 Hãy nói theo cách của bạn	TẬP ĐOÀN VIỄN THÔNG QUÂN ĐỘI			Mã hiệu: TC.00.CNTT.xx
	THAM KHẢO HƯỚNG DẪN TIÊU CHUẨN AN TOÀN THÔNG TIN ỨNG DỤNG WEB			Ngày có hiệu lực:
				Ngày hết hiệu lực:
				Lần ban hành: 01
				Trang: 1/21
	ĐV soạn thảo	TT.ANM	ĐV kiểm tra	PCNTT - TĐ
	Phạm Vi	Toàn Tập đoàn.		

HƯỚNG DẪN LẬP TRÌNH AN TOÀN TRONG PHÁT TRIỂN ỨNG DỤNG WEB TRÊN FRAMEWORK STRUTS

1. Quản lý xác thực

➤ Thông tin định danh

1.1. Tên đăng nhập phải là duy nhất, chỉ nên chứa tập các ký tự là chữ cái, chữ số, dấu gạch dưới.

```
public static boolean isExistUsername(String username) {
    SessionFactory sessionFactory = HibernateUtil.getSessionFactory();
    Session session = sessionFactory.openSession();
    Query createQuery = session.createQuery("from Customer where name = :username");
    createQuery.setParameter("username", username);
    List list = createQuery.list();
    session.close();
    return list.size() > 0;
}


public static boolean isValidUsername(String username) {
    if (username == null || username.equals("")) {
        return false;
    }
    return username.matches("^[a-zA-Z0-9_]{6,30}$");
}
```

1.2. Thiết lập chính sách mật khẩu mạnh:

- Mật khẩu có độ dài tối thiểu là 8 ký tự.
- Chứa chữ cái, chữ số và ký tự đặc biệt.
- Thiết lập blacklist các mật khẩu yếu (VD: 123456A@, 123456a@,...).

```
public static boolean isStrongPassword(String password) {
    if (password == null) {
        return false;
    }
    return (password.matches("^(?=.*[a-zA-Z])(?=.*[0-9])(?=.*[!@#$%^&*~`{|}~'\s])?.{8,}$") &&
    !isBlacklistPassword(password));
}

private static boolean isBlacklistPassword(String password) {
    BufferedReader reader = null;
    try {
        reader = new BufferedReader(new
        InputStreamReader(AuthenValidation.class.getResourceAsStream("/password_blacklist.txt")));
        String line = null;
        while ((line = reader.readLine()) != null) {
            if (password.equals(line)) {
                return true;
            }
        }
    } catch (Exception e) {
        throw new RuntimeException(e);
    }
}
```

 Hãy nói theo cách của bạn	TẬP ĐOÀN VIỄN THÔNG QUÂN ĐỘI			Mã hiệu: TC.00.CNTT.xx
	THAM KHẢO HƯỚNG DẪN TIÊU CHUẨN AN TOÀN THÔNG TIN ỨNG DỤNG WEB			Ngày có hiệu lực:
				Ngày hết hiệu lực:
				Lần ban hành: 01
				Trang: 2/21
	ĐV soạn thảo	TT.ANM	ĐV kiểm tra	PCNTT - TĐ
	Phạm Vi	Toàn Tập đoàn.		

```

    } finally {
        if (reader != null) {
            try {
                reader.close();
            } catch (IOException ex) {
            }
        }
    }
    return false;
}

```

1.3. Thiết lập thời gian hết hiệu lực cho mật khẩu tối đa 90 ngày, mật khẩu mới không được trùng với mật khẩu hiện tại.

```

public static boolean isExpirePassword(long customerId) {
    SessionFactory sessionFactory = HibernateUtil.getSessionFactory();
    Session session = sessionFactory.openSession();
    Query createQuery = session.createQuery("select lastPasswordChange
from Customer where customerId = :customerId");
    createQuery.setParameter("customerId", customerId);
    Date uniqueResult = (Date) createQuery.uniqueResult();
    session.close();
    Date today = Calendar.getInstance().getTime();
    long diffInMillis = today.getTime() - uniqueResult.getTime();
    long diffInDays = TimeUnit.DAYS.convert(diffInMillis,
TimeUnit.MILLISECONDS);
    return diffInDays > 90;
}

```

1.4. Đối với chức năng reset/ quên mật khẩu:


- Đường dẫn reset/quên mật khẩu được gửi qua email phải bị mất hiệu lực sau lần truy cập đầu tiên hoặc sau 8 giờ nếu không được truy cập.
- Nếu chức năng reset/quên mật khẩu thực hiện gửi mật khẩu qua email thì mật khẩu phải được sinh ngẫu nhiên và phải tuân theo chính sách mật khẩu mạnh tại mục 2.

```

public static boolean isExpirePasswordChange(long customerId) {
    SessionFactory sessionFactory = HibernateUtil.getSessionFactory();
    Session session = sessionFactory.openSession();
    // expiredPasswordChange = today.getTime() + 8*60*60*1000 tại thời
    // điểm người dùng sử dụng chức năng reset/ quên mật khẩu.
    Query createQuery = session.createQuery("select expiredPasswordChange
from Customer where customerId = :customerId");
    createQuery.setParameter("customerId", customerId);
    Date uniqueResult = (Date) createQuery.uniqueResult();
    session.close();
    Date today = Calendar.getInstance().getTime();
    long diff = today.getTime() - uniqueResult.getTime();
    return diff > 0;
}

public static boolean isValidToken(long customerId, String token) {
    SessionFactory sessionFactory = HibernateUtil.getSessionFactory();
    Session session = sessionFactory.openSession();
    // tokenPasswordChange được sinh ngẫu nhiên, độ dài tối thiểu là 128

```

 Hãy nói theo cách của bạn	TẬP ĐOÀN VIỄN THÔNG QUÂN ĐỘI			Mã hiệu: TC.00.CNTT.xx
	THAM KHẢO HƯỚNG DẪN TIÊU CHUẨN AN TOÀN THÔNG TIN ỨNG DỤNG WEB			Ngày có hiệu lực:
				Ngày hết hiệu lực:
				Lần ban hành: 01
				Trang: 3/21
	ĐV soạn thảo	TT.ANM	ĐV kiểm tra	PCNTT - TĐ
	Phạm Vi	Toàn Tập đoàn.		

```

bit tại thời điểm người dùng sử dụng chức năng reset/ quên mật khẩu
    Query createQuery = session.createQuery("select tokenPasswordChange
from Customer where customerId = :customerId");
    createQuery.setParameter("customerId", customerId);
    String uniqueResult = (String) createQuery.uniqueResult();
    boolean isValidate = token.equals(uniqueResult);
    session.beginTransaction();
    if (isValidate || isExpirePasswordChange(customerId)) {
        Query query = session.createQuery("update Customer set
tokenPasswordChange = NULL where customerId = :customerId");
        query.setParameter("customerId", customerId).executeUpdate();
    }
    session.getTransaction().commit();
    session.close();
    return isValidate;
}

public static String generateSecurePassword() {
    String randomString = RandomStringUtils.randomAlphabetic(4) +
RandomStringUtils.random(4, "~!@#$%^&*()") +
RandomStringUtils.randomNumeric(4);
    String securePassword = RandomStringUtils.random(12, randomString);
    return securePassword;
}

