

## [ERR07-J. Do not throw RuntimeException, Exception, or Throwable](#)

Red color indicates CERT code violation.

File path: Assume we are in source code home folder.

File Path : frameworks/base/core/java/android/net/nsd/NsdManager.java

Line No: 439

```
private void init() {
    final Messenger messenger = getMessenger();
    if (messenger == null) throw new RuntimeException("Failed to initialize");
    HandlerThread t = new HandlerThread("NsdManager");
    t.start();
    mHandler = new ServiceHandler(t.getLooper());
    mAsyncChannel.connect(mContext, mHandler, messenger);
    try {
        mConnected.await();
    } catch (InterruptedException e) {
        Log.e(TAG, "interrupted wait at init");
    }
}
```

File Path : frameworks/base/core/java/android/net/NetworkPolicyManager.java

Line No: 310

```
if (!ALLOW_PLATFORM_APP_POLICY) {
    final PackageManager pm = context.getPackageManager();
    final HashSet<Signature> systemSignature;
    try {
        systemSignature = Sets.newHashSet(
            pm.getPackageInfo("android", GET_SIGNATURES).signatures);
    } catch (NameNotFoundException e) {
        throw new RuntimeException("problem finding system signature", e);
    }
}
```

File Path : frameworks/base/core/java/android/net/NetworkUtils.java

Line No: 317

```
public static void maskRawAddress(byte[] array, int prefixLength) {
    if (prefixLength < 0 || prefixLength > array.length * 8) {
        throw new RuntimeException("IP address with " + array.length +
            " bytes has invalid prefix length " + prefixLength);
    }
}
```

File Path : frameworks/base/core/java/android/net/NetworkUtils.java

Line No: 347

```
public static InetAddress getNetworkPart(InetAddress address, int prefixLength) {
    byte[] array = address.getAddress();
    maskRawAddress(array, prefixLength);

    InetAddress netPart = null;
    try {
        netPart = InetAddress.getByAddress(array);
    } catch (UnknownHostException e) {
        throw new RuntimeException("getNetworkPart error - " + e.toString());
    }
    return netPart;
}
```

File Path : frameworks/base/core/java/android/net/TrafficStats.java

Line No:272

```
public static void incrementOperationCount(int tag, int operationCount) {

    final int uid = android.os.Process.myUid();

    try {

        getStatsService().incrementOperationCount(uid, tag, operationCount);

    } catch (RemoteException e) {

        throw new RuntimeException(e);

    }

}
```

File Path : frameworks/base/core/java/android/net/TrafficStats.java

Line No: 615

```
private static NetworkStats getDataLayerSnapshotForUid(Context context) {

    // TODO: take snapshot locally, since proc file is now visible
    final int uid = android.os.Process.myUid();
    try {
        return getStatsService().getDataLayerSnapshotForUid(uid);
    } catch (RemoteException e) {
        throw new RuntimeException(e);
    }
}
```

```
}
```

File Path : frameworks/base/core/java/android/net/TrafficStats.java  
Line No: 628

```
private static String[] getMobileIfaces() {  
    try {  
        return getStatsService().getMobileIfaces();  
    } catch (RemoteException e) {  
        throw new RuntimeException(e);  
    }  
}
```

File Path : /frameworks/base/core/java/android/net/LocalSocketImpl.java  
Line No: 428

```
protected void sendUrgentData(int data) throws IOException  
{  
    throw new RuntimeException ("not implmed");  
}
```

AccountManager.java

Note: More than 10 ERR07 violations found in AccountManager.java. Some are commented like won't ever happen and will never happen

File Path : frameworks/base/core/java/android/accounts/AccountManager.java  
Line No: 337

```
public String getPassword(final Account account) {  
    if (account == null) throw new IllegalArgumentException("account is null");  
    try {  
        return mService.getPassword(account);  
    } catch (RemoteException e) {  
        // won't ever happen  
        throw new RuntimeException(e);  
    }  
}
```

File Path : frameworks/base/core/java/android/accounts/AccountManager.java  
Line No: 366

```
public String getUserData(final Account account, final String key) {  
    if (account == null) throw new IllegalArgumentException("account is null");  
    if (key == null) throw new IllegalArgumentException("key is null");  
    try {  
        return mService.getUserData(account, key);  
    } catch (RemoteException e) {  
        // won't ever happen  
        throw new RuntimeException(e);  
    }  
}
```

