AppMill

CONFIGURATION FILES

Table of Contents

1. Introduction

Configuration files are used to specify the settings and parameters required by software applications or systems to operate effectively. These files contain various directives or options that determine the behavior, functionality, and customization of the software or system.

⚠ All basic configuration settings are made automatically during the service installation via the installer.

It is not recommended to modify the configuration files manually without a specific need, as it may lead to incorrect application work.

2. Codiac.HTTPService.

Codiac.HTTPService.config is a configuration file for service settings in the XML format.

For example, the file can contain the following data:

```
<config>
  <service name="appmill.service" useHttps="false" balancer_node="false" />
  <logs store="false"/>
  <bind address="localhost" port="34567"/>
  <interprocess port="30002"/>
  <balancer is_balancer = "true" heartbeat_timeout = "300" server_choose =
"next_in_line">
    <service address="192.168.100.50" port="34561" useHttps = "false" />
    <service address="192.168.100.50" port="34562" useHttps = "false" />
  </balancer>
  <devicecryptor deviceaeskey="" />
  <plugin_sync_servers>
  <service address="192.168.100.50" port="34561" useHttps = "false" public_key=""
product_id="" />
  </plugin_sync_servers>
  <passthru is_passthru="" acceptXML="" />
</config>
```

Note that the root element must be **config**, and the sub elements should be categorized based on the areas where their settings can be applied.

The list of the configuration file sub elements:

- **service** - service configuration.
- **logs** - log configuration.
- **bind** - the end point that configures port and address where the service handles the incoming requests.
- **interprocess** - the sub element that configures port to be opened during communication between subprocesses.
- **balancer** - the sub element that configures the balancer cluster.
- **devicecryptor** - the sub element that encrypts or decrypts data stored on a device to ensure that data remains secure and inaccessible to unauthorized users. At the moment, this sub element will only be used if it is specified in the **CodiacSDK.Core.config** configuration file.
- **plugin_sync_servers** - the sub element that configures the plugin synchronization process.
- **passthru** - the sub element that provides the possibility to bypass the Alias Framework security that is defined for a specific field or alias. At the moment, this sub element will only be used if it is specified in the **CodiacSDK.Core.config** configuration file.
- **threads** - the sub element that configures threads pools. At the moment, this sub element will only be used if it is specified in the **CodiacSDK.zJobsScheduler.config** configuration file.

Note that if the **devicecryptor**, **passthru**, and **threads** sub elements' values are empty in the **Codiac.HTTPService.config** configuration file, such sub elements are not used by the service, and these features are disabled.

To enable these features:

- the **devicecryptor** and **passthru** sub elements' attributes should be specified in the **CodiacSDK.Core.config** configuration file.
- the **threads** sub element attributes should be specified in the **CodiacSDK.zJobsScheduler.config** configuration file.

The sub elements attributes of the **Codiac.HTTPService.config** configuration file are described below:

Service

The list of the **Service** attributes:

- **name** - the name that can be used at the service registration.
- **useHttps** - the attribute that defines the use of the HTTPS and HTTP protocols.
  In case the value is set to:
  - ***true*** - the HTTPS protocol will be used,
    or
  - **false** - the HTTP protocol will be used.
- **balancer_node** - the parameter that incorporates service into the balancer cluster.
  In case this attribute is set to:
  - ***true*** - the service will function as a balancer node.
    or
  - **false** - the service will not work as a balancer node.

Logs

The list of the **Logs** attributes:

- **store** - this attribute can enable or disable logs in the verbose mode.
  In case the attribute is set to:
  - ***true*** - the logs will be enabled in the verbose mode.
    or
  - ***false*** - the logs will be disabled in the verbose mode.
- **store_db** - this attribute can enable or disable the logs storing into the database.
  In case the attribute is set to:
  - ***true*** - the logs will be stored into the database.
    or
  - ***false*** - the logs will not be stored into the database.
- **plugins_logs** - this attribute can enable or disable logs in verbose mode for plugins.
  In case the attribute is set to:
  - ***true*** - the logs will be enabled in verbose mode for plugins.
    or
  - ***false*** - the logs will be disabled in verbose mode for plugins.

Bind

The list of the **Bind** attributes:

- **address** - IP or FQDN name.
- **port** - port where the service handles incoming requests.

Interprocess

The list of the **Interprocess** attributes:

- **port** - port that will be opened for the incoming requests from the sub processes initiated by service. These sub processes include those executing PHP, JS, Python, and C# scripts.

Balancer

The list of the **Balancer** attributes:

- **is_balancer** - this attribute can activate the server as balancer.
  In case the attribute is set to:
  - ***true*** - the service works as a balancer and redirects all requests to the balancer nodes.
    or
  - ***false*** - the balancer is disabled.
- **heartbeat_timeout** - the time interval in the seconds, when the balancer checks that all services from the list are in the available status.
- **server_choose** - algorithm that will be used at the request redirections to the end services. Possible values: **cpu_load, next_in_line, any_server_no_sessions**.

The services list, that is in the balancer cluster, should contain at least one **service** element with the following attributes to allow connection and communication with these services:

- **address** – IP or FQDN name of the services.
- **port** – port on which this service listens to incoming requests.
- **useHttps** – boolean value (true or false) that shows the type of the protocol for communication with this service. When the value is equal to:
  - ***true*** - HTTPS will be selected,
    or
  - ***false*** - HTTP will be selected.

Devicecryptor

The list of the **Devicecryptor** attributes:

- **deviceaeskey** - this key can encrypt or decrypt the username and password to access the device. Also, the key can be used in the Column Encryption functionality.

Plugin_sync_servers

The **Plugin_sync_servers** is a list of the services that will be included into the plugin synchronization.

Each service in the list has the following attributes:

- **address** - IP or FQDN name.
- **port** - port of a target service.
- **useHttps** - the attribute that defines the use of the HTTPS and HTTP protocols.
  In case the value is set to:
  - ***true*** - the HTTPS protocol will be used,
    or
  - **false** - the HTTP protocol will be used.
- **public_key** - a public part of an asymmetric key that is used to encrypt the files content during synchronization process.
- **product_id** - the target service product ID that can be included into the synchronization process.
  In case the product IDs are not identical, the communication will be terminated.

Passthru

The list of the **Passthru** attributes:

- **is_passthru** - this attribute activates the pass-through mode.
  In case the attribute is set to:
  - ***true*** - the pass-through mode is enabled.
    or
  - ***false*** - the pass-through mode is disabled.
- **acceptXML** - this attribute activates the support to accept requests in the XML form.
  In case the attribute is set to:
  - ***true*** - the support to accept requests in the XML form is enabled.
    or
  - ***false*** - the support to accept requests in the XML form is disabled.

Threads

The **Threads** sub element attribute is described below:

- **max_thread_count** - this attribute specifies how many jobs can be executed simultaneously.

3. JobsScheduler.config

JobsScheduler.config is a configuration file for the job scheduler tool settings in the XML format.

For example, the file can contain the following data:

```
<config>
    <scheduler service_address="10.0.2.116" service_port="34560" user="Admin"

password="3c9909afec25354d551dae21590bb26e38d53f2173b8d3dc3eee4c047e7ab1c1eb8b85103e3
    be7ba613b31bb5c9c36214dc9f14a42fd7a2fdb84856bca5c44c2"  max_run_time="30" />
</config>
```

The **JobsScheduler.config** file contains settings parameters only for the **Scheduler**.

The **Scheduler** sub element attributes are described below:

- **service_address** - the service address where request will be sent.
- **service_port** - the service port where request will be sent.
- **user** - a username that will be used at the authentication on the service side.
- **password** - a password that will be used at the authentication on the service side.
- **max_run_time** - the request execution timeout duration.

4. CodiacSDK.zJobsScheduler.config

CodiacSDK.zJobsScheduler.config is a configuration file for the job scheduler.

The **CodiacSDK.zJobsScheduler.config** file contains settings parameters only for **Threads**.

- **threads** - the sub element that configures threads pools.

For example, the file can contain the following data:

```
<config>
    <threads max_thread_count = "10" />
</config>
```

Threads

The **Threads** sub element attribute is described below:

- **max_thread_count** - this attribute specifies how many jobs can be executed simultaneously.

5. CodiacSDK.Core.config

CodiacSDK.Core.config is a configuration file for the fundamental settings or options that define how the software operates.

