

CEC RSA

Proiect tehnic



30 IANUARIE 2015

Comisia electorală centrală a republicii moldova

Vasile Alecsandri, 119, Chișinău, Republica Moldova

**CUPRINS**

[1 Descrierea generală a sistemului CEC RSA 7](#_Toc410403145)

[1.1 Cerințe Față de Sistem 7](#_Toc410403146)

[1.2 Arhitectura generală a sistemului 7](#_Toc410403147)

[1.3 Diagrama nomenclatoarelor sistemului 9](#_Toc410403148)

[2 Cazuri de utilizare 10](#_Toc410403149)

[2.1 Nivelul general de utilizare 10](#_Toc410403150)

[2.2 Actualizarea datelor din RSP 11](#_Toc410403151)

[2.3 Administrarea nomenclatoarelor 12](#_Toc410403152)

[2.4 Adăugarea înregistrării 12](#_Toc410403153)

[2.5 Eliminarea înregistrării 13](#_Toc410403154)

[2.6 Modificarea înregistrării 14](#_Toc410403155)

[2.7 Restabilirea datelor 15](#_Toc410403156)

[2.8 Administrarea Utilizatorilor 15](#_Toc410403157)

[2.9 Adăugarea utilizatorilor 16](#_Toc410403158)

[2.10 Alocarea regiunilor 16](#_Toc410403159)

[2.11 Eliminarea utilizatorilor 17](#_Toc410403160)

[2.12 Modificarea în grup a utilizatorilor 18](#_Toc410403161)

[2.13 Modificarea utilizatorilor 18](#_Toc410403162)

[2.14 Gestionarea Alegătorilor 19](#_Toc410403163)

[2.15 Depunerea declarației de ședere 20](#_Toc410403164)

[2.16 Înregistrarea declarației de ședere 20](#_Toc410403165)

[2.17 Modificarea statutului alegătorului 21](#_Toc410403166)

[2.18 Gestionarea Profilului utilizatorilor 22](#_Toc410403167)

[2.19 Gestionarea profilului personal Și Modificarea Parolei 22](#_Toc410403168)

[2.20 Gestionarea profilului Utilizatorului 23](#_Toc410403169)

[2.21 Resetarea parolei 24](#_Toc410403170)

[2.22 Gestionarea Sectoarelor De Votare 24](#_Toc410403171)

[2.23 Adăugarea Sectoarelor De Votare 25](#_Toc410403172)

[2.24 Modificarea Sectoarelor De Votare 26](#_Toc410403173)

[2.25 Generarea conflictelor 26](#_Toc410403174)

[2.26 Rezolvarea conflictelor De Adresă 28](#_Toc410403175)

[2.27 Rezolvarea conflictelor De Secție de Votare 29](#_Toc410403176)

[2.28 Rezolvarea conflictelor De Lipsă Localitate 30](#_Toc410403177)

[2.29 Rezolvarea conflictelor De Localