



AppAPI
Backup & Restore

GRIP ON SOL

2024-04-16

Contents

1	Introduction	3
2	Maintenance Mode.....	3
3	Configure AppAPI (QGrip-Admin)	3
3.1	Create AppAPI Login	3
3.2	Authorise AppAPI Login	4
3.2.1	Add/Edit Authorisation row	5
3.2.2	Status: IsHacked	5
3.3	Share AppAPI Login (with Teams/Projects).....	6
3.4	Trouble Shoot AppAPI Logins	6
4	Using AppAPI (Teams/Projects)	6
4.1	Connect Info	6
4.2	Connectivity (Firewalls)	6
4.3	Execute: [AppApi].[UspDatabaseAction].....	7
4.4	Parameters: [AppApi].[UspDatabaseAction].....	7
4.5	Examples: Execute [AppApi].[UspDatabaseAction].....	7
4.6	Return values: [AppApi].[UspDatabaseAction]	8

1 Introduction

With the AppAPI Backup & Restore Plugin, database backup and restores can be initiated without using the QGrip-UI. The normal QGrip backup and restore procedures will still be used.

This plugin can be used by teams/projects when backup of a database is needed before an automated process, like installing software in a database. If the installation fails, the backup can be restored from the same automated process.

The restore of the database is restricted to the last database backup made by the AppAPI.

The backups made using the AppAPI will always be of type BASELINE, a CopyOnly backup of the database that will not interfere with the normal backup chain.

2 Maintenance Mode

To prevent that QGrip jobs are running during maintenance (patching, reboots, etc.), the administrators can put components managed by QGrip, in Maintenance Mode for a period of time. The Maintenance Mode can be scheduled or on the fly. It is important that teams using the AppAPI are informed of planned and/or spontaneous Maintenance Mode of the components they use. The AppAPI process will check for planned Maintenance Mode but that is unfortunately not possible for non-planned.

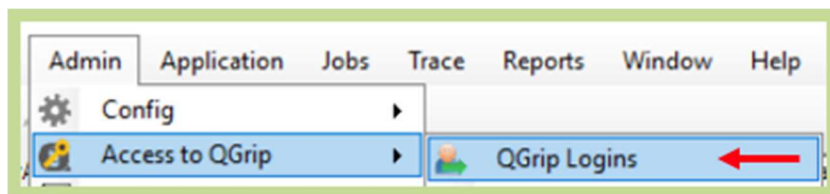
In the AppAPI Stored Procedure, there is a parameter, @RestoreWithinMinutes, that can be used to check if a Maintenance Mode window will start between backup and a possible database Restore.

3 Configure AppAPI (QGrip-Admin)

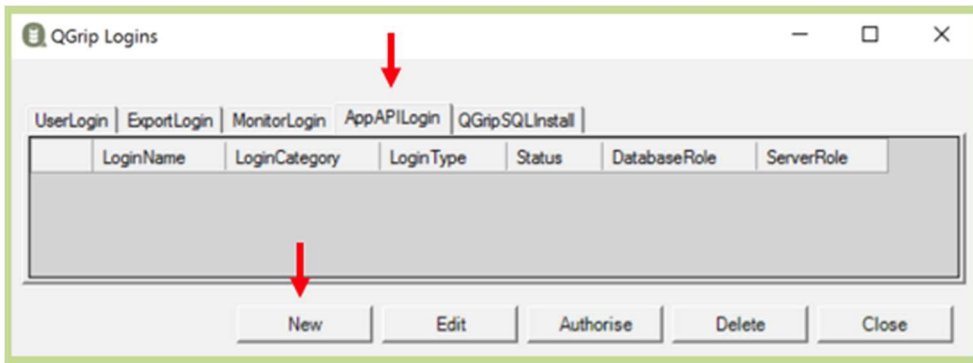
Configuring AppAPI is done by the QGrip-Admin and straight forward:

1. Create AppAPI Login
2. Authorise the AppAPI Login
3. Share Login credentials with team/project

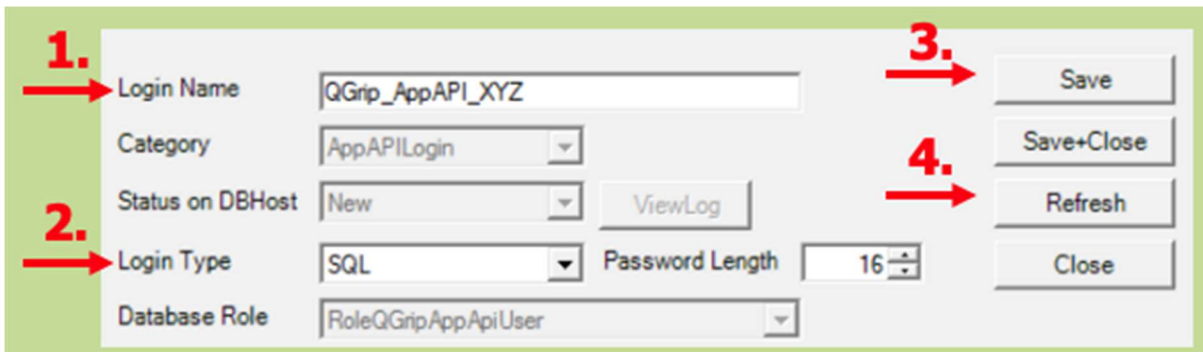
3.1 Create AppAPI Login



An AppAPI Login is a QGrip Login and must be created using the QGrip-UI. If the AppAPI Login is an AD Account, it must already exist on the Active Directory domain.

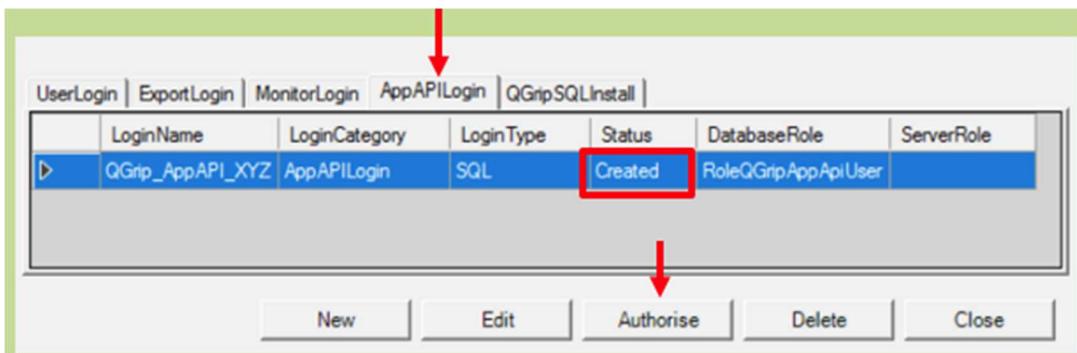


Select the tab AppAPI, press new.



1. Enter the name of the new login.
2. Choose Login type (SQL/ADUser).
3. Press Save, QGrip will now create the login on the QGrip DB Host.
4. Press Refresh until Status on DBHost is Created.
5. Close the window.

3.2 Authorise AppAPI Login



When the AppAPI Login has status [Created], select the row and press [Authorise]. If the status of the login is not yet created, you can refresh the Tab page content by selecting different Tabs.

The Authorise window shows the Authorisation that has been given to the AppAPI Login.

Status	Application	Environment	ADDomain	Authorisation	BackupShare	MaxKeepBackup(Days)
OK	MSY-Messy	Test	GOS-A	Backup&Restore (BR)	CopyOnly	2
OK	MSY-Messy	Acceptance	GOS-A	Backup Only (BO)	CopyOnly	3

To add Authorisation, press [Add].

Status	Application	Environment	ADDomain	Authorisation	BackupShare	MaxKeepBackup(Days)
OK	MSY-Messy	Test	GOS-A	Backup&Restore (BR)	CopyOnly	2
OK	MSY-Messy	Acceptance	GOS-A	Backup Only (BO)	CopyOnly	3

To Edit or Delete Authorisation, select the row and press [Edit] or [Delete].

3.2.1 Add/Edit Authorisation row

Login Name: QGrip_AppAPI_XYZ
 Application: MSY-Messy
 Environment: Acceptance
 Authorised For: Backup Only (BO) ← **1.**
 Domain: GOS-A
 Backup Share: CopyOnly ← **2.**
 Max Keep Backup: 3 (Days) ← **3.**

Choose the setting for the authorisation row. Note that you will need to add one authorisation row for each Application/Environment/Domain combination.