For example, the file can contain the following data:

```
<config>
  <datapool dbtype="Postgres" address="10.0.2.82" name="geo" port="5432"
user="postgres"
password="VOj4UTFu3I0XgyuKudZ8RQ==" useConnectionPool="true"
minimumActiveConnections="10"
maximumActiveConnections="50" encrypted_password="true"/>
  <logs store="false" plugins_logs="false" store_db="false"/>
  <security language="Language.en-US" encryption="false" hashpassword="both"/>
  <passthru is_passthru="false" acceptXML="true"/>
</config>
```

The list of the core sub elements:

- **datapool** - the sub element that contains the database connection details and some additional settings on how the AppMill service should manage the database connections.
- **logs** - log configuration.
- **security** - the sub element that contains a variety of security options.
- **passthru** - the sub element that provides the possibility to bypass the Alias Framework security that is defined for a specific field or alias.

Note that the security for the field or alias that is marked as not eligible for passthru (PassthruEnabled) can never be bypassed. For example, the security for the field or alias with the c_AliasTable.PassthruEnabled name will not be bypassed.

- **devicecryptor** - the sub element that encrypts or decrypts data stored on a device to ensure that data remains secure and inaccessible to unauthorized users.

The Core sub elements attributes are described below:

Datapool

The list of the **Datapool** attributes:

- **dbtype** - a database type.
  The following database types can be used:
  - Postgres
  - Oracle
  - MsSQL
  - MySQL
  - SQLite
- **address** - IP or FQDN name of the database server.
- **name** - a database name.
- **port** - the database server port that was opened for incoming connections.
- **user** - a user or a role that has enough rights to connect to the database and execute operations on it.
- **password** - a password of the database user or database role.

- **useConnectionPool** - this attribute indicates the use of the connection pool.
  In case the attribute is set to:
  - ***true*** - a pool of the connections will be created to allow the faster queries execution on the already opened connections.
    or
  - ***false*** - the use of the connection pool will be disabled.
- **minimumActiveConnections** - the minimal number of the connections to the database that will be created during the service starting.
- **maximumActiveConnections** - the maximal number of the connections to the database in the pool.
- **encrypted_password** - this attribute can be changed only by service.

⚠ Note that the **encrypted_password** attribute must not be changed. The form of the password is shown in the **password** attribute.

Logs

The list of the **Logs** attributes:

- **store** - this attribute can enable or disable logs in the verbose mode.
  In case the attribute is set to:
  - ***true*** - the logs will be enabled in the verbose mode.
    or
  - ***false*** - the logs will be disabled in the verbose mode.
- **store_db** - this attribute can enable or disable the logs storing into the database.
  In case the attribute is set to:
  - ***true*** - the logs will be stored into the database.
    or
  - ***false*** - the logs will not be stored into the database.
- **plugins_logs** - this attribute can enable or disable logs in verbose mode for plugins.
  In case the attribute is set to:
  - ***true*** - the logs will be enabled in verbose mode for plugins.
    or
  - ***false*** - the logs will be disabled in verbose mode for plugins.

Security

The list of the **Security** attribute:

- **encryption** - this attribute activates an additional encryption for the body data in the each request.
  In case the attribute is set to:
  - ***true*** - the data additional encryption will be enabled.
    or
  - ***false*** - the data additional encryption will be disabled.
- **language** - this attribute defines the default language code for returning responses from the service.
- **hashpassword** - this attribute specifies how the user password should be sent to the AppMill service for authentication, i.e., how client applications need to prepare the password. There are three options at the moment:

- **hash** - the raw password should be hashed and then sent to the AppMill service.
- **no_hash** - the raw password should be sent to the AppMill service.
- **both** - both hashed and raw passwords are acceptable.
- **client_verification** - this attribute provides the possibility to control and manage the level of access that client applications have to the entire service.
  In case the attribute is set to:
  - ***true*** - the client verification functionality will be enabled.
    or
  - ***false*** - the client verification functionality will be disabled.

For example, the attribute can contain the following data:

```
<security language="Language.en-US" encryption="false" hashpassword="both"
client_verification="true"/>
```

Passthru

The list of the **Passthru** attributes:

- **is_passthru** - this attribute activates the pass-through mode.
  In case the attribute is set to:
  - ***true*** - the pass-through mode is enabled.
    or
  - ***false*** - the pass-through mode is disabled.
- **acceptXML** - this attribute activates the support to accept requests in the XML form.
  In case the attribute is set to:
  - ***true*** - the support to accept requests in the XML form is enabled.
    or
  - ***false*** - the support to accept requests in the XML form is disabled.

Devicecryptor

The list of the **Devicecryptor** attributes:

- **deviceaeskey** - this key can encrypt or decrypt the username and password to access the device. Also, the key can be used in the Column Encryption functionality.