

Interceptor and plug-in classes are the primary way to extend server-side behavior. Use these classes to add specialized authentication and other security features, intervene in the query and save pipelines, customize entity key generation, and substitute your alternatives to some of the DevForce default components.

The general approach, described in the "[extend](#)" topic, is to create a class that implements a DevForce interface or overrides a DevForce class, then deploy the containing assembly to the server. DevForce uses [MEF](#) to find your class and includes it in its processes.

Some examples

Concern	Solution
EntityKey gen	Implement IIdGenerator to customize EntityKey generation .
Exceptions	Override EntityServerErrorInterceptor to block or change exceptions returned to the client.
Login	Implement IEntityLoginManager to customize login and logout .
Logging	Implement ITraceLogger to customize DevForce activity logging.
Queries	Override EntityServerQueryInterceptor to authorize, filter, modify, and otherwise alter the pipeline that processes client queries .
Saves	Override EntityServerSaveInterceptor to authorize, audit, modify entities to be saved, and otherwise alter the save pipeline .
Save POCOs	Override EntityServerPocoSaveAdapter to prescribe how POCO entities are saved .
Startup	Override EntityServiceApplication to add startup and shutdown behavior .
WCF Settings	Override ServiceHostEvents to modify the DevForce default WCF configuration.

Many more examples are [described here](#).