Πίνακας περιεχομένων

[Hit Services 2](#_Toc509837532)

[Παραμετροποίηση 4](#_Toc509837533)

[Database 4](#_Toc509837534)

[Configuration (HitServicesConfig) 4](#_Toc509837535)

[Configuration HitServicesApi 10](#_Toc509837536)

[Εγκατάσταση του Service HitServicesScheduler 11](#_Toc509837537)

[**Διαθέσιμα Services** 13](#_Toc509837538)

[**A.** **Generic Services** 13](#_Toc509837539)

[**RunScript (generic)** 13](#_Toc509837540)

[**Export Data (generic)** 14](#_Toc509837542)

[**Save To Table (generic)** 16](#_Toc509837543)

[**Read from csv (generic)** 17](#_Toc509837544)

[**B.** **Generic Protel Services** 18](#_Toc509837545)

[**Protel Availabilities** 18](#_Toc509837546)

[**Protel Rate Codes** 19](#_Toc509837547)

[**C.** **Custom Services for SmartIn** 21](#_Toc509837548)

[**ELock (SmartIn)** 21](#_Toc509837549)

[**LockInfo (SmartIn)** 21](#_Toc509837550)

[**D.** **Custom Services for Gorissia** 22](#_Toc509837551)

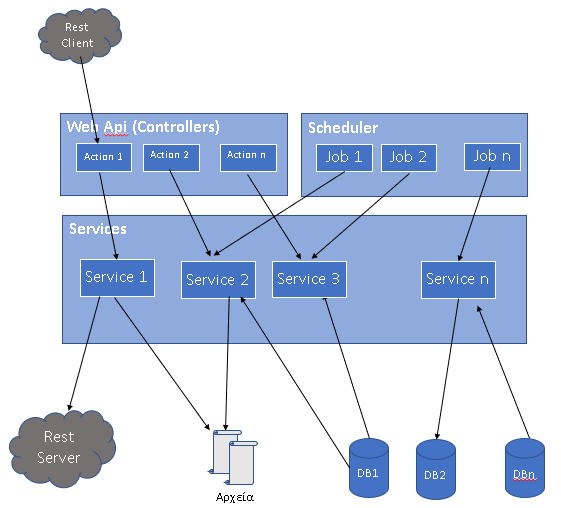
[**GetProfilesJob (Corissia)** 22](#_Toc509837552)

[**E.** **Custom Services for WaterPark** 23](#_Toc509837553)

[**OnlineRegistration (WaterPark)** 23](#_Toc509837554)

# Hit Services

Τα Hit Services είναι μία σειρά από services τα οποία μπορούν να εκτελεστούν με χρονοπρογραμματισμό μέσω jobs (Hit Scheduler) ή να κληθούν μέσω Rest κλήσεων σε Controller’s Actions. (Hit Web Api)



Μερικά Services είναι generic καθώς μπορούν να παραμετροποιηθούν ανάλογα την περίπτωση πχ εκτέλεση sql script, export δεδομένων σε αρχεία, μεταφορά δεδομένων. Άλλα Services έχουν δημιουργηθεί για να ικανοποιήσουν συγκεκριμένες ανάγκες πελατών.

Σκοπός των Hit Services είναι να ενσωματωθούν σε αυτά μικρά και μεσαία projects που αφορούν integration μεταξύ προϊόντων και εφαρμογών της HIT ή μεταξύ προϊόντων της HIT και άλλων εταιριών καθώς και generic εργασίες. Αποτελούνται από :

**HitServicesApi** : Web Api για την επικοινωνία μεταξύ των εφαρμογών που θέλουμε να κάνουμε integrate. Επίσης στο web api τρέχει και το dashboard του scheduler.

**HitServicesScheduler**: Windows service το οποίο είναι επιφορτισμένο να τρέχει scheduled Jobs. Στο dashboard φαίνεται η εξέλιξη των jobs.

**HitServices**: περιέχει τα services τα οποία εκτελούνται από τον Scheduler ή από το Api.

**HitServicesConfig**: εφαρμογή για το configuration των scheduled Jobs.

Ta παραπάνω βρίσκονται στο folder [\\sisifos\ftp\HitServices](file:///\\sisifos\ftp\HitServices)

# Παραμετροποίηση

## Database

O HitServicesScheduler απαιτεί μία **κενή** **database**, με όνομα **HitServices**, στην οποία αποθηκεύεται η εξέλιξη των jobs.

## Configuration (HitServicesConfig)

Στην εφαρμογή **HitServicesConfig.exe** μπορούν να πραγματοποιηθούν οι παρακάτω ενέργειες:

* Διαχείριση και χρονοπρογραμματισμός των scheduled jobs
* Άμεση Εκτέλεση job (Fire-And-forget)
* Διαχείριση settings για τα jobs (παραμετροποίηση των jobs).
* Διαχείριση sql scripts για όσα jobs εκτελούν sql scripts
* Διαχείριση χρηστών για το HitServicesApi
* Encrypt/Decrypt κειμένων.

Όλη η παραπάνω πληροφορία αποθηκεύεται κρυπτογραφημένη σε αρχεία.

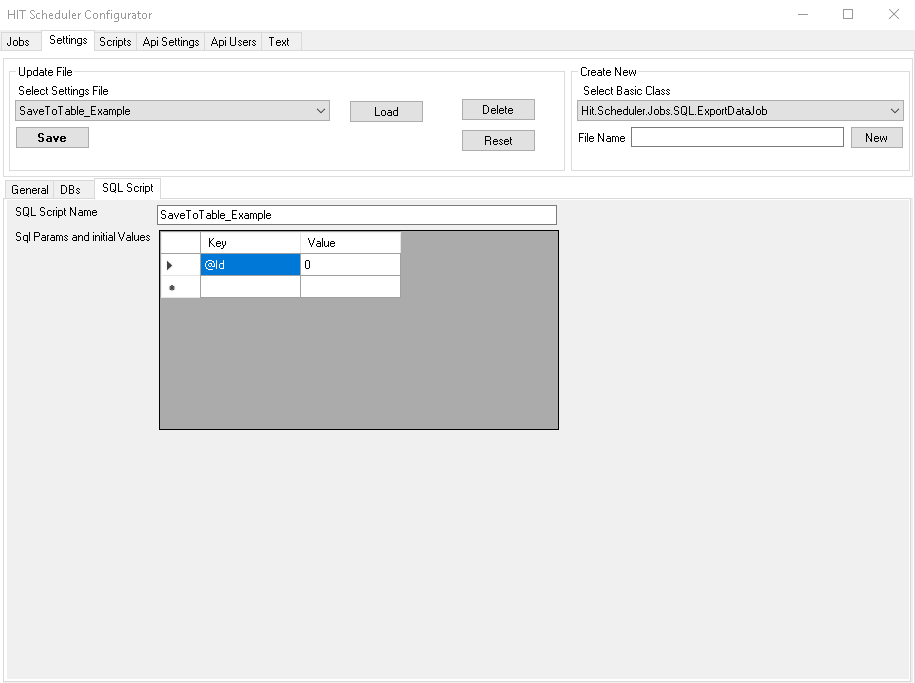
Η εφαρμογή HitServicesConfig βρίσκεται στο path \HitServices\bin\HitServicesConfig.exe

Η πρόσβαση στην εφαρμογή HitServicesConfig γίνεται μέσω δύο passwords:

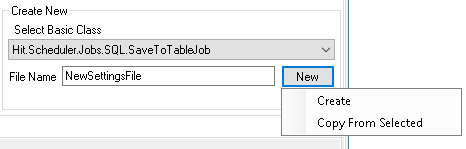
* !4502! : δίνει πλήρη δικαιώματα στον χρήστη
* hit12!@ : δίνει δικαιώματα στον χρήστη μόνο να τρέχει manually ένα job (Fire-and-forget) καθώς και να μεταβάλει το χρονοπρογραμματισμό των jobs.

**Settings**

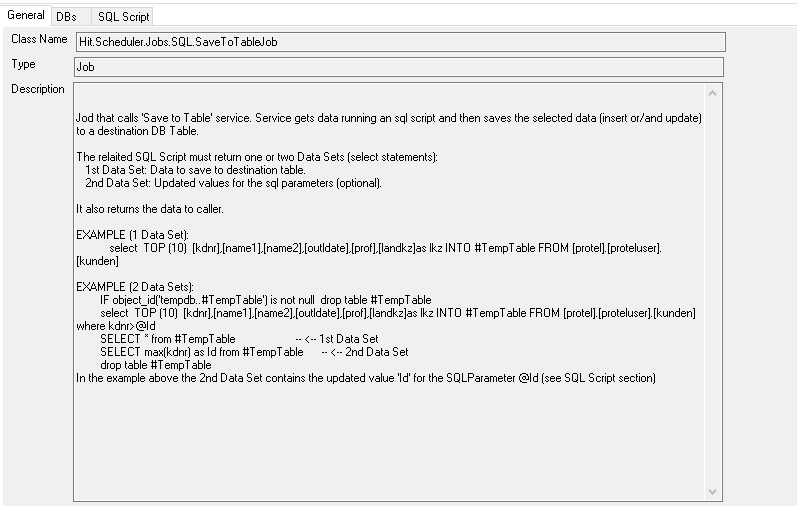
Στη καρτέλα των settings ο χρήστης μπορεί να διαχειριστεί Settings αρχεία, στα οποία ορίζεται η παραμετροποίηση (configuration) κάθε job ή controller (πχ: connection strings, sql script, files κτλ)

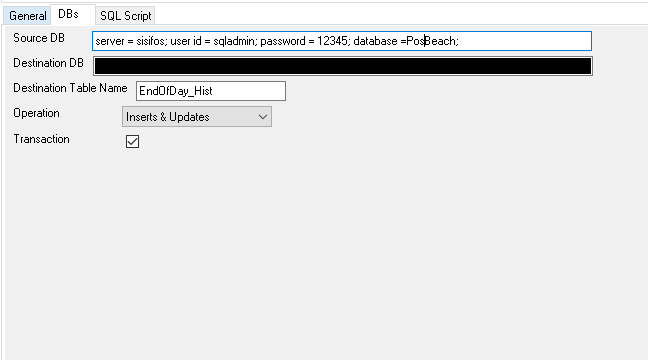


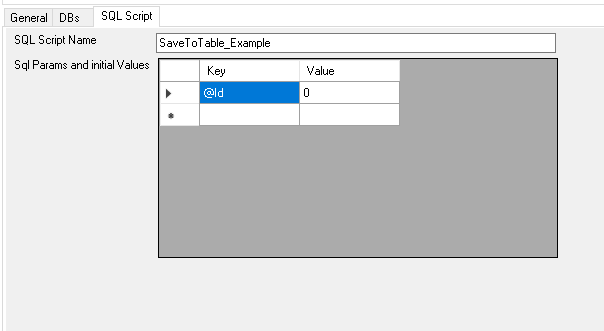
Για να δημιουργηθεί ένα **νέο settings file** , στη περιοχή ‘Create New’ ο χρήστης πρέπει να επιλέξει το Job ή τον Controller (Select Basic Class) για τον οποίο θέλει να δημιουργήσει το νέο settings file και να ορίσει ένα μοναδικό όνομα του settings file.



Μόλις ο χρήστης πατήσει New η εφαρμογή θα δημιουργήσει το νέο settings file. Ο χρήστης θα πρέπει να συμπληρώσει στις καρτέλες που θα εμφανιστούν το απαιτούμενο configuration για τη λειτουργία του Job ή Controller. Ανάλογα το επιλεγμένο Job ή Controller εμφανίζονται τα απαραίτητα πεδία για configuration τα οποία ο χρήστης θα πρέπει να συμπληρώσει :







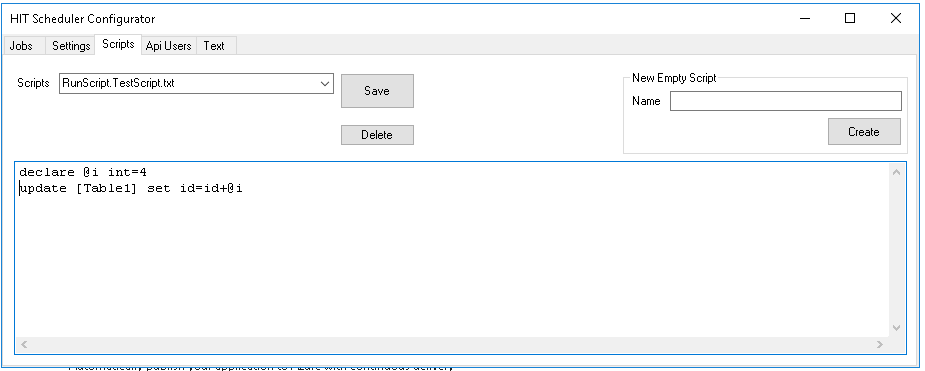
Στη συνέχεια ο χρήστης θα πρέπει να πατήσει Save για να αποθηκεύσει το configuration.

Αντί για δημιουργία νέου ο χρήστης μπορεί να αντιγράψει το configuration από υπάρχον settings file σε νέο.

Επίσης ο χρήστης μπορεί να μεταβάλει το configuration από υπάρχον settings file ή να διαγράψει υπάρχον settings file.