```

- 1.5. Chỉ lưu dạng mã hash của mật khẩu trong DB (khuyến nghị thuật toán hash là SHA-256), thêm chuỗi salt ngẫu nhiên vào mật khẩu trước khi thực hiện hash.


```

public static String hashPassword(String password) throws
NoSuchAlgorithmException {
    byte[] salt = new byte[20];
    new SecureRandom().nextBytes(salt);
    MessageDigest digest = MessageDigest.getInstance("SHA-256");
    String input = new String(salt) + password;
    byte[] hash = digest.digest(input.getBytes());
    return Base64.encodeBase64String(hash);
}

```

➤ Xử lý xác thực

- 1.6. Trả về thông báo chung cho trường hợp người dùng đăng ký thông tin định danh (username, email,...) đã tồn tại tại chức năng đăng ký, hoặc gửi sai thông tin định danh tại các chức năng đăng nhập, reset/quên mật khẩu, đổi địa chỉ email,...
- 1.7. Bật cơ chế bảo vệ bằng Captcha hoặc các hình thức tương đương khi đăng nhập sai quá 5 lần liên tiếp. Cần triển khai cơ chế này tại các chức năng quan trọng khác của ứng dụng.
- Sử dụng Captcha an toàn theo Chỉ thị sử dụng Captcha an toàn Tập đoàn đã ban hành.
 - Thực hiện kiểm tra tính hợp lệ của Captcha trước khi thực hiện chức năng được request.

 Hãy nói theo cách của bạn	TẬP ĐOÀN VIỄN THÔNG QUÂN ĐỘI			Mã hiệu: TC.00.CNTT.xx
	THAM KHẢO HƯỚNG DẪN TIÊU CHUẨN AN TOÀN THÔNG TIN ỨNG DỤNG WEB			Ngày có hiệu lực:
				Ngày hết hiệu lực:
				Lần ban hành: 01
				Trang: 4/21
	ĐV soạn thảo	TT.ANM	ĐV kiểm tra	PCNTT - TĐ
	Phạm Vi	Toàn Tập đoàn.		

1.8. Chỉ sử dụng phương thức POST để submit thông tin định danh, khuyến nghị sử dụng HTTPS cho đường truyền để tăng tính bảo mật.

2. Quản lý phiên đăng nhập

2.1. Session phải được quản lý bởi server, sinh ngẫu nhiên và độ dài tối thiểu là 128 bit.

Ví dụ sử dụng sessionId được sinh bởi Tomcat server, giá trị sessionId đã đảm bảo được sinh ngẫu nhiên và độ dài mặc định là 128 bit.

2.2. Session phải được thiết lập thời gian timeout, giá trị timeout cần cân bằng giữa nhu cầu thương mại và yếu tố bảo mật.

Ví dụ trên tomcat 7.0, thiết lập cấu hình timeout tại *tomcat_path/conf/web.xml*:

```
<!-- ===== Default Session Configuration ===== -->
<!-- You can set the default session timeout (in minutes) for all newly -->
<!-- created sessions by modifying the value below. -->
<session-config>
<session-timeout>30</session-timeout>
</session-config>
```

2.3. Tạo mới session sau khi đăng nhập thành công.

2.4. Xóa giá trị sessionId và các dữ liệu gắn với session đó khi người dùng đăng xuất.

- Đối với cả 2 trường hợp 2.3 và 2.4, Java hỗ trợ hàm invalidate() thực hiện việc xóa sessionId và toàn bộ dữ liệu lưu trên sessionId đó. Khi client submit lên sessionId không còn tồn tại trên server, server sẽ tự động sinh 1 sessionId mới và trả về cho client.

```
// sau khi đăng nhập thành công hoặc đăng xuất gọi hàm
session.invalidate();
```

2.5. Cấu hình thuộc tính “Secure” đối với các ứng dụng sử dụng HTTPS và “HTTP-Only” cho trường session cookie.


- Cấu hình thuộc tính “Secure” trong file *tomcat_path/conf/web.xml*:

```
<Connector port="8443" protocol="org.apache.coyote.http11.Http11Protocol"
maxThreads="150" SSLEnabled="true" scheme="https" secure="true"
clientAuth="false" sslProtocol="TLS" />
```

- Cấu hình “HTTP-Only” trong file *tomcat_path/conf/context.xml*:

```
<Context useHttpOnly="true">
...
</Context>
```

2.6. Đối với các chức năng quan trọng có tương tác với database, ứng với mỗi phiên cần sinh thêm 1 token ngẫu nhiên, và thực hiện kiểm tra tính hợp lệ của token này trước khi xử lý truy vấn từ người dùng.

 Hãy nói theo cách của bạn	TẬP ĐOÀN VIỄN THÔNG QUÂN ĐỘI			Mã hiệu: TC.00.CNTT.xx
	THAM KHẢO HƯỚNG DẪN TIÊU CHUẨN AN TOÀN THÔNG TIN ỨNG DỤNG WEB			Ngày có hiệu lực:
				Ngày hết hiệu lực:
				Lần ban hành: 01
				Trang: 5/21
	ĐV soạn thảo	TT.ANM	ĐV kiểm tra	PCNTT - TĐ
	Phạm Vi	Toàn Tập đoàn.		

- Sử dụng cơ chế “token interceptor” được hỗ trợ bởi Struts2 như sau:


Bước 1:

<!-- Trong file struts.xml:

1. Khai báo thêm interceptor “tokenSession” vào danh sách interceptor hiện có.
2. Khai báo trang thông báo lỗi mặc định trả về client khi token được submit không hợp lệ.
3. Khai báo danh sách các hàm cần thực hiện validate token.

-->

```
<package name="actions" namespace="/Customer" extends="struts-default">
  <interceptors>
    <interceptor-stack name="defaultSecurityStack">
      <interceptor-ref name="defaultStack" />
      <interceptor-ref name="tokenSession">
        <param name="excludeMethods">*</param>
      </interceptor-ref>
    </interceptor-stack>
  </interceptors>
  <default-interceptor-ref name="defaultSecurityStack" />
  <global-results>
    <result name="invalid.token">/WEB-INF/jsp/error.jsp</result>
  </global-results>
  <action name="addCustomerAction"
    class="com.baseline.actions.CustomerAction" method="addCustomer"
  >
    <interceptor-ref name="defaultSecurityStack">
      <param name="tokenSession.includeMethods"> addCustomer
    </param>
    </interceptor-ref>
    <result name="success">/WEB-INF/jsp/customer.jsp</result>
  </action>
  <action name="listCustomerAction"
    class="com.baseline.actions.CustomerAction" method="listCustomer"
  >
```

 Hãy nói theo cách của bạn	TẬP ĐOÀN VIỄN THÔNG QUÂN ĐỘI			Mã hiệu: TC.00.CNTT.xx
	THAM KHẢO HƯỚNG DẪN TIÊU CHUẨN AN TOÀN THÔNG TIN ỨNG DỤNG WEB			Ngày có hiệu lực:
				Ngày hết hiệu lực:
				Lần ban hành: 01
				Trang: 6/21
	ĐV soạn thảo	TT.ANM	ĐV kiểm tra	PCNTT - TĐ
	Phạm Vi	Toàn Tập đoàn.		

<pre> <result name="success">/WEB-INF/jsp/customer.jsp</result> </action> </package> </pre> <p>Bước 2:</p> <p><!-- Trong file customer.jsp:</p> <p>Trong form submit tương ứng của các hàm đã khai báo tại bước 1.3, thêm thẻ <s:token/></p> <pre> --> <s:form action="addCustomerAction" > <s:token/> <s:property value="errMsg" /> <s:textfield name="name" label="Name" value="" /> <s:password name="password" label="Password" value="" /> <s:textarea name="address" label="Address" value="" cols="50" rows="5" /> <s:submit /> </s:form> </pre>
--


3. Phân quyền

- 3.1. Kiểm tra phân quyền dựa trên các đối tượng được lưu tại server (VD: tham số lưu trên session server, dữ liệu lưu trên DB,...).
- 3.2. Phân quyền tối thiểu, chỉ đáp ứng đủ chức năng và tài nguyên cho người dùng/ứng dụng.
- 3.3. Phía giao diện người dùng: Chỉ hiển thị các thành phần giao diện, đường dẫn, hàm,... tương ứng với quyền của người dùng.
- 3.4. Phía server: Kiểm tra quyền tác động của người dùng/ứng dụng trên các hàm và tài nguyên tương ứng trước khi thực hiện bất cứ tác vụ nào tới hệ thống.
- 3.5. Phải có tính năng xóa phiên làm việc hiện tại của người dùng hoặc các cơ chế tương đương đối với các trường hợp quyền người dùng bị thay đổi hoặc bị disable bởi người dùng có thẩm quyền.
- 3.6. Không đặt trang quản trị public internet, trong trường hợp bắt buộc phải đặt public cần giới hạn các IP được phép truy cập hoặc sử dụng cơ chế xác thực đa nhân tố (multiple authentications).

4. Tương tác với back-end

- Mã hóa các dữ liệu nhạy cảm trước khi lưu trữ (thông tin tài khoản ngân hàng, private key,...).
 - Sử dụng thuật toán mã hóa AES, secret key lưu trữ trong file cấu hình đặt trên DB.

<pre> public class Encryptor { private static Logger LOGGER = </pre>

 Hãy nói theo cách của bạn	TẬP ĐOÀN VIỄN THÔNG QUÂN ĐỘI			Mã hiệu: TC.00.CNTT.xx
	THAM KHẢO HƯỚNG DẪN TIÊU CHUẨN AN TOÀN THÔNG TIN ỨNG DỤNG WEB			Ngày có hiệu lực:
				Ngày hết hiệu lực:
				Lần ban hành: 01
				Trang: 7/21
	ĐV soạn thảo	TT.ANM	ĐV kiểm tra	PCNTT - TĐ
	Phạm Vi	Toàn Tập đoàn.		

```

LoggerFactory.getLogger(Encryptor.class);
private SecretKey secret;

public Encryptor(String secretKey) {
    /* Derive the key, given secretKey and salt. */
    SecretKeyFactory factory;
    try {
        factory = SecretKeyFactory.getInstance("PBKDF2WithHmacSHA1");
        KeySpec spec = new PBEKeySpec(secretKey.toCharArray(),
            secretKey.getBytes(), 65536, 128);
        SecretKey tmp = factory.generateSecret(spec);
        secret = new SecretKeySpec(tmp.getEncoded(), "AES");
    } catch (NoSuchAlgorithmException | InvalidKeySpecException e) {
        LOGGER.error("Khoi tao ma hoa khong thanh cong", e);
        throw new RuntimeException("Khoi tao ma hoa khong thanh cong",
e);
    }
}


public String encrypt(String message) {
    /* Encrypt the message. */
    try {
        Cipher cipher = Cipher.getInstance("AES/CBC/PKCS5Padding");
        cipher.init(Cipher.ENCRYPT_MODE, secret);
        AlgorithmParameters params = cipher.getParameters();
        byte[] iv =
params.getParameterSpec(IvParameterSpec.class).getIV();
        byte[] ciphertext = cipher.doFinal(message.getBytes("UTF-8"));
        byte[] result = new byte[iv.length + ciphertext.length];
        System.arraycopy(iv, 0, result, 0, iv.length);
        System.arraycopy(ciphertext, 0, result, iv.length,
            ciphertext.length);
        // encode base64 string
        return Base64.encodeBase64String(result);

    } catch (NoSuchAlgorithmException | NoSuchPaddingException |
InvalidKeyException | InvalidParameterSpecException |
IllegalBlockSizeException | BadPaddingException |
UnsupportedEncodingException e) {
        LOGGER.error("Ma hoa khong thanh cong", e);
        throw new RuntimeException("Ma hoa khong thanh cong", e);
    }
}

public String decrypt(String message) {
    // decode base64 string
    byte[] result = Base64.decodeBase64(message);
    byte[] iv = Arrays.copyOfRange(result, 0, 16);
    byte[] m = Arrays.copyOfRange(result, 16, result.length);

    /* Decrypt the message, given derived key and initialization
vector. */
    try {
        Cipher decryptor = Cipher.getInstance("AES/CBC/PKCS5Padding");
        decryptor
            .init(Cipher.DECRYPT_MODE, secret, new

```


 Hãy nói theo cách của bạn	TẬP ĐOÀN VIỄN THÔNG QUÂN ĐỘI			Mã hiệu: TC.00.CNTT.xx
	THAM KHẢO HƯỚNG DẪN TIÊU CHUẨN AN TOÀN THÔNG TIN ỨNG DỤNG WEB			Ngày có hiệu lực:
				Ngày hết hiệu lực:
				Lần ban hành: 01
				Trang: 8/21
	ĐV soạn thảo	TT.ANM	ĐV kiểm tra	PCNTT - TĐ
	Phạm Vi	Toàn Tập đoàn.		

```

IvParameterSpec(iv));
    return new String(decryptor.doFinal(m), "UTF-8");

    } catch (NoSuchAlgorithmException | NoSuchPaddingException |
InvalidKeyException | InvalidAlgorithmParameterException |
UnsupportedEncodingException | IllegalBlockSizeException |
BadPaddingException e) {
        LOGGER.error("Giải mã không thành công", e);
        throw new RuntimeException("Giải mã không thành công", e);
    }
}
}

```

– Sử dụng:

```

// 1. Lưu trữ ssecretKey trong file cấu hình an toàn
Properties prop = new Properties();
prop.load(Encryptor.class.getResourceAsStream("/config.properties"));
String secretKey = prop.getProperty("secretKey");

// 2. Mã hóa các dữ liệu nhạy cảm, ví dụ credit card
// 3. Sử dụng thuật toán mã hóa mạnh (AES), secretKey khó đoán
// 4. Xử lý exception trả về thông báo chung
// 5. Ghi log exception trên server

```

➤ SQL

4.1. Sử dụng mô hình truy vấn prepared statement (parameterized query) hoặc các hình thức tương đương.

– Ví dụ sử dụng mô hình truy vấn tham số trong Hibernate:

```

SessionFactory sessionFactory = HibernateUtil.getSessionFactory();
Session session = sessionFactory.openSession();
Query createQuery = session.createQuery("from Customer where name =
:username");
createQuery.setParameter("username", username);
List list = createQuery.list();
session.close();

```

4.2. Trong 1 số trường hợp không sử dụng được các mô hình ở trên, cần thiết lập danh sách whitelist các đầu vào mong muốn.

– Ví dụ với trường hợp không thể sử dụng mô hình truy vấn tham số là “order by” query:

```


private static final String[] orderByWhitelist = new String[]{"name",
"address", "age"};
public static boolean isInOrderByWhitelist(String input){
    return Arrays.asList(orderByWhitelist).contains(input);
}

```

➤ NoSQL

4.1. Không mở service ra ngoài public internet, cài đặt trong môi trường mạng an toàn (trusted enviroments).

- Cấu hình chỉ mở localhost interface trong trường hợp ứng dụng cùng host với database, tắt các dịch vụ không cần thiết. Trường hợp, cần kết nối database remote, cấu hình firewall giới hạn ip cho phép kết nối.
- Ví dụ, với mongodb 3.0.4, cấu hình chỉ lắng nghe trên interface 127.0.0.1, tắt http, json, rest interface.

 Hãy nói theo cách của bạn	TẬP ĐOÀN VIỄN THÔNG QUÂN ĐỘI		Mã hiệu: TC.00.CNTT.xx
	THAM KHẢO HƯỚNG DẪN TIÊU CHUẨN AN TOÀN THÔNG TIN ỨNG DỤNG WEB		Ngày có hiệu lực:
			Ngày hết hiệu lực:
			Lần ban hành: 01
			Trang: 9/21
	ĐV soạn thảo	TT.ANM	ĐV kiểm tra
	Phạm Vi	Toàn Tập đoàn.	
			PCNTT - TĐ

```
net:
port: 27017
bindIp: 127.0.0.1
http:
  enabled: false
  JSONPEnabled: false
  RESTInterfaceEnabled: false
```

4.2. Đối với các hệ NoSQL có hỗ trợ xác thực, cần cấu hình xác thực khi truy cập. Mặc định các hệ NoSQL tắt tính năng xác thực, cụ thể đối với mongodb 3.0.4, sau khi cài đặt, server chưa có người dùng và cũng không bật xác thực, thực hiện các bước sau để tạo người dùng với các quyền tương ứng và bật tính năng xác thực:

– Tạo người dùng quản trị server

```
use admin
db.createUser(
{
  user: "administrator",
  pwd: "M@tkhauKh0Do@n",
  roles: [ { role: "userAdminAnyDatabase", db: "admin" } ]
}
)
```

– Tạo người dùng quản trị cho 1 database (ví dụ DB baseline)

```
use baseline
db.createUser(
{
  user: "admin_baseline",
  pwd: "M@tkhauKh0Do@n",
  roles: [ { role: "userAdmin", db: "baseline" } ]
}
)
```

– Tạo người dùng sử dụng DB đó (có quyền thêm, sửa, xóa truy vấn)


```
use baseline
db.createUser(
{
  user: "user_baseline",
  pwd: "M@tkhauKh0Do@n",
  roles: [ { role: "readWrite", db: "baseline" } ]
}
)
```

– Cuối cùng, cấu hình mongodb bật tính năng xác thực và phân quyền

```
security:
  authorization: true
```

– Sử dụng mongo-java-driver để kết nối đến mongodb

```
String user = "user_baseline";
String database = "baseline";
```

 Hãy nói theo cách của bạn	TẬP ĐOÀN VIỄN THÔNG QUÂN ĐỘI			Mã hiệu: TC.00.CNTT.xx
	THAM KHẢO HƯỚNG DẪN TIÊU CHUẨN AN TOÀN THÔNG TIN ỨNG DỤNG WEB			Ngày có hiệu lực:
				Ngày hết hiệu lực:
				Lần ban hành: 01
				Trang: 10/21
	ĐV soạn thảo	TT.ANM	ĐV kiểm tra	PCNTT - TĐ
	Phạm Vi	Toàn Tập đoàn.		

```
char[] password = "M@tkhauKh0Do@n".toCharArray();
MongoCredential credential =
    MongoCredential.createCredential(user, database, password);
MongoClient mongo = new MongoClient(
    new ServerAddress("localhost", 27017),
    Arrays.asList(new MongoCredential[] { credential }));
MongoDatabase baseline = mongo.getDatabase("baseline");
```

4.3. Phụ thuộc vào hệ NoSQL sử dụng, sử dụng các api hỗ trợ truy vấn an toàn hoặc thực hiện escape các ký tự đặc biệt khi xây dựng câu truy vấn.

Đối với mongodb, sử dụng api mongo-java-driver khi thực hiện các thao tác thêm, sửa, xóa, tìm kiếm:

– Thêm document cho collection

```
MongoCollection<Document> users = guideline.getCollection("books");
Document doc = new Document("name", "MongoDB")
    .append("type", "database")
    .append("count", 1)
    .append("info", new Document("x", 203)
    .append("y", 102));
books.insertOne(doc);
```

– Sửa document trong collection

```
books.updateOne(eq("name", "MongoDB"),
    new Document("$set", new Document("name", "MongoDB 3.04")));
UpdateResult updateResult = collection.updateMany(1t("count", 2),
    new Document("$inc", new Document("count", 1)));
```

– Xóa document trong collection

```
books.deleteOne(eq("name", "MongoDB 3.04"));
DeleteResult deleteResult = books.deleteMany(gte("count", 2));
```

– Tìm kiếm document trong collection. Cách an toàn là sử dụng các api Filters, Sorts, Projections để tạo truy vấn tìm kiếm

```
import static com.mongodb.client.model.Filters.*;
Document myDoc = books.find(eq("name", "MongoDB")).first();
myDoc = books.find(exists("count")).sort(descending("count")).first();
myDoc = books.find().projection(excludeId()).first();
```

– Khi sử dụng Filters.where(String javascriptExpression). Tránh truyền dữ liệu người dùng vào biểu thức javascript, nếu bắt buộc phải truyền thì thực hiện escape javascript

```
String name = "'" + true + "' | '1"; // request.getParameter("name")


// unsafe
myDoc = books.find(where("this.name == '" + name + "'")).first();

// safe
myDoc = books.find(where("this.name == '" +
StringEscapeUtils.escapeEcmaScript(name) + "'")).first();
```

➤ **XPath**

4.1. Thiết lập danh sách whitelist các ký tự đầu vào mong muốn, đầu vào nên là tập hợp của chữ cái, chữ số.

```
public static boolean isValidInput(String input) {
    if (input == null) {
        return false;
    }
```

 Hãy nói theo cách của bạn	TẬP ĐOÀN VIỄN THÔNG QUÂN ĐỘI			Mã hiệu: TC.00.CNTT.xx
	THAM KHẢO HƯỚNG DẪN TIÊU CHUẨN AN TOÀN THÔNG TIN ỨNG DỤNG WEB			Ngày có hiệu lực:
				Ngày hết hiệu lực:
				Lần ban hành: 01
				Trang: 11/21
	ĐV soạn thảo	TT.ANM	ĐV kiểm tra	PCNTT - TĐ
	Phạm Vi	Toàn Tập đoàn.		

```

    }
    return input.matches ("^[a-zA-Z0-9]*$");
}

```

4.2. Lập danh sách blacklist các ký tự đặc biệt (() = '[] : , * / và dấu cách), loại bỏ các đầu vào có chứa các ký tự nằm trong blacklist.

```

public static boolean isValidInput(String input) {
    if (input == null) {
        return true;
    }
    Pattern p = Pattern.compile ("[ ( ) = '\\[ \\] : , * / ]");
    Matcher m = p.matcher(input);
    return m.find();
}

```

➤ LDAP

4.1. Thiết lập danh sách whitelist các ký tự đầu vào mong muốn, đầu vào nên là tập hợp của chữ cái, chữ số.

```

public static boolean isValidInput(String input) {
    if (input == null){
        return false;
    }
    return input.matches ("^[a-zA-Z0-9]*$");
}

```

4.2. Lập danh sách blacklist các ký tự đặc biệt (() ; , * | & = và nullbyte), loại bỏ các đầu vào có chứa các ký tự nằm trong blacklist.

```

public static boolean isValidInput(String input) {
    if (input == null) {
        return true;
    }
    Pattern p = Pattern.compile ("[ ( ) ; , * | & = \\0 ]");
    Matcher m = p.matcher(input);
    return m.find();
}

```

➤ Tương tác với OS

4.1. Sử dụng các API hỗ trợ việc thực thi câu lệnh hệ thống.


Ví dụ check reachable thông qua hostname được truyền từ người dùng:

```

public static void useSafeApi() throws IOException {
    String hostname = "google.com && echo VULNERABLE"; //
    request.getParameter("hostname")
    try {
        InetAddress inetAddress = InetAddress.getByName(hostname);
        boolean reachable = inetAddress.isReachable(5000);
        System.out.println("Network Reachable " + hostname + ": " +
        reachable);
    } catch (Exception e) {
        System.out.println("except: " + e);
    }
}

```

4.2. Không truyền trực tiếp dữ liệu người dùng truyền lên tới OS, trong trường hợp bắt buộc cần thiết lập danh sách whitelist các đầu vào mong muốn.

 Hãy nói theo cách của bạn	TẬP ĐOÀN VIỄN THÔNG QUÂN ĐỘI			Mã hiệu: TC.00.CNTT.xx
	THAM KHẢO HƯỚNG DẪN TIÊU CHUẨN AN TOÀN THÔNG TIN ỨNG DỤNG WEB			Ngày có hiệu lực:
				Ngày hết hiệu lực:
				Lần ban hành: 01
				Trang: 12/21
	ĐV soạn thảo	TT.ANM	ĐV kiểm tra	PCNTT - TĐ
	Phạm Vi	Toàn Tập đoàn.		

Ví dụ sử dụng câu lệnh “ping” với tham số hostname được truyền từ người dùng:

```
public static void processWhitelist() throws IOException,
InterruptedException {
    // Giả thiết hostname = request.getParameter("hostname");
    String hostname = "google.com && echo VULNERABLE";

    String DOMAIN_NAME_PATTERN = "^((?!-)[A-Za-z0-9-]{1,63}(?!-)\.)+[A-
Za-z]{2,6}$";
    String IPADDRESS_PATTERN
        = "^[01]?\\d\\d?|2[0-4]\\d|25[0-5])\\.\"
        + "([01]?\\d\\d?|2[0-4]\\d|25[0-5])\\.\"
        + "([01]?\\d\\d?|2[0-4]\\d|25[0-5])\\.\"
        + "([01]?\\d\\d?|2[0-4]\\d|25[0-5])$";

    if (!hostname.matches(DOMAIN_NAME_PATTERN) &&
        !hostname.matches(IPADDRESS_PATTERN)) {
        System.out.println("Hostname is not valid");
        return;
    }

    Process p = Runtime.getRuntime().exec("cmd /c ping " + hostname);
    p.waitFor();

    try (BufferedReader reader = new BufferedReader(new
        InputStreamReader(p.getInputStream()))) {
        String line;
        while ((line = reader.readLine()) != null) {
            System.out.println(line);
        }
    }
}
```


➤ Tương tác với file

- 4.1. Không truyền trực tiếp dữ liệu từ người dùng đến các hàm include file.
- 4.2. Lập danh sách whitelist các định dạng file được phép upload.

Struts2 có hỗ trợ interceptor “fileUpload” như sau:

```
<!-- Khai báo interceptor vào action cần kiểm tra file upload -->
<action name="resultAction" class="security.backend.file.FileUploadAction">
    <interceptor-ref name="fileUpload">
        <param
name="allowedTypes">image/png,image/gif,image/jpeg</param>
        <param name="allowedExtensions">jpg,png,gif</param>
        <param name="maximumSize">10240</param>
    </interceptor-ref>
    <interceptor-ref name="defaultStack"/>
    <result name="success">/WEB-INF/jsp/backendfile/result.jsp</result>
    <result name="input">/WEB-INF/jsp/backendfile/fileupload.jsp</result>
</action>
```

- 4.3. Validate file hợp lệ bằng cách kiểm tra đồng thời file header, phần mở rộng của file và nội dung file.

 Hãy nói theo cách của bạn	TẬP ĐOÀN VIỄN THÔNG QUÂN ĐỘI			Mã hiệu: TC.00.CNTT.xx
	THAM KHẢO HƯỚNG DẪN TIÊU CHUẨN AN TOÀN THÔNG TIN ỨNG DỤNG WEB			Ngày có hiệu lực: Ngày hết hiệu lực:
				Lần ban hành: 01
				Trang: 13/21
	ĐV soạn thảo	TT.ANM	ĐV kiểm tra	PCNTT - TĐ
	Phạm Vi	Toàn Tập đoàn.		

Ví dụ upload 1 file ảnh lên server:

```
private static final String[] allowedExtension = new String[]{"jpg", "png", "gif"};
private static final String[] allowedMimetype = new String[]{"image/jpeg", "image/png", "image/gif"};
private boolean validateFile() {
    boolean ok = true;

    // Kiểm tra phần mở rộng của file
    String extension = FilenameUtils.getExtension(fileUploadFileName);
    boolean extAccepted = Arrays.asList(allowedExtension).contains(extension);
    if (!extAccepted) {
        addActionError("Extension not allowed, Choose " + Arrays.toString(allowedExtension));
        ok &= false;
        if (!ok) return ok;
    }


    // Kiểm tra mime type
    boolean mimeTypeAccepted = Arrays.asList(allowedMimetype).contains(fileUploadContentType);
    if (!mimeTypeAccepted) {
        addActionError("Mime-Type not allowed, Choose " + Arrays.toString(allowedMimetype));
        ok &= false;
        if (!ok) return ok;
    }

    // Kiểm tra nội dung file có phải file ảnh hay không?
    try {
        BufferedImage image = ImageIO.read(fileUpload);
        ImageIO.write(image, extension, fileUpload);
    } catch (Exception e) {
        addActionError("Cannot read image file, File Upload must be image");
        ok &= false;
        if (!ok) return ok;
    }

    return ok;
}
```

- 4.4. Với các trường hợp không bắt buộc thì không lưu file upload trong thư mục web, bỏ quyền thực thi trên thư mục upload.
- 4.5. Khi cần refer tới các file tồn tại trên hệ thống cần thiết lập danh sách whitelist đầu vào mong muốn hoặc gán các giá trị định danh tương ứng cho các file thay vì truyền tên file.
- 4.6. Không trả về đường dẫn tuyệt đối của file.
- 4.7. Tất cả dữ liệu, tài nguyên hệ thống (báo cáo, file upload, file cấu hình...) không được lưu trong thư mục cho phép truy cập trực tiếp không qua xác thực.

➤ **Xử lý back-end HTTP request**

 Hãy nói theo cách của bạn	TẬP ĐOÀN VIỄN THÔNG QUÂN ĐỘI			Mã hiệu: TC.00.CNTT.xx
	THAM KHẢO HƯỚNG DẪN TIÊU CHUẨN AN TOÀN THÔNG TIN ỨNG DỤNG WEB			Ngày có hiệu lực:
				Ngày hết hiệu lực:
				Lần ban hành: 01
				Trang: 14/21
	ĐV soạn thảo	TT.ANM	ĐV kiểm tra	PCNTT - TD
	Phạm Vi	Toàn Tập đoàn.		

4.1. Khi tạo http request phía server, các tham số GET, POST cho request đó tránh tạo từ dữ liệu phía người dùng, hoặc phải được kiểm tra cẩn thận để chống ghi đè các tham số khác.

Ví dụ trong trường hợp tạo request với tham số POST được truyền từ người dùng, sử dụng hàm encode() của class URLEncoder để encode dữ liệu người dùng truyền lên, chỉ rõ kiểu encode, thường là UTF-8:

```
// Không an toàn
String unsafePostData = "action=transfer&amount=" + amount + "&recipient="
+ recipient;
// An toàn
String safePostData = "action=transfer&amount=" + URLEncoder.encode(amount,
"UTF-8") + "&recipient=" + URLEncoder.encode(recipient, "UTF-8");
```

➤ Tương tác với XML

4.1. Tắt tính năng external entity resolve và remote doctype retrieval của xml parser khi đọc dữ liệu xml.

– DOM Parser:

```
DocumentBuilderFactory dbFactory = DocumentBuilderFactory.newInstance();

// SAFE: disable xxe attack
// This is the PRIMARY defense. If DTDs (doctypes) are disallowed, almost
all XML entity attacks are prevented
// Xerces 2 only - http://xerces.apache.org/xerces2-j/features.html#disallow-doctype-decl
String FEATURE = "http://apache.org/xml/features/disallow-doctype-decl";
dbFactory.setFeature(FEATURE, true);

// If you can't completely disable DTDs, then at least do the following:
// Xerces 1 - http://xerces.apache.org/xerces-j/features.html#external-general-entities
// Xerces 2 - http://xerces.apache.org/xerces2-j/features.html#external-general-entities
FEATURE = "http://xml.org/sax/features/external-general-entities";
dbFactory.setFeature(FEATURE, false);


// Xerces 1 - http://xerces.apache.org/xerces-j/features.html#external-parameter-entities
// Xerces 2 - http://xerces.apache.org/xerces2-j/features.html#external-parameter-entities
FEATURE = "http://xml.org/sax/features/external-parameter-entities";
dbFactory.setFeature(FEATURE, false);

// and these as well, per Timothy Morgan's 2014 paper: "XML Schema, DTD,
and Entity Attacks" (see reference below)
dbFactory.setIncludeAware(false);
dbFactory.setExpandEntityReferences(false);
```

– JDOM Parser:

```
SAXBuilder builder = new SAXBuilder();
/* SAFE: */
builder.setExpandEntities(false); //Retain Entities
builder.setValidation(false);
```

– SAX Parser:

 Hãy nói theo cách của bạn	TẬP ĐOÀN VIỄN THÔNG QUÂN ĐỘI			Mã hiệu: TC.00.CNTT.xx
	THAM KHẢO HƯỚNG DẪN TIÊU CHUẨN AN TOÀN THÔNG TIN ỨNG DỤNG WEB			Ngày có hiệu lực:
				Ngày hết hiệu lực:
				Lần ban hành: 01
				Trang: 15/21
	ĐV soạn thảo	TT.ANM	ĐV kiểm tra	PCNTT - TĐ
	Phạm Vi	Toàn Tập đoàn.		

```

SAXParserFactory factory = SAXParserFactory.newInstance();

/* SAFE: */
// This is the PRIMARY defense. If DTDs (doctypes) are disallowed, almost
all XML entity attacks are prevented
// Xerces 2 only - http://xerces.apache.org/xerces2-
j/features.html#disallow-doctype-decl
String FEATURE = "http://apache.org/xml/features/disallow-doctype-decl";
factory.setFeature(FEATURE, true);

// If you can't completely disable DTDs, then at least do the following:
// Xerces 1 - http://xerces.apache.org/xerces-j/features.html#external-
general-entities
// Xerces 2 - http://xerces.apache.org/xerces2-j/features.html#external-
general-entities
FEATURE = "http://xml.org/sax/features/external-general-entities";
factory.setFeature(FEATURE, false);

// Xerces 1 - http://xerces.apache.org/xerces-j/features.html#external-
parameter-entities
// Xerces 2 - http://xerces.apache.org/xerces2-j/features.html#external-
parameter-entities
FEATURE = "http://xml.org/sax/features/external-parameter-entities";
factory.setFeature(FEATURE, false);

// and these as well, per Timothy Morgan's 2014 paper: "XML Schema, DTD,
and Entity Attacks" (see reference below)
factory.setIncludeAware(false);

```

– *StAXX Parser:*

```

XMLInputFactory factory = XMLInputFactory.newInstance();

/*SAFE: */
factory.setProperty(XMLInputFactory.IS_SUPPORTING_EXTERNAL_ENTITIES,
false);
factory.setProperty(XMLInputFactory.SUPPORT_DTD, false);

```

4.2. Kiểm tra dữ liệu người dùng, encode các kí tự đặc biệt (<>/) khi tạo dữ liệu xml.

5. Kiểm soát dữ liệu đầu vào


- 5.1. Việc kiểm tra dữ liệu đầu vào phải được thực hiện phía server.
- 5.2. Thực hiện việc kiểm tra dữ liệu từ tất cả các nguồn dữ liệu có tương tác với người dùng (Các tham số lấy từ GET/POST request, HTTP Headers, dữ liệu lấy từ DB, dữ liệu từ file upload,...).
- 5.3. Xác định rõ chuẩn định dạng encode của dữ liệu đầu vào, thực hiện validate dữ liệu sau khi đã decode đầu vào về 1 định dạng chuẩn và nhất quán.

