));  
    mail.setRecipients(RecipientType.TO, to);  
    mail.setSubject(subject, "utf-8");  
    mail.setText(body, "utf-8", "html");  
    mail.setReplyTo(mail.getFrom());  
    // Gửi mail  
    Transport.send(mail);  
} catch (Exception e) {
```

```
e.printStackTrace();  
}
```

Chạy thử bằng cách gọi Mailer.send() bất kỳ đâu.

```
String from = "from@gmail.com";  
String to = "to@gmail.com";  
String subject = "i love you";  
String body = "Have you forgot me?";  
Mailer.send(from, to, subject, body);
```

Hãy tạo form nhập from, to, subject, body. Sau đó viết Servlet nhận thông tin từ form và gọi Mailer.send() để gửi email từ trang web.

Chú ý: bạn cần khai báo thư viện phụ thuộc vào pom.xml trước khi viết code

```
<dependency>  
  <groupId>javax.mail</groupId>  
  <artifactId>mail</artifactId>  
  <version>1.4.7</version>  
</dependency>
```

PHẦN II: COOKIE VÀ SESSION

BÀI 3: SỬ DỤNG COOKIE VÀ SESSION

Xây dựng trang web đăng nhập cho phép ghi nhớ tài khoản bằng google.

Tạo form đăng nhập

```
<form action="/login" method="post">  
  <input name="username" value="${username}"> <br>  
  <input name="password" value="${password}"> <br>  
  <input type="checkbox" name="remember-me"> Remember me?  
  <hr>  
  <button>Login</button>  
</form>
```

Tạo servlet @WebServlet("/login"):

- doGet(): đọc cookie user gửi từ trình duyệt. Nếu có thì giải mã Base64 và chuyển sang jsp để hiển thị lên form để người dùng ko cần phải nhập lại nếu lần đăng nhập trước đã ghi nhớ.

```
Cookie[] cookies = req.getCookies();
if(cookies != null) {
    for(Cookie cookie: cookies) {
        if(cookie.getName().equals("user")) {
            String encoded = cookie.getValue();
            byte[] bytes = Base64.decodeBase64(encoded);
            String[] userInfo = new String(bytes).split(",");
            req.setAttribute("username", userInfo[0]);
            req.setAttribute("password", userInfo[1]);
        }
    }
}
```

- doPost(): tạo và gửi cookie về trình duyệt để lưu lại nếu đăng nhập đúng và có tích vào Remember me.

```
String username = req.getParameter("username");
String password = req.getParameter("password");
String remember = req.getParameter("remember-me");
if(username.equalsIgnoreCase("FPT") && password.equals("poly")) {
    req.setAttribute("message", "Login successfully!");
    req.getSession().setAttribute("username", username); // session

    if(remember != null) {
        byte[] bytes = (username+","+password).getBytes();
        String userInfo = Base64.encodeBase64String(bytes);
        Cookie cookie = new Cookie("user", userInfo);
        cookie.setMaxAge(30*24*60*60); // hiệu lực 30 ngày
        cookie.setPath("/"); // hiệu lực toàn ứng dụng
        // Gửi về trình duyệt
        resp.addCookie(cookie); // cookie
    }
} else {
    req.setAttribute("message", "Invalid login info!");
}
```

BÀI 4: GIẢNG VIÊN CHO THÊM