Security Services - Developer

> ⓘ **Note**
>
> This is a bit outdated (copied from this archive). So it's more a complement to OFBiz Security Permissions which is the reference and should be read first.

Overview: The OFBiz Security services control access to various parts of the application. The security services are built upon user login IDs, security permissions, security groups, and roles.

Be sure to read OFBiz Security Permissions and Security Administration before proceeding any further.

Security Services are not the same as Login Services - although the two are related.

OFBiz Security Services are divided among a handful of Java classes and related entities. The service that checks a user's permissions is extended from the abstract class org.ofbiz.security.Security. A number of other classes are used to create, modify, and delete user permissions - they can be found in the org.ofbiz.securityext package.

OFBizSecurity: This is the default security service. It handles checking a user's permissions and it is used throughout the application. It can be replaced with a custom security service - see below.

The securityext Component: This package of classes provide services for creating, modifying, and deleting user permissions. All security information is stored in the database.

Implementing A Custom Security Service: The default security service in OFBiz can be easily replaced with a custom service. Basically, it involves 2 steps:

1. Write a custom security service class
2. Create a service provider configuration file

You can write your custom security service class by extending it from org.ofbiz.security.Security or from org.ofbiz.security.OFBizSecurity. Extending from abstract class - advantage is you have the whole service written the way you want, disadvantage is increased development time. Extending from OFBizSecurity - advantage is you have a working security service to build on, disadvantage is your modifications could break if someone changes the superclass. By the way, modifying OFBizSecurity itself is NOT a good idea.

The service engine uses the Java ServiceLoader API to locate and load your custom Security implementation. Follow the instructions in the JavaDocs to create your service provider configuration file.