– Cấu hình filter mapping trong file web.xml

```

<filter>
<filter-name>CharsetFilter</filter-name>
<filter-class>security.baseline.CharsetFilter</filter-class>
<init-param>
<param-name>requestEncoding</param-name>
<param-value>UTF-8</param-value>

```


 Hãy nói theo cách của bạn	TẬP ĐOÀN VIỄN THÔNG QUÂN ĐỘI			Mã hiệu: TC.00.CNTT.xx
	THAM KHẢO HƯỚNG DẪN TIÊU CHUẨN AN TOÀN THÔNG TIN ỨNG DỤNG WEB			Ngày có hiệu lực:
				Ngày hết hiệu lực:
				Lần ban hành: 01
				Trang: 16/21
	ĐV soạn thảo	TT.ANM	ĐV kiểm tra	PCNTT - TĐ
	Phạm Vi	Toàn Tập đoàn.		

```

</init-param>
</filter>

<filter-mapping>
<filter-name>CharsetFilter</filter-name>
<url-pattern>/*</url-pattern>
</filter-mapping>

```

– Xây dựng class CharsetFilter để thực hiện set encoding

```

public class CharsetFilter implements Filter {
    private String encoding;

    @Override
    public void init(FilterConfig config) throws ServletException {
        encoding = config.getInitParameter("requestEncoding");

        if (null == encoding) {
            encoding = "UTF-8";
        }
    }

    @Override
    public void doFilter(ServletRequest request, ServletResponse response,
        FilterChain next) throws IOException, ServletException {
        if (null == request.getCharacterEncoding()) {
            request.setCharacterEncoding(encoding);
        }
        response.setContentType("text/html; charset=UTF-8");
        response.setCharacterEncoding("UTF-8");
        next.doFilter(request, response);
    }
}

```

5.4. Validate kiểu dữ liệu, phạm vi, độ dài dữ liệu và định dạng dữ liệu.

Ví dụ một số kiểu validator được hỗ trợ bởi thư viện opensymphony.xwork2 như sau:

```


public class InputAction extends ActionSupport {

    private String name;
    private int age;
    private Date dob;
    private String url;
    private String email;
    private String income;

    public String getName() {
        return name;
    }

    @RequiredStringValidator(message = "Name must be required",
        shortCircuit = true, trim = true)
    @StringLengthFieldValidator(message = "Name length must be between 5
        and 12", shortCircuit = true, trim = true, minLength = "5", maxLength =
        "12")
    @RegexFieldValidator(message = "Name contain only letter, number, _",

```

 Hãy nói theo cách của bạn	TẬP ĐOÀN VIỄN THÔNG QUÂN ĐỘI			Mã hiệu: TC.00.CNTT.xx
	THAM KHẢO HƯỚNG DẪN TIÊU CHUẨN AN TOÀN THÔNG TIN ỨNG DỤNG WEB			Ngày có hiệu lực:
				Ngày hết hiệu lực:
				Lần ban hành: 01
				Trang: 17/21
	ĐV soạn thảo	TT.ANM	ĐV kiểm tra	PCNTT - TĐ
	Phạm Vi	Toàn Tập đoàn.		

```

type = ValidatorType.FIELD, regexExpression = "([a-zA-Z0-9_]*)"
    public void setName(String name) {
        this.name = name;
    }

    public int getAge() {
        return age;
    }

    @IntRangeFieldValidator(message = "Age must from 0 to 60",
shortCircuit = true, min = "0", max = "60")
    public void setAge(int age) {
        this.age = age;
    }

    public Date getDob() {
        return dob;
    }

    @DateRangeFieldValidator(message = "Date must in range 01/01/1955 to
31/12/2015", dateFormat = "dd/MM/yyyy", shortCircuit = true, min =
"01/01/1955", max = "31/12/2015")
    public void setDob(Date dob) {
        this.dob = dob;
    }

    public String getURL() {
        return url;
    }

    @UrlValidator(message = "Url format not correct", shortCircuit =
true)
    public String getUrl() {
        return url;
    }


    public void setUrl(String url) {
        this.url = url;
    }

    public String getEmail() {
        return email;
    }

    @RequiredFieldValidator(message = "Email is required", shortCircuit =
true)
    @EmailValidator(message = "Email format not correct", shortCircuit =
true)
    public void setEmail(String email) {
        this.email = email;
    }

    public String getIncome() {
        return income;
    }

```

 Hãy nói theo cách của bạn	TẬP ĐOÀN VIỄN THÔNG QUÂN ĐỘI			Mã hiệu: TC.00.CNTT.xx
	THAM KHẢO HƯỚNG DẪN TIÊU CHUẨN AN TOÀN THÔNG TIN ỨNG DỤNG WEB			Ngày có hiệu lực:
				Ngày hết hiệu lực:
				Lần ban hành: 01
				Trang: 18/21
	ĐV soạn thảo	TT.ANM	ĐV kiểm tra	PCNTT - TD
	Phạm Vi	Toàn Tập đoàn.		

```
@DoubleRangeFieldValidator(message = "Income is double in range 0.123 to 99.987", shortCircuit = true, minInclusive = "0.123", maxInclusive = "99.987")
public void setIncome(String income) {
    this.income = income;
}
}
```

5.5. Nếu dữ liệu đầu vào bắt buộc là các ký tự đặc biệt, cần thiết lập danh sách whitelist các ký tự đầu vào mong muốn.

6. Kiểm soát dữ liệu đầu ra

6.1. Phải chỉ rõ character encoding cho dữ liệu đầu ra.

Thực hiện tương tự mục 5.3.

6.2. Response body phải được encode theo ngữ cảnh sử dụng. Một số trường hợp phổ biến:

- Đầu ra là html, thực hiện html encode các ký tự đặc biệt (<"'&) từ các nguồn dữ liệu không an toàn (Các tham số lấy từ GET/POST request, HTTP Headers, dữ liệu lấy từ DB, dữ liệu từ file upload,... có thể điều khiển được bởi người dùng).

```
<!-- Ví dụ: Sử dụng thư viện JSTL encode dữ liệu in ra trên trang response trả về để chống lỗi XSS -->
```

Cách 1:

```
<%@ taglib prefix="fn" uri="http://java.sun.com/jsp/jstl/functions"%>
<h1> Hello ${fn:escapeXml(username)}! </h1>
```

Cách 2:

```
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<h1> Hello <c:out value="${username}">! </h1>
```

- Đầu ra là json, thực hiện encode dữ liệu trả về dạng object, không trả về dạng mảng.

```
// Thay vì khởi tạo đối tượng kiểu ArrayList
List<String> lists = new ArrayList<String>(); //UNSAFE
// Khởi tạo đối tượng kiểu HashMap như sau
Map<String, String> maps = new HashMap<String, String>(); //SAFE
```


6.3. Response header: lọc bỏ các ký tự đặc biệt (\n, \r) do dữ liệu người dùng truyền vào.

```
// remove control character [\x00-\x1F\x7F]
header = header.replaceAll("\\p{Cntrl}", "");
```