1. Authorised For: 'Backup Only ' or 'Backup&Restore'
2. Backup Share: Must be of type 'CopyOnly' or 'Baseline' (Other shares will not be shown).
3. Max Keep Backup: What is the max number of days that can be used in the parameter @KeepBackupDays when requesting a backup.

3.2.2 Status: IsHacked

If somebody has been tampering with the data in the table containing the Authorisation, the Status of the Authorisation row and the AppAPI Login will be set to 'IsHacked'. A backup and/or restore attempt will then fail with an 'Authorisation failure' message.

Login Name	QGrip_AppAPI_XYZ						
Status	IsHacked						
Authorised For							
Status	Application	Environment	ADDomain	Authorisation	BackupShare	MaxKeepBackup(Days)	
IsHacked	MSY-Messy	Test	GOS-A	Backup&Restore (BF)	CopyOnly	2	
OK	MSY-Messy	Acceptance	GOS-A	Backup Only (BO)	CopyOnly	3	

You can only fix this problem by Deleting the row and add it again.

3.3 Share AppAPI Login (with Teams/Projects)

The AppAPI Login is considered a QGrip Login and if it is a SQL Server Login, the password can be found at application QGrip in the Password Safe and or Connect Info.

3.4 Trouble Shoot AppAPI Logins

Every time the [AppApi].[UspDatabaseAction] stored procedure is called, the used parameters are added to a table in the QGrip Database. This table cannot (yet) be seen in the QGrip-UI but you can query it via SSMS:

- [AppApi].[DatabaseActionCall]

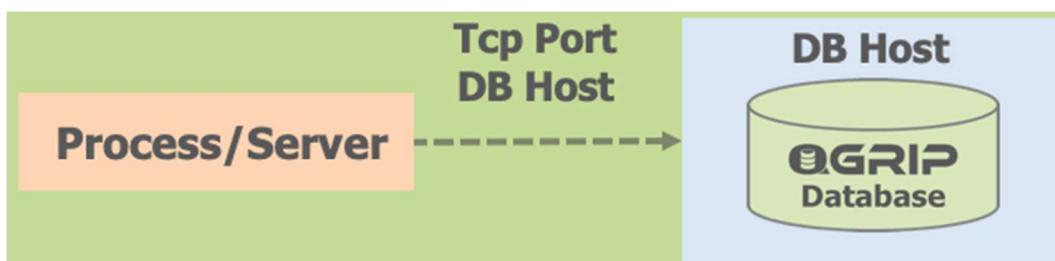
4 Using AppAPI (Teams/Projects)

4.1 Connect Info

Your QGrip Administrators must provide you with the information needed to setup the connection between your process and the QGrip database:

Parameter	Value	Description
QGrip DB Host Name		Server or Listener name
QGrip DB Host IP Address		If the DB Host Name cannot be resolved, it is possible that you need to connect to the IP Address instead of the name.
QGrip DB Host Port		Regularly 1433
QGrip Database Name		
AppAPI Login		
AppAPI Login Password		When the login is a SQL Server Login

4.2 Connectivity (Firewalls)



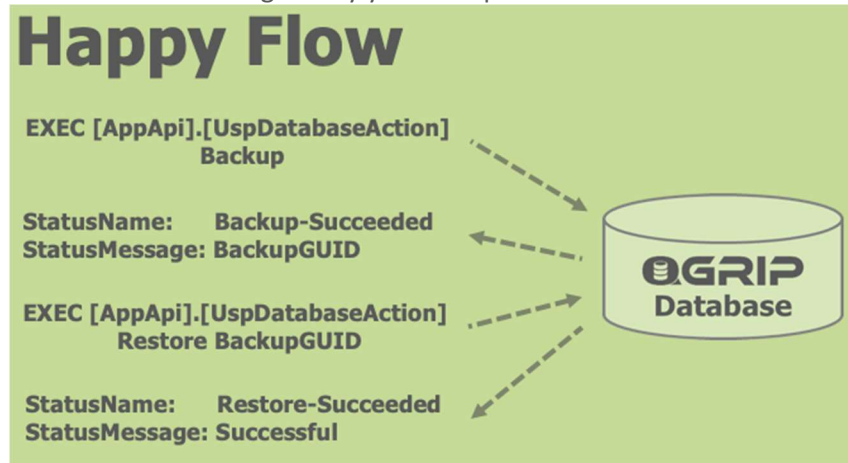
Before you can start using the AppAPI, it is likely that firewall changes are needed to enable connectivity between your source and the QGrip DB Host (Listener or Server).