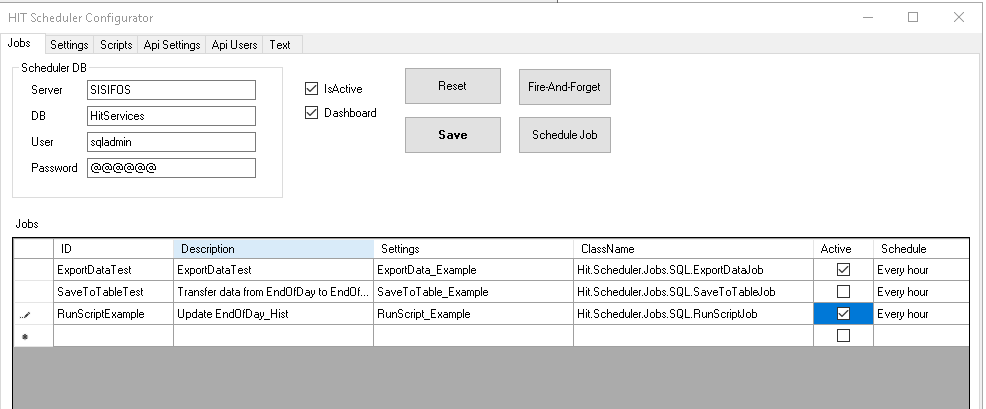
**Scripts**

Στη καρτέλα των Scripts ο χρήστης μπορεί να διαχειριστεί sql scripts αρχεία τα οποία στη συνέχεια ορίζονται σαν παράμετρος στα settings files (σε όσα απαιτούν sql scripts ):



**Διαχείριση και χρονοπρογραμματισμός των scheduled jobs**

Στην καρτέλα των Jobs ο χρήστης πρέπει να εισάγει τα στοιχεία σύνδεσης της κενής DB η οποία δημιουργήθηκε στο προηγούμενο βήμα (περιοχή *Scheduler DB*), να ενεργοποιήσει ή να απενεργοποιήσει τον scheduler (*IsActive*), και να ενεργοποιήσει ή να απενεργοποιήσει τον dashboard του scheduler (*Dashboard*).



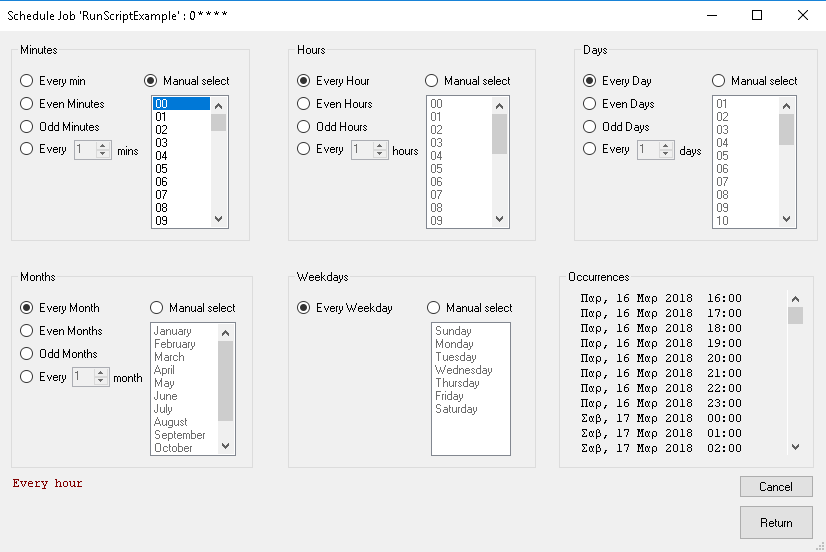
Επίσης ο χρήστης μπορεί να εισάγει νέο scheduled job, να μεταβάλει ή να διαγράψει (πλήκτρο Delete) υπάρχον scheduled job. Για κάθε scheduled job o χρήστης θα πρέπει να ορίσει τα παρακάτω :

* Μοναδικό **ID**
* Περιγραφή (**Description**)
* **Settings** αρχείο από λίστα με τα Settings αρχεία (Τα αρχεία Settings δημιουργούνται στην καρτέλα των settings)
* **IsActive**: για ενεργοποίηση ή απενεργοποίηση του scheduled job
* **Χρονοπρογραμματισμό του job**

Για την αποθήκευση οποιαδήποτε αλλαγής, δημιουργίας ή διαγραφής ο χρήστης πρέπει να πατήσει «**Save**».

**Χρονοπρογραμματισμός Job**

Ο χρήστης επιλέγει ένα scheduled job και πατά το κουμπί «Schedule Job». Στην οθόνη που εμφανίζεται ο χρήστης μπορεί να ορίσει τον χρονοπρογραμματισμό του job. Στο παρακάτω παράδειγμα το job θα τρέχει κάθε μια ώρα:



Στον χρήστη εμφανίζονται οι πρώτες χρονικές στιγμές στις οποίες το job θα εκτελεστεί (Occurrences) καθώς και λεκτική περιγραφή του χρονοπρογραμματισμού.

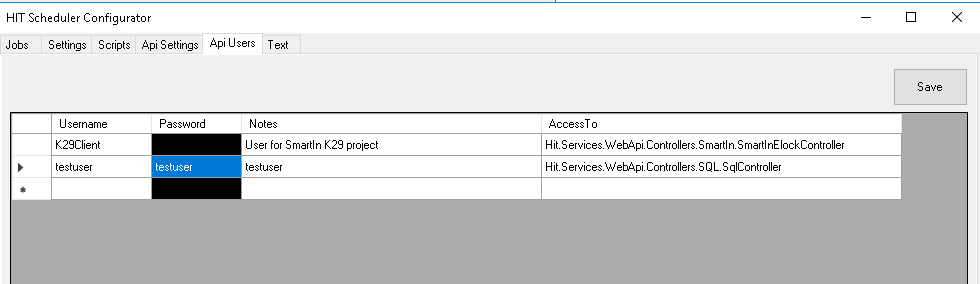
Ο χρήστης πατώντας “**Return”** μεταφέρει τον χρονοπρογραμματισμό που δημιούργησε στο grid των jobs στην αρχική οθόνη (για την αποθήκευσή του απαιτείται “Save” ).

**Fire-And-forget**

Για την άμεση εκτέλεση ενός job ο χρήστης επιλέγει το job που θέλει να εκτελέσει και στη συνέχεια πατάει «Fire-And-Forget». Το job θα ξεκινήσει αμέσως, και ο χρήστης θα πρέπει να συμβουλευτεί το dashboard ή τα log files για την επιτυχή εκτέλεσή του. Κατά τη διάρκεια εκτέλεσης του job ο χρήστης δε θα πρέπει να κλείσει την εφαρμογή.

**Api Users**

Στην καρτέλα των api users ο χρήστης ορίζει χρήστες για το web api interfase (**HitServicesApi**).

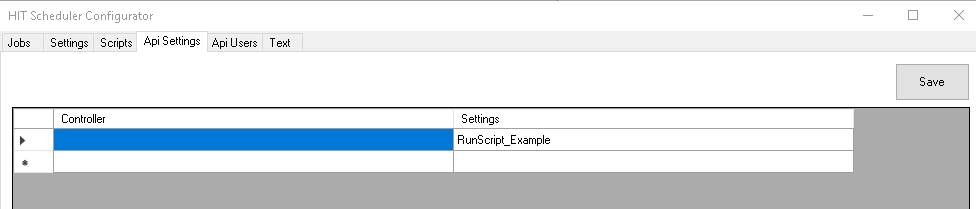


Για κάθε χρήστη θα πρέπει να οριστούν: username, password,και τον Controller στον οποίο ο χρήστης έχει πρόσβαση (AccessTo).