6.4. Cookie trả về cần giới hạn tối thiểu nhất các thuộc tính (domain, path, httponly, expire, secure). Tránh lưu trữ các dữ liệu nhạy cảm trên cookie, nếu cần lưu trữ các dữ liệu nhạy cảm thì phải thực hiện mã hóa các dữ liệu này với thuật toán đối xứng mạnh và key chỉ được lưu trên server.

```
Cookie cookie = new Cookie("cookieOnAdminPage", cookieValue);
/* 1. Giới hạn thuộc tính của cookie */

// ví dụ cookie cookieOnAdminPage để nhớ đăng nhập trên trang /admin/
cookie.setHttpOnly(true);
cookie.setMaxAge(7*24*60*60);
cookie.setPath("/admin/");
cookie.setComment("Cookie nho dang nhap nguoi dung, co gia tri tren link
```

 Hãy nói theo cách của bạn	TẬP ĐOÀN VIỄN THÔNG QUÂN ĐỘI			Mã hiệu: TC.00.CNTT.xx
	THAM KHẢO HƯỚNG DẪN TIÊU CHUẨN AN TOÀN THÔNG TIN ỨNG DỤNG WEB			Ngày có hiệu lực:
				Ngày hết hiệu lực:
				Lần ban hành: 01
				Trang: 19/21
	ĐV soạn thảo	TT.ANM	ĐV kiểm tra	PCNTT - TĐ
	Phạm Vi	Toàn Tập đoàn.		

```

/admin trong thời gian 7 ngày");
//cookie.setSecure(true); // uncomment nếu sử dụng https
cookie.setValue(cookieValue);
response.addCookie(cookie);

/* Mã hóa cookie */
// Thực hiện tương tự mục 4. Tương tác với backend => Mã hóa các dữ liệu
nhạy cảm trước khi lưu trữ

```

6.5. Hạn chế việc chuyển hướng, chuyển tiếp đến các URI khác. Nếu ứng dụng có chức năng này cần phải lập danh sách whitelist các URI được phép thực hiện chuyển hướng, chuyển tiếp.

```

private static final String[] allowedURL = new
String[]{"voffice.viettel.vn", "thongtinnhansu.viettel.vn",
"viettelfamily.vn"};

public static boolean isAllowedURL(String url) {
    return Arrays.asList(allowedURL).contains(url);
}

```

7. Kiểm soát ngoại lệ và ghi log ứng dụng

7.1. Xử lý các ngoại lệ bằng try-catch và trả về các thông báo lỗi chung đã custom, thông báo lỗi trả về không được chứa các thông tin nhạy cảm của người dùng, hệ thống,...

```

<!--
1. Trong file web.xml cấu hình trang thông báo lỗi khi có exception.
2. Định nghĩa thông báo lỗi trả về trong file error.jsp.
-->
<web-app>
  <display-name>Archetype Created Web Application</display-name>
  <error-page>
    <exception-type>java.lang.Exception</exception-type>
    <location>/WEB-INF/jsp/error.jsp</location>
  </error-page>
  <filter>
    <filter-name>struts2</filter-name>
    <filter-
class>org.apache.struts2.dispatcher.FilterDispatcher</filter-class>
  </filter>
  <filter-mapping>
    <filter-name>struts2</filter-name>
    <url-pattern>/*</url-pattern>
  </filter-mapping>
</web-app>

```

7.2. Các thông tin lỗi, ngoại lệ này phải được log lại để phục vụ bảo trì, xác định nguyên nhân lỗi ứng dụng.

Sử dụng thư viện log4j để ghi log ứng dụng.

7.3. File log phải được đặt tại thư mục an toàn ngoài thư mục web.


7.4. Không log lại các dữ liệu nhạy cảm (thông tin người dùng, session id, thông tin hệ thống).

Mục 7.3 và 7.4 thực hiện cấu hình log4j tại file log4j.xml như sau:

```

<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE log4j:configuration SYSTEM "log4j.dtd">

```

 Hãy nói theo cách của bạn	TẬP ĐOÀN VIỄN THÔNG QUÂN ĐỘI			Mã hiệu: TC.00.CNTT.xx
	THAM KHẢO HƯỚNG DẪN TIÊU CHUẨN AN TOÀN THÔNG TIN ỨNG DỤNG WEB			Ngày có hiệu lực:
				Ngày hết hiệu lực:
				Lần ban hành: 01
				Trang: 20/21
	ĐV soạn thảo	TT.ANM	ĐV kiểm tra	PCNTT - TĐ
	Phạm Vi	Toàn Tập đoàn.		

```
<log4j:configuration xmlns:log4j="http://jakarta.apache.org/log4j/">

  <appender name="ConsoleAppender"
class="org.apache.log4j.ConsoleAppender">
    <layout class="org.apache.log4j.PatternLayout">
      <param name="ConversionPattern" value=
        "Time: %d{ISO8601} %-5p %n
        Location: %l%n
        Message: %m %n %n"/>
    </layout>
  </appender>

  <appender name="wflog"
class="org.apache.log4j.DailyRollingFileAppender">
    <param name="file" value="${catalina.base}/logs/wflog.log" />
    <param name="append" value="true" />
    <param name="datePattern" value="'. 'yyyy-MM-dd-HH" />
    <layout class="org.apache.log4j.PatternLayout">
      <param name="ConversionPattern"
        value="%d{HH:mm:ss.SSS} %t %C{1} %-5p %n%l%n%m%n%n" />
    </layout>
  </appender>
<!-- Ví dụ cấu hình logger cho package com.viettel -->
<logger name="com.viettel">
  <level value="error"/>
  <appender-ref ref="ConsoleAppender"/>
  <appender-ref ref="wflog"/>
</logger>

<root>
  <level value ="error" />
  <appender-ref ref="ConsoleAppender"/>
  <appender-ref ref="wflog"/>
</root>
</log4j:configuration>
```

7.5. Giới hạn người dùng cho phép truy cập file log.

8. Sử dụng framework, lib (third-party components)

8.1. Loại các thành phần, lib không cần thiết.

8.2. Sử dụng phiên bản mới nhất của framework tại thời điểm phát triển ứng dụng.

8.3. Thường xuyên cập nhật các bản vá lỗi cho framework.

8.4. Tắt chế độ development của framework khi triển khai ứng dụng thực tế.


Cấu hình trong file struts.xml:

```
<struts>
  <constant name="struts.devMode" value="false" />
</struts>
```

9. Xử lý bussiness logic

Xử lý business logic phụ vào từng ứng dụng nhưng lưu ý:

9.1. Lập trình viên phải nắm rõ được toàn bộ luồng nghiệp vụ của ứng dụng, phải xây dựng các cây testcase cho từng nghiệp vụ để tránh bỏ sót các trường hợp có thể xảy ra.

 Hãy nói theo cách của bạn	TẬP ĐOÀN VIỄN THÔNG QUÂN ĐỘI			Mã hiệu: TC.00.CNTT.xx
	THAM KHẢO HƯỚNG DẪN TIÊU CHUẨN AN TOÀN THÔNG TIN ỨNG DỤNG WEB			Ngày có hiệu lực:
				Ngày hết hiệu lực:
				Lần ban hành: 01
				Trang: 21/21
	ĐV soạn thảo	TT.ANM	ĐV kiểm tra	PCNTT - TĐ
	Phạm Vi	Toàn Tập đoàn.		

9.2. Các chức năng quan trọng (ví dụ chuyển khoản ngân hàng), sử dụng lock hoặc các hình thức tương đương để tránh lỗi race condition.