

# fSeries App Settings

fSeries has a number of settings that are across all accounts and refer to the server. These are held in the AppSettings.json file. In addition, account specific config values may be added, which are then accessible via a function.

## fSeries Section

The following relate to fSeries generally.

### AuthMethod

Indicates how authentication is dealt with by the system:

Forms: by entry of User Id and Password

Auth0: attaching the customer's Auth0 account for authentication

AzureAD: attaching the customer's Microsoft Azure Active Directory authentication

See separate sections below for settings specific to the authentication method.

### HoldPersonalData

Personal data (name, email, telephone number) are only held in the fSeries database if this setting is "true".

Note that email address may be held as the unique id for users for some authentication methods.

### ScriptVersion

The value is attached to all fSeries scripts (e.g. JavaScript and CSS files) to prevent browser caching when a new version is released. By default, it is taken from the WebSite dll file version but may be set here to temporarily override the default settings (for example, if a script file has been changed on site but the whole system has not yet been upgraded). In this situation make sure the setting is removed (not just set to null) in order for the default to be reactivated.

### fSeriesDatabase

Although similar to the DBContextConnection under the standard ConnectionStrings section, this is the connection string that is passed to data gatherers for them to have access back into the fSeries database (whereas the DBContextConnection is for the fSeries application to connection to it).

### ApiKey, ApiIssuer, ApiAudience

These settings are used to generate JWT Tokens for security.

### SingleAccountServer

fSeries can operate as a single account system or with multiple accounts. Set this to "true" to indicate that it is a single account. Some options are changed automatically for convenience between single and multiple account systems.

If this is set for a single account there are further settings specific to this type of system, in the SingleAccountSettings subsection:

AccountId – the id of the single account (e.g. Live)

CacheConfig – if “true” the account settings are cached for all users. This makes the system more efficient but is not possible for multiple accounts.

### HelpSite

The location of the online help site, linked to by the help icon (?) throughout the system.

### PreRegistrationOnly

(Not relevant for Forms authentication method)

If set to true new users must be pre-registered by an account administrator. Otherwise, new users may request access to the system by registering with Auth0. It is recommended to use the “true” option.

### AutoAuthorise

If set to true, a user is automatically considered to be authorised by virtue of their passing through initial security. The method used depends on the AuthMethod:

#### Forms

If set to true, and the ‘Authentication DSD (Auto)’ is set (to check users when auto-authorising) a user is assumed to be authorised if the DSD returns a valid single record. This means that user may enter the system without administrator intervention, so only use this option if you are fully aware of the consequences and the security DSD is properly set up.

#### Auth0

If set to true, the user is deemed to be authorised by virtue of the fact that Auth0 passed them through to fSeries. This is useful if your users are managed through Auth0 and you don’t want to add them manually in fSeries. Only use this option if you are sure that your Auth0 setup is managing users properly.

## Forms Section

If the AuthMethod is set to “Forms”, authentication, by default, is by User Id and Password, but other options are also available within this method.

### AutoAuthentication

If set to “true” the system will attempt to authenticate the user by their Active Directory identity. If successful no further entries are requested. If not, the user will go through the User Id and Password route.

### Two Factor Authentication

When the user has logged in with User Id and Password the system may be set to request a further entry of an authenticator code from an app such as Google Authenticator or Microsoft Authenticator.

#### TwoFactor

Set to one of the following:

- off – two factor authentication is not available
- on – two factor authentication is mandatory for all users
- optional – two factor authenticator is at the individual user’s option

If this option is activated, the system will take care of inviting users to register.

TwoFactorIssuer

The identity of the system which will be shown to the user in their authenticator app.

TwoFactorKey

This forms part of the encryption key that is used to validate the authenticator app code.

## Auth0 Section

If the AuthMethod is set to "Auth0", the following settings in the Auth0 section apply.

Domain, ClientId, ClientSecret

These settings will be provided for you in your Auth0 account and enable fSeries to connect to Auth0 for authentication.

ManagementId and ManagementSecret

These settings relate to the optional Auth0 Machine to Machine application. If HoldPersonalData is not set to true, setting these values will permit the user claims and user search to access user information from Auth0.

## AzureAD Section

If the AuthMethod is set to "AzureAD", the following settings in the AzureAD section apply.

Instance, Domain, ClientId, TenantId, ClientSecret, CallbackPath

These settings will be provided for you in your Azure AD account and enable fSeries to connect to Azure AD for authentication.

## fSeries Account Config

The fSeriesAccount:[AccountId] section contains custom settings specific to the account which may be accessed wherever a function may be entered. They may also be used to retrieve the connection string in a database connection source.

A file, fSeriesAccount.json (at the same location as appSettings.json) is available to hold these values.

For example:

```
{
  "fSeriesAccount": {
    "Live" : {
      "APIVersion": "1.0"
    }
  }
}
```

In this case the function =AccountConfig(APIVersion) will return the specified version number, to include in the resource required for an API data group.

When setting up a database connection, select the "Account Config" option for specifying the connection string and then enter the key (e.g. "DBConnection"). The config value will be retrieved at

run time. Note: this makes the value usable and therefore visible in DSDs and Menus. Directly entered connection strings are visible only when editing the source and are encrypted when saved in the database.