File Path : frameworks/base/core/java/android/accounts/AccountManager.java  
Line No: 386

```
public AuthenticatorDescription[] getAuthenticatorTypes() {  
    try {  
        return mService.getAuthenticatorTypes(UserHandle.getCallingUserId());  
    } catch (RemoteException e) {  
        // will never happen  
        throw new RuntimeException(e);  
    }  
}
```

File Path : frameworks/base/core/java/android/accounts/AccountManager.java  
Line No: 408

```

public AuthenticatorDescription[] getAuthenticatorTypesAsUser(int userId) {

    try {
        return mService.getAuthenticatorTypes(userId);
    } catch (RemoteException e) {
        // will never happen
        throw new RuntimeException(e);
    }
}

```

File Path : frameworks/base/core/java/android/accounts/AccountManager.java  
Line No: 432

```

public Account[] getAccounts() {

    try {
        return mService.getAccounts(null, mContext.getOpPackageName());
    } catch (RemoteException e) {
        // won't ever happen
        throw new RuntimeException(e);
    }
}

```

File Path : frameworks/base/core/java/android/accounts/AccountManager.java  
Line No: 457

```

public Account[] getAccountsAsUser(int userId) {

    try {
        return mService.getAccountsAsUser(null, userId,
mContext.getOpPackageName());
    } catch (RemoteException e) {
        // won't ever happen
        throw new RuntimeException(e);
    }
}

```

File Path : frameworks/base/core/java/android/accounts/AccountManager.java  
Line No: 474

```

public Account[] getAccountsForPackage(String packageName, int uid) {

```

```

        try {
            return mService.getAccountsForPackage(packageName, uid,
mContext.getOpPackageName());
        } catch (RemoteException re) {
            // won't ever happen
            throw new RuntimeException(re);
        }
    }
}

```

File Path : frameworks/base/core/java/android/accounts/AccountManager.java

Line No: 492

```

public Account[] getAccountsByTypeForPackage(String type, String packageName) {

    try {
        return mService.getAccountsByTypeForPackage(type, packageName,
mContext.getOpPackageName());
    } catch (RemoteException re) {
        // won't ever happen
        throw new RuntimeException(re);
    }
}

```

File Path : frameworks/base/core/java/android/accounts/AccountManager.java

Line No: 530

```

public Account[] getAccountsByTypeAsUser(String type, UserHandle userHandle) {

    try {
        return mService.getAccountsAsUser(type, userHandle.getIdentifier(),
mContext.getOpPackageName());
    } catch (RemoteException e) {
        // won't ever happen
        throw new RuntimeException(e);
    }
}

```

File Path : frameworks/base/core/java/android/accounts/AccountManager.java

Line No: 550

```

public void updateAppPermission(Account account, String authTokenType, int uid, boolean value) {

    try {
        mService.updateAppPermission(account, authTokenType, uid, value);
    } catch (RemoteException e) {
        // won't ever happen
        throw new RuntimeException(e);
    }
}

```

File Path : frameworks/base/core/java/android/accounts/AccountManager.java  
Line No: 715

```
public boolean addAccountExplicitly(Account account, String password, Bundle userdata) {  
    if (account == null) throw new IllegalArgumentException("account is null");  
    try {  
        return mService.addAccountExplicitly(account, password, userdata);  
    } catch (RemoteException e) {  
        // Can happen if there was a SecurityException was thrown.  
        throw new RuntimeException(e);  
    }  
}
```

File Path : frameworks/base/core/java/android/accounts/AccountManager.java  
Line No: 740

```
public boolean notifyAccountAuthenticated(Account account) {  
    if (account == null)  
        throw new IllegalArgumentException("account is null");  
    try {  
        return mService.accountAuthenticated(account);  
    } catch (RemoteException e) {  
        throw new RuntimeException(e);  
    }  
}
```

File Path : frameworks/base/core/java/android/accounts/AccountManager.java  
Line No: 809

```
public String getPreviousName(final Account account) {  
    if (account == null) throw new IllegalArgumentException("account is null");  
    try {  
        return mService.getPreviousName(account);  
    } catch (RemoteException e) {  
        // will never happen  
        throw new RuntimeException(e);  
    }  
}
```

File Path : frameworks/base/core/java/android/accounts/AccountManager.java  
Line No: 980

```
public boolean removeAccountExplicitly(Account account) {  
    if (account == null) throw new IllegalArgumentException("account is null");  
    try {  
        return mService.removeAccountExplicitly(account);  
    } catch (RemoteException e) {  
        // May happen if the caller doesn't match the signature of the  
        authenticator.  
    }  
}
```

```
        throw new RuntimeException(e);
    }
}
```