Καλό είναι σε κάθε εγκατάσταση να διαγράφονται οι χρήστες που δεν χρειάζονται για τη λειτουργία της εγκατάστασης.

**Api Settings**

Στην καρτέλα των api settings ο χρήστης ορίζει setting file για controllers του web api interfase (**HitServicesApi**).



Επιπλέων το configuration γίνεται στα παρακάτω αρχεία:

**HitServices\_Scheduler.exe.config**

Σε αυτό το αρχείο ορίζονται:

* Θέση των log files :

<log4net>

<appender name="RollingLogFileAppender" type="log4net.Appender.RollingFileAppender">

<file value="**C:\logs\HitScheduler.log**" />

## Configuration HitServicesApi

**Web.config**

Σε αυτό το αρχείο ορίζεται η γενική παραμετροποίηση του api:

* Θέση των log files :

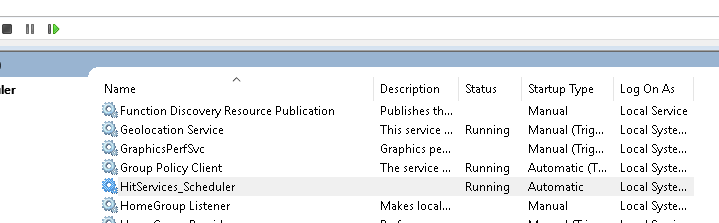
<log4net>

<appender name="RollingLogFileAppender" type="log4net.Appender.RollingFileAppender">

<file value="**C:\logs\ HitServices.log**" />

## Εγκατάσταση του Service HitServicesScheduler

Μετά την εγκατάσταση και παραμετροποίηση των εφαρμογών HitServicesScheduler και HitServicesApi, εκτελείται το **HitServices\_Scheduler.exe** και στη συνέχεια επιλέγουμε “**I**” ώστε να γίνει η εγκατάσταση του service. Στη συνέχεια στα services θα πρέπει να φαίνεται ότι το service είναι σε κατάσταση Running:



Επιπλέον στο log file θα πρέπει να φαίνονται ότι τα jobs είναι ενεργά και δεν υπάρχει error:

07/02/2018 15:59:00,444 [43] [INFO ] [Hit.Services.Scheduler.Core.HangfireBootstrapper] [Start] - Initialising Scheduler...

07/02/2018 15:59:00,460 [32] [INFO ] [Hit.Services.Scheduler.Core.HangfireBootstrapper] [Start] - SchedulerDB: server=SISIFOS;user id=sqladmin;pas...

07/02/2018 15:59:00,476 [32] [INFO ] [Hangfire.SqlServer.SqlServerStorage] [CallSite.Target] - **Start installing Hangfire SQL objects...**

07/02/2018 15:59:00,491 [32] [INFO ] [Hangfire.SqlServer.SqlServerStorage] [CallSite.Target] - Hangfire SQL objects installed.

07/02/2018 15:59:00,491 [43] [INFO ] [Hit.Services.Scheduler.Core.HangfireBootstrapper] [Start] **- SchedulerDB**: server=SISIFOS;user id=sqladmin;pas...

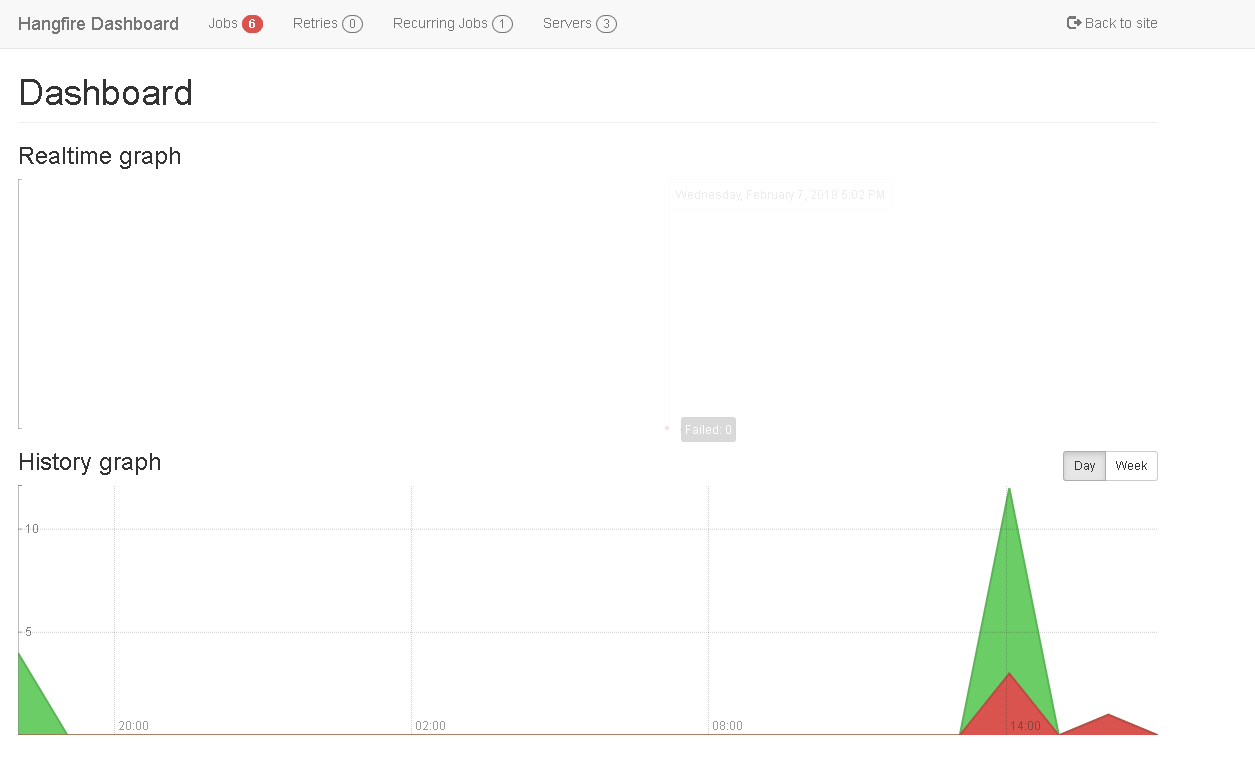
07/02/2018 15:59:00,507 [32] [INFO ] [Hit.Services.Scheduler.Core.HangfireBootstrapper] [addJobs] - **Adding Scheduled Jobs...**