4.3 Execute: [AppApi].[UspDatabaseAction]

Both Backup & Restore is accomplished by executing the stored procedure

- [AppApi].[UspDatabaseAction]

on the QGrip Database with the correct parameters. If you are allowed to perform Restore depend on the authorisation given by your QGrip Administrator.



4.4 Parameters: [AppApi].[UspDatabaseAction]

When the Stored Procedure is called, the correct parameters of the correct type must be supplied:

Parameter	Data Type	Mandatory For	Remark
@DBHostName	NVarchar(255)	Backup,Restore	Server or Listener name
@DBHostPort	SmallInt	Optional	Default = 1433
@DatabaseName	NVarchar(255)	Backup,Restore	
@DatabaseAction	NVarchar(255)	Backup,Restore	'Backup' or 'Restore'
@BackupGUID	NVarchar(255)	Restore	
@KeepBackupDays	Integer	Backup	
@RestoreWithinMinutes	Integer	Backup (Optional)	Default = 0 (*)

(*) @RestoreWithinMinutes

This should be an estimate of the time needed between the backup and a possible restore of the database. This parameter will be used to check for future Maintenance Mode windows. If there is a Maintenance Mode window planned in this period, the backup will not be executed but a message will be returned indicating when the Maintenance Mode window will be over.

4.5 Examples: Execute [AppApi].[UspDatabaseAction]

```
EXEC [AppApi].[UspDatabaseAction]
    @DBHostName          = 'ADEVSQL16\TST'
    , @DBHostPort        = 1440
    , @DatabaseName      = 'BIGDB_T_MAIN'
    , @DatabaseAction    = 'Backup'
```

```
, @KeepBackupDays = 1
```

Just backup the database and keep the backup file for 1 day.

```
EXEC [AppApi].[UspDatabaseAction]
    @DBHostName = 'ADEVSQL16\TST'
, @DBHostPort = 1440
, @DatabaseName = 'BIGDB_T_MAIN'
, @DatabaseAction = 'Backup'
, @KeepBackupDays = 2
, @RestoreWithinMinutes = 10
```

Backup the database and keep the backup file for 2 days. If there is a planned Maintenance Mode within 10 minutes, the backup job will not be executed because a database restore cannot be started within the given time frame.

```
EXEC [AppApi].[UspDatabaseAction]
    @DBHostName = 'ADEVSQL16\TST'
, @DBHostPort = 1440
, @DatabaseName = 'BIGDB_T_MAIN'
, @DatabaseAction = 'Restore'
, @BackupGUID = '4C366F3D-A19B-4731-ACFB-1B228133D949'
```

Restore the backup with BackupGUID. The BackupGUID was received when backing up the database.

4.6 Return values: [AppApi].[UspDatabaseAction]

When the Stored Procedure is called, it will always return 1 row with the following columns:

Column Name	Data Type
Status	Integer
StatusName	NVarchar(255)
StatusMessage	NVarchar(255)

The row will be returned when the action has been completed. If a large database is being backed-up or restored, it can take some time..

Successful Backup:

Status	StatusName	StatusMessage
1	Backup-Succeeded	BackupGUID

The BackupGUID will look something like: A2259015-D654-4F71-B2E0-453BA94AD4CA

This value must be stored in a variable as it will be needed to initiate a restore the database backup.

Successful Restore:

Status	StatusName	StatusMessage
1	Restore-Succeeded	Successful

Maintenance Mode

Status	StatusName	StatusMessage
-2	MaintenanceMode	MaintenanceMode Until: 2023-04-27 14:50:18

If the action is prevented by a (Future) Maintenance Mode window, the StatusMessage will contain an indication when the Maintenance Mode window is over.

If an action fails, one of the following rows will be returned:

Status	StatusName	StatusMessage
-1	Failed	Arguments invalid or empty
-1	Failed	Authorisation failure
-1	Failed	Component failure

Component failure will only happen if a component that is needed, is not available.