File Path : frameworks/base/core/java/android/accounts/AccountManager.java  
Line No: 1008

```
public void invalidateAuthToken(final String accountType, final String authToken) {
    if (accountType == null) throw new IllegalArgumentException("accountType is
null");
    try {
        if (authToken != null) {
            mService.invalidateAuthToken(accountType, authToken);
        }
    } catch (RemoteException e) {
        // won't ever happen
        throw new RuntimeException(e);
    }
}
```

File Path : frameworks/base/core/java/android/accounts/AccountManager.java  
Line No: 1040

```
public String peekAuthToken(final Account account, final String authTokenType) {
    if (account == null) throw new IllegalArgumentException("account is null");
    if (authTokenType == null) throw new
IllegalArgumentException("authTokenType is null");
    try {
        return mService.peekAuthToken(account, authTokenType);
    } catch (RemoteException e) {
        // won't ever happen
        throw new RuntimeException(e);
    }
}
```

File Path : frameworks/base/core/java/android/accounts/AccountManager.java  
Line No: 1070

```
public void setPassword(final Account account, final String password) {
    if (account == null) throw new IllegalArgumentException("account is null");
    try {
        mService.setPassword(account, password);
    } catch (RemoteException e) {
        // won't ever happen
    }
}
```



```

        throw new RuntimeException(e);
    }
}

```

File Path : frameworks/base/core/java/android/accounts/AccountManager.java  
Line No: 1099

```

public void clearPassword(final Account account) {

    if (account == null) throw new IllegalArgumentException("account is null");
    try {
        mService.clearPassword(account);
    } catch (RemoteException e) {
        // won't ever happen
        throw new RuntimeException(e);
    }
}

```

File Path : frameworks/base/core/java/android/accounts/AccountManager.java  
Line No: 1128

```

public void setUserData(final Account account, final String key, final String value) {

    if (account == null) throw new IllegalArgumentException("account is null");
    if (key == null) throw new IllegalArgumentException("key is null");
    try {
        mService.setUserData(account, key, value);
    } catch (RemoteException e) {
        // Will happen if there is not signature match.
        throw new RuntimeException(e);
    }
}

```

File Path : frameworks/base/core/java/android/accounts/AccountManager.java  
Line No: 1158

```

public void setAuthToken(Account account, final String authTokenType, final String authToken) {

    if (account == null) throw new IllegalArgumentException("account is null");
    if (authTokenType == null) throw new
IllegalArgumentException("authTokenType is null");
    try {
        mService.setAuthToken(account, authTokenType, authToken);
    } catch (RemoteException e) {
        // won't ever happen
    }
}

```

```
        throw new RuntimeException(e);
    }
}
```

File Path : frameworks/base/core/java/android/accounts/AccountManager.java

Line No: 1556

```
public boolean addSharedAccount(final Account account, UserHandle user) {
    try {
        boolean val = mService.addSharedAccountAsUser(account,
user.getIdentifier());
        return val;
    } catch (RemoteException re) {
        // won't ever happen
        throw new RuntimeException(re);
    }
}
```

File Path : frameworks/base/core/java/android/accounts/AccountManager.java

Line No: 1607

```
public boolean removeSharedAccount(final Account account, UserHandle user) {
    try {
        boolean val = mService.removeSharedAccountAsUser(account,
user.getIdentifier());
        return val;
    } catch (RemoteException re) {
        // won't ever happen
        throw new RuntimeException(re);
    }
}
```

File Path : frameworks/base/core/java/android/accounts/AccountManager.java

Line No: 1621

```
public Account[] getSharedAccounts(UserHandle user) {
    try {
        return mService.getSharedAccountsAsUser(user.getIdentifier());
    } catch (RemoteException re) {
        // won't ever happen
        throw new RuntimeException(re);
    }
}
```

File Path : frameworks/base/core/java/android/accounts/AccountManager.java  
Line No: 1607

```
else if (cause instanceof RuntimeException) {  
    throw (RuntimeException) cause;  
}
```

File Path : frameworks/base/core/java/android/accounts/AccountManager.java  
Line No: 1621

```
else if (cause instanceof RuntimeException) {  
    throw (RuntimeException) cause;  
}
```