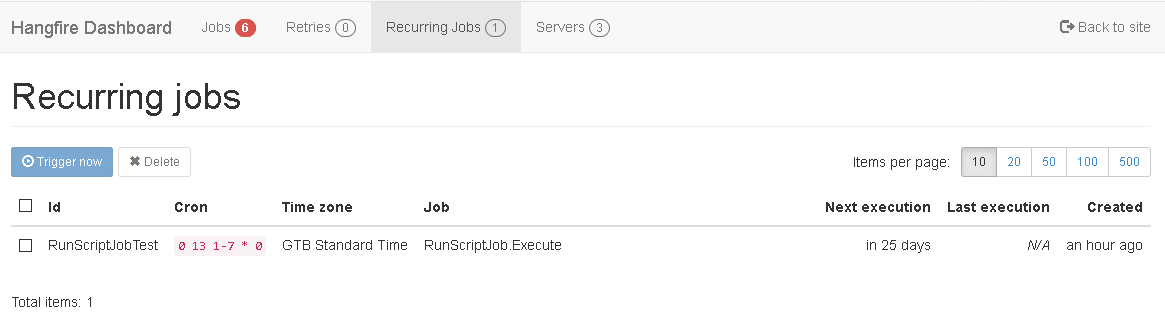
07/02/2018 15:59:00,522 [32] [INFO ] [Hit.Services.Scheduler.Core.HangfireBootstrapper] [addJobs] - **Job: Test Job 1, ClassName: Hit.Services.Scheduler.Jobs.TestJob, IsActive: False, Cron:\* \* \* \* \*, Params: [TestJob1,TestJob1]**

07/02/2018 15:59:00,538 [32] [INFO ] [Hit.Services.Scheduler.Core.HangfireBootstrapper] [addJobs] - **Job is ACTIVE**

Αν έχει ενεργοποιηθεί το dashboard η πρόσβαση σε αυτόν γίνεται από ένα web browser :

**<api url>/scheduler**





# **Διαθέσιμα Services**

Υπάρχουν τα παρακάτω διαθέσιμα Services**:**

## **Generic Services**

### **RunScript (generic)**

Το Service εκτελεί ένα sql script σε μία DB.

|  |  |
| --- | --- |
| **Job** | **RunScriptJob** |
| **Web Api** | **Controller: SqlController**  **URL: <basic url>/api/data/run/<settings>**  **Return: -**  Όπου <settings> το όνομα ενός settings file (προαιρετικά). Αν στο url δε δοθεί settings file τότε θα πρέπει να έχει προ-οριστεί setting file για τον controller στον HitServicesConfig (βλ καρτέλες των *settings* και *api settings*). |

Το configuration του setting file περιλαμβάνει:

* Connection string για την DB
* Sql script το οποίο θα εκτελεί τις αναμενόμενες ενέργειες
* Αρχικές τιμές παραμέτρων του sql script (προαιρετικά)

Στο setting file θα πρέπει να οριστεί ένα sql script. To sql script εκτελεί τις αναμενόμενες ενέργειες στην DB και προαιρετικά επιστρέφει select με τιμές παραμέτρων τις οποίες το sql script θα χρησιμοποιήσει στην επόμενη εκτέλεσή του.

Παράδειγμα:

*Κάθε φορά που τρέχει το service θέλουμε να εκτελείται update στο table EndOfDay\_Hist toy web\_pos από μία τιμή των id και πάνω.*

Θα δημιουργηθεί ένα sql script το οποίο θα εκτελεί το επιθυμητά updates για ids μεγαλύτερα από την τιμή της μεταβλητής @Id ,και στο τέλος θα επιστρέφει το μεγαλύτερο id που χρησιμοποίησε:

IF object\_id('tempdb..#TempTable') is not null drop table #TempTable

select id INTO #TempTable FROM EndOfDay\_Hist where **Id**>**@Id**

**update** EndOfDay\_Hist set Barcodes=0

where Barcodes is null and **id>@Id** **-- <-- update**

**SELECT max(Id) as Id** from #TempTable  **-- <-- το μεγαλύτερο id που χρησιμοποίησε**

drop table #TempTable

επίσης στο configuration θα πρέπει να οριστεί η αρχική τιμή του **@Id**:

(προσοχή στο **@**)

Οι μεταβλητές είναι case sensitive, πχ: σωστό ζευγάρι:Id και @Id, λάθος ζευγάρι: Id και @id

Πριν τη πρώτη εκτέλεση του sql script το service θα αντικαταστήσει τη μεταβλητή @Id με την αρχική τιμή 0 που ορίστηκε στο configuration. Μετά την εκτέλεση του sql script το service θα λάβει τη νέα τιμή της μεταβλητής @Id και θα την αποθηκεύσει στην DB HitServices. Στη συνέχεια, σε κάθε νέα εκτέλεση του Service αυτό θα λαμβάνει την τιμή της @Id από την DB για να την αντικαταστήσει στο sql script, και θα αποθηκεύει στην DB κάθε φορά την ανανεωμένη τιμή που προκύπτει από την εκτέλεση του sql script.

|  |  |  |
| --- | --- | --- |
| Εφαρμογές | | |
| Πελάτης | **Folder** | **Περιγραφή** |
|  |  |  |
|  |  |  |
|  |  |  |



### **Export Data (generic)**

Το Service λαμβάνει data από μία DB και τη συνέχεια τα αποθηκεύει σε αρχείο(α) ή/και τα αποστέλλει σε τρίτο rest server(Post). Σε περίπτωση που χρησιμοποιηθεί ο αντίστοιχος controller τότε το Service επιστρέφει τα data στον caller.

|  |  |
| --- | --- |
| **Job** | **ExportDataJob** |
| **Web Api** | **Controller: SqlController**  **URL: <basic url>/api/data/getdata/<settings>**  **Return:** exported data as IEnumerable<dynamic>  Όπου <settings> το όνομα ενός settings file (προαιρετικά). Αν στο url δε δοθεί settings file τότε θα πρέπει να έχει προ-οριστεί setting file για τον controller στον HitServicesConfig (βλ καρτέλες των *settings* και *api settings*). |

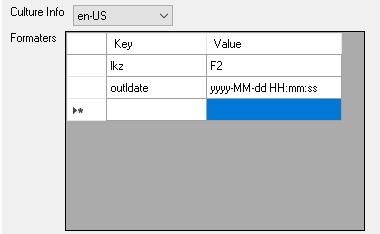
Το configuration του setting file περιλαμβάνει:

* Connection string για την DB
* Απαιτούμενη πληροφορία για export σε αρχεία **csv, json, xml, html, fixed length, pdf** ή αποστολή **σε τρίτο rest server**. Για έξοδο σε αρχείο απαιτείται τουλάχιστον το path του αρχείου. Αν δε δοθεί path για κάποιο τύπο αρχείου τότε δεν θα εκδοθεί το αντίστοιχο αρχείο. Παρόμοια αν δεν οριστεί url για τρίτο rest server τότε το service δε θα αποστείλει τα data σε rest server.
* Formatters και cultural info για όσα πεδία επιθυμούμε να έχουν συγκεκριμένη μορφοποίηση.

Πχ: για ημερομηνίες : **yyyy-MM-dd**,

Για αριθμούς: **F2**: 1234.45, **N2**: 1.234,56

Στο configuration θέτουμε για key το όνομα του πεδίου για το οποίο επιθυμούμε ειδικό format και σαν value τη μορφοποίηση:



* Sql script το οποίο θα εκτελεί τις αναμενόμενες ενέργειες (selects)
* Αρχικές τιμές παραμέτρων του sql script (προαιρετικά)

Το sql script το οποίο θα χρησιμοποιήσει το service για να λάβει τα data από την DB θα πρέπει να επιστρέφει ένα ή δύο selects. To 1ο select επιστρέφει τα data, και το 2ο select (προαιρετικό) επιστρέφει τις τιμές των παραμέτρων τις οποίες το sql script θα χρησιμοποιήσει στην επόμενη εκτέλεσή του (βλ παράδειγμα στο RunScript).

|  |  |  |
| --- | --- | --- |
| Εφαρμογές | | |
| Πελάτης | **Folder** | **Περιγραφή** |
|  |  |  |
|  |  |  |
|  |  |  |

### **Save To Table (generic)**

Το Service λαμβάνει data από μία DB και τη συνέχεια τα αποθηκεύει σε Table στην ίδια DB ή σε διαφορετική DB.

|  |  |
| --- | --- |
| **Job** | **SaveToTableJob** |
| **Web Api** | **Controller: -** |

Το configuration του setting file περιλαμβάνει:

* Connection string για τις DBs (source και destination)
* Destination Table name
* Type of operation: inserts & updates, inserts only, updates only
* Use transaction
* Sql script το οποίο θα εκτελεί τις αναμενόμενες ενέργειες (selects)
* Αρχικές τιμές παραμέτρων του sql script (προαιρετικά)
* Script to οποίο θα τρέχει στην destination DB πριν τρέξουν τα insert/update queries σε αυτή.

Το sql script το οποίο θα χρησιμοποιήσει το service για να λάβει τα data από την DB θα πρέπει να επιστρέφει ένα ή δύο selects. To 1ο select επιστρέφει τα data τα οποία θα αποθηκευτούν στο table, και το 2ο select (προαιρετικό) επιστρέφει τις τιμές των παραμέτρων τις οποίες το sql script θα χρησιμοποιήσει στην επόμενη εκτέλεσή του (βλ παράδειγμα στο RunScript).

Ακολουθεί παράδειγμα script το οποίο κάθε φορά λαμβάνει 10 εγγραφές από το table EndOfDay του webpos. Επιστρέφει επίσης και το μεγαλύτερο id που χρησιμοποιήθηκε για χρήση κατά την επόμενη εκτέλεση του script :

IF object\_id('tempdb..#TempTable') is not null drop table #TempTable

select TOP (10) \*,year(FODay) nYear INTO #TempTable FROM EndOfDay where **Id>@Id**

SELECT \* from #TempTable **-- <-- 1st Data Set (Data)**

**SELECT max(Id) as Id** from #TempTable **-- <-- 2nd Data Set (Updated sql parameters)**

drop table #TempTable

|  |  |  |
| --- | --- | --- |
| Εφαρμογές | | |
| Πελάτης | **Folder** | **Περιγραφή** |
| Μήτσης | \\sisifos\ftp\HitServices\Implementations\Mitsis\Reservations | * Αποθήκευση νέων κρατήσεων σε βάση εκτός protel στον πίνακα **Reservations**. * Απαιτεί την υπαρξη της 2ης βάσης. * Στα settings στην παραμετρο ‘Sql params and initial values’ το value της @Id είναι η τιμή του bnr\_neu.kdnr με την οποία θα γίνει το 1ο run. Θα πρέπει να οριστεί αρχική τιμή για το **bnr\_neu.kdnr** που θα κάνει select τα data. * Operation: Insert only   **Για DB scripts βλ \Implementations\Mitsis\Reservations\readme.txt** |
| Μήτσης | \\sisifos\ftp\HitServices\Implementations\Mitsis\Invoices | * Αποθήκευση παραστατικών της προηγούμενης ημέρας (εκτός των ακυρωμένων και των ακυρωτικών) σε DB εκτός protel στον πίνακα **Invoices**. * Απαιτεί την υπαρξη της 2ης βάσης. * Operation: Insert only   **Για DB scripts βλ \Implementations\Mitsis\Invoices\readme.txt** |
|  |  |  |

### **Read from csv (generic)**

Το Service διαβάζει data από ένα csv αρχείο και τα εισάγει (inserts & updates, inserts only, updates only) σε ένα destination DB table. Τα ονόματα των πεδίων του header του αρχείου πρέπει να αντιστοιχούν σε ονόματα πεδίων στο destination DB table.

|  |  |
| --- | --- |
| **Job** | **ReadFromCsvJob** |
| **Web Api** | **Controller: -** |

Το configuration του setting file περιλαμβάνει:

* Connection string για την destination
* Destination Table name
* Type of operation: inserts & updates, inserts only, updates only
* Use transaction
* Path για το csv αρχείο και άλλες πληροφορίες
* Αρχικές τιμές παραμέτρων του sql script (προαιρετικά)
* Script to οποίο θα τρέχει στην destination DB πριν τρέξουν τα insert/update queries σε αυτή.

### **Forex (generic)**

Job το οποίο συνδέεται με web service της fixer.io, λαμβάνει τα rates από επιλεγμένα νομίσματα και τα αποθηκεύει (insert/update) σε επιλεγμένη DB στο table ‘ForexService’ (Αν το table δεν υπάρχει τότε το δημιουργεί). Η ημερομηνία που καταχωρείται στην DB προέρχεται και αυτή από το service της fixer.

Σε περίπτωση σφάλματος το service εκτελείται και πάλι (έως 4 φορές).

Περισσότερες πληροφορίες: https://fixer.io/documentation

|  |  |
| --- | --- |
| **Job** | **FixerJob** |
| **Web Api** | **Controller: -** |

Το configuration περιλαμβάνει:

* Currency From: το νόμισμα για το οποίο λαμβάνονται τα rates (EUR)
* Currencies To: λίστα με τα νομίσματα των rates. Πχ: USD,AUD,CAD,PLN,MXN
* Delete Old Rates: Διαγραφή των υπαρχόντων rates στην DB πριν ληφθούν τα νέα rates
* Api Key: Api Key για την επικοινωνία με το web service (fa1da865ec00e61261158866d19f9d68)
* Base Url: Base Url του web service (http://data.fixer.io/api/latest)
* Custom DB 1: Connection string για την DB
* Email configuration για αποστολή email σε περίπτωση σφάλματος (προαιρετικά).

Η εγγραφή στο site της fixer.io για τη λήψη του api key έχει γίνει με:

Username: programmers@hit.com.gr

Password: 123456!@

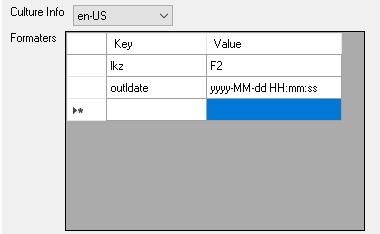
## **Generic Protel Services**

### **Protel Availabilities**

Το Service εξάγει διαθεσιμότητες ανά τύπο δωματίου (για όσους τύπους έχουν οριστεί να είναι διαθέσιμοι στο web, inet=1) και τη συνέχεια τις αποθηκεύει σε αρχείο(α) ή/και τις αποστέλλει σε τρίτο rest server(*Post*). Σε περίπτωση που χρησιμοποιηθεί ο αντίστοιχος controller τότε το Service επιστρέφει τα data στον caller.

|  |  |
| --- | --- |
| **Job** | **AvailabilityJob** |
| **Web Api** | **Controller: ProtelController**  **URL: <basic url>/api/protel/availability/<settings>**  **Return:** exported data as List<ProtelAvailabilityModels>  Όπου <settings> το όνομα ενός settings file (προαιρετικά). Αν στο url δε δοθεί settings file τότε θα πρέπει να έχει προ-οριστεί setting file για τον controller στον HitServicesConfig (βλ καρτέλες των *settings* και *api settings*). |
| **Additional files** | \\sisifos\ftp\HitServices\Implementations\Protel Generic\Availibilities  Αποστολή διαθεσιμότητας για συγκεκριμένη περίοδο μέσω API |

Το configuration του setting file περιλαμβάνει:

* Connection string για την protel DB
* mpehotel
* Απαιτούμενη πληροφορία για export σε αρχεία **csv, json, xml, html, fixed length, pdf** ή αποστολή **σε τρίτο rest server**. Για έξοδο σε αρχείο απαιτείται τουλάχιστον το path του αρχείου. Αν δε δοθεί path για κάποιο τύπο αρχείου τότε δεν θα εκδοθεί το αντίστοιχο αρχείο. Παρόμοια αν δεν οριστεί url για τρίτο rest server τότε το service δε θα αποστείλει τα data σε rest server.
* Formatters και cultural info για όσα πεδία επιθυμούμε να έχουν συγκεκριμένη μορφοποίηση.

Πχ: Για ημερομηνίες : **yyyy-MM-dd**,

Για αριθμούς: **F2**: 1234.45, **N2**: 1.234,56

Στο configuration θέτουμε για key το όνομα του πεδίου για το οποίο επιθυμούμε ειδικό format και σαν value τη μορφοποίηση.

* **Ημερομηνία από** ‘DateFrom’ συλλογής διαθεσιμοτήτων και **περίοδος σε ημέρες**. Αν το DateFrom είναι κενό τότε ημερομηνία έναρξης θα είναι η τρέχουσα ημερομηνία του protel.

### **Protel Rate Codes**

Το Service εξάγει τους τιμοκαταλόγους του Protel. Υπάρχει η επιλογή για την εξαγωγή όλων των τιμοκαταλόγων, όσων εμφανίζονται στο web ή όσων δεν εμφανίζονται στο web. Η πληροφορία που λαμβάνεται από το service για τους τιμοκαταλόγους έχει την ακόλουθη μορφή.

Id τιμοκαταλόγου

Περιγραφή

VipCode

Ο τύπος του τιμοκαταλόγου

Αν είναι εμφανής στο web

Το id του όρου

Η περιγραφή του όρου

Λίστα από ανάλυση του τιμοκαταλόγου με την ακόλουθη μορφή

Το id του detail του τιμοκαταλόγου

Περιγραφή του Detail

Το id του yield του detail

Ημερομηνία έναρξης

Ημερομηνία λήξης

Λιστα από seasons για το detail με την ακόλουθη μορφή

Το id της season

Ημερομηνία έναρξης της season

Ημερομηνία λήξης της season

Λίστα από τύπους δωματίων που ανήκουν στην season με την μορφή

Id για τον τύπο δωματίου

Κωδικός τύπου δωματίου

Περιγραφή τύπου δωματίου

Τιμή για τον ένα ενήλικα

Τιμή για δύο ενήλικες

Τιμή για τρεις ενήλικες

Τιμή για τέσσερις ενήλικες

Τιμή για πέντε ενήλικες

Τιμή για περισσότερους από πέντε ενήλικες

Τιμή για ένα παιδί

Τιμή για δύο παιδιά

|  |  |
| --- | --- |
| **Job** | **RateCodesJob** |
| **Web Api** | **Controller: ProtelController**  **URL: <basic url>/api/protel/ratecodes/<settings>**  **Return:** exported data as ProtelRateCodesForWebModel  Όπου <settings> το όνομα ενός settings file (προαιρετικά). Αν στο url δε δοθεί settings file τότε θα πρέπει να έχει προ-οριστεί setting file για τον controller στον HitServicesConfig (βλ καρτέλες των *settings* και *api settings*). |
| **Additional files** | \\sisifos\ftp\HitServices\Implementations\Protel Generic\RateCodes  **Δημιουργία αρχείων τύπου json και δυνατότητα αποστολής σε API** |

Το configuration του setting file περιλαμβάνει:

* Connection string για την protel DB
* mpehotel
* Επιλογή για τους τ΄πυους τιμοκαταλόγων που θα εξάγουμε.
  + 0 Να μην εμφανίζονται στο web
  + 1 Να εμφανίζονται στο web
  + 2 Όλους τους τιμοκαταλόγους
* Λίστα από ids για τα travel agents για τα οποία θέλουμε να εξάγουμε δεδομένα. Αν είναι άδεια η λίστα τότε εξάγοεμε δεδομένα για όλα τα travel agent.
* Path για την αποθήκευση του αρχείου σε μορφή JSON. Το path είναι η διαδρομή μαζί με το όνομα του αρχείου.
* Στην περίπτωση που θέλουμε να στείλουμε το αρχείο σε συγκεκριμένο API πρέπει να δηλώσουμε και τα ακόλουθα
  + RestServerUrl. Το URL στο οποίο θα γίνει η αποστολή
  + RestServerAuthenticationHeader. Το authentication του URL
  + RestServerHttpMethod. Η μέθοδος (POST)
  + RestServerMediaType. Ο τύπος αποστολής (application/json)

## **Custom Services for SmartIn**

### **ELock (SmartIn)**

Job το οποίο ενημερώνει την κατάσταση των δωματίων στην DB της SmartIn. Θέτει στην DB της SmartIn το πεδία Room.Room\_Blocked true ή false ανάλογα με το αν το δωμάτιο είναι check-in ή checkout.

|  |  |
| --- | --- |
| **Job** | **SmartIn.ELockJob** |
| **Web Api** | **Controller: -** |
| **Additional files** | \\sisifos\ftp\HitServices\Implementations\SmartIn\ELock\files   * **Xml για το protel** * **Εφαρμογή κινητού** στην οποία θα πρέπει να οριστεί το url επικοινωνίας με το api: http://<api url>/api/smartin/elock/GetLockInfo/   (βλ και LockInfo) |

Το configuration του setting file περιλαμβάνει:

* Connection string για την protel DB
* Mpehotel
* Proteluser
* Connection string για την SmartInElockDB

### **LockInfo (SmartIn)**

Controller o οποίος επιστρέφει τον κωδικό της κλειδαριάς στην εφαρμογή android.

|  |  |
| --- | --- |
| **Job** | **-** |
| **Web Api** | **Controller: SmartInElockController**  **Url: <basic url>/api/smartin/elock/GetLockInfo**  **Return:** KeyModel |
| **Additional files** | \\sisifos\ftp\HitServices\Implementations\SmartIn\ELock\files   * **Xml για το protel** * **Εφαρμογή κινητού** στην οποία θα πρέπει να οριστεί το url επικοινωνίας με το api: http://<api url>/api/smartin/elock/GetLockInfo/   (βλ και ELock) |

Το configuration του setting file περιλαμβάνει:

* Connection string για την SmartInElockDB

Θα πρέπει να έχει προ-οριστεί setting file για τον controller στον HitServicesConfig (βλ καρτέλες των *settings* και *api settings*).

To usernameκαι password του Api user πρέπει να είναι: **K29Client** και **!K29Cli@nt**

## **Custom Services for Gorissia**

### **GetProfilesJob (Corissia)**

Job το οποίο ενημερώνει μέσω κλήση σε api του crm σύστημα της Corissia για μεταβολές σε πελάτες. Ηοοκ στο protel αποθηκεύει νέα και αλλαγμένα profiles στο table hit\_CorissiaAPI. To job διαβάζει το table και αποστέλλει τα δεδομένα στο api της Corissia. To api της Corissia για κάθε profile επιστρέφει id to οποία αποθηκεύεται στο πεδίο kunden.gdsident. Τελικά το job διαγράφει τις εγγραφές του table hit\_CorissiaAPI.

Αν το table hit\_CorissiaAPI δεν υπάρχει τότε το job το δημιουργεί.

|  |  |
| --- | --- |
| **Job** | **Corissia. GetProfilesJob** |
| **Web Api** | **Controller:** |
| **Additional files** | \\sisifos\ftp\HitServices\Implementations\Gorissia\GetProfiles |

Το configuration του setting file περιλαμβάνει:

* Connection string για την protel DB
* Mpehotel
* Settings για αποστολή στο api: Όπου Api Url θα πρέπει να οριστεί το base url

Test Corissa api url : http://sisifos:9899/api/Gorissia

Production Corissia api url : https://corissia.org/api/protel

* Authentication για το api
* Μέθοδος αποστολής (πάντα POST)
* Τύπος αποστολής (πάντα application/json)

## **Custom Services for WaterPark**

### **OnlineRegistration (WaterPark)**

Rest web api το οποίο δέχεται αγορές εισιτηρίων από το internet για το waterpark. Οι κρατήσεις αποθηκεύονται σε webpos DB στον πίνακα OnlineRegistration

|  |  |
| --- | --- |
| **Job** | **-** |
| **Web Api** | **Controller: WaterPark. OnlineRegistrationController** |
| **Additional files** |  |

Το configuration του setting file περιλαμβάνει:

* Connection string για την webpos DB

Στην καρτέλα ‘ApiSettings’ θα πρέπει να αντιστοιχηθεί το setting file me ton controller, και στην καρτέλα ‘ApiUsers’ θα πρέπει να δημιουργηθεί user που θα έχει πρόσβαση στον controller. O user πρέπει να δοθεί στην εταιρία του web site του waterpark ώστε το web site να έχει πρόσβαση στον controller.

To service επιστρέφει:

200 Ok

400 Bad Request with an appropriate model's error message

500 Internal Server Error followed by an error code:

2098 :Barcode already exists

3000 :other internal errors

## **Custom Services for Iberostar**

### **SendReservations**

Scheduler Job με την οποία συλλέγονται πληροφορίες για τις κρατήσεις του ξενοδοχείου (σε multi property περιβάλλοντα το κάθε ξενοδοχείο ξεχωριστά) που έχουν σαν παραμέτρους η ημερομηνία αναχώρησης της κράτησης να είναι μεγαλύτερη από την ημερομηνία του Protel και το status της κράτησης να μην είναι 6 (block).

|  |  |
| --- | --- |
| **Job** | **SendReservationsJob** |
| **Web Api** |  |
| **Additional files** | \\sisifos\ftp\HitServices\Implementations\Iberostar\SendReservations |

Το configuration του setting file περιλαμβάνει:

* Connection string για την protel DB
* mpeHotel. Αν είναι μικρότερο του 1 (ένα) τότε βρίσκει όλα τα Id των ξενοδοχείων και εκτελεί την διαδικασία για κάθε ένα ξεχωριστά αλλιώς εκτελεί την διαδικασία για το επιλεγμένο ξενοδοχείο.
* xmlPath. Η αρχική τιμή είναι directory+Reservations.xml. Κατά την αποθήκευση των αρχείων το όνομα αλλάζει σε mpehotel\_short.xml όπου mpehotel είναι το Id του ξενοδοχείου και short το όνομα του.

Η επιστροφή του scheduler Job είναι ένα μορφοποιημένο XML αρχείο. Μέσα στο directory file υπάρχει δείγμα του XML.

## **Custom Services From Singular**

### **Fuel Exports**

Scheduler Job το οποίο διαβάσει text αρχείο με εκροές καυσίμων, από το σύστημα StationManager της Singular, και εισάγει τις εγγραφές του στο table leist του protel.

|  |  |
| --- | --- |
| **Job** | **FuelExportsJob** |
| **Web Api** |  |
| **Additional files** | \\sisifos\ftp\HitServices\Implementations\Singular\FuelExports |

Το configuration του setting file περιλαμβάνει:

* ProtelDB. Connection string για την protel DB
* mpeHotel. το επιλεγμένο ξενοδοχείο.
* ProtelTypeToRoom. Ορίζεται το matching κάθε τύπου καυσίμου σε ένα room.
* ProtelTypeToDepartment. Ορίζεται το matching κάθε τύπου καυσίμου σε ένα τμήμα.
* FileInfoDirectory. Το folder στο οποίο υπάρχει το text αρχείο
* FileInfoFileName. Το όνομα του αρχείου. Πάντα: SMexp-\*.exp
* FileInfoCommentsChar. Χαρακτήρας σχολίων. Κάθε γραμμή αρχείου η οποία αρχίζει με αυτόν τον χαρακτήρα θα θεωρείται σχόλιο και δε θα επεξεργάζεται. Πάντα: ;

Σε περίπτωση σφάλματος καμία εγγραφή του αρχείου δεν εισάγεται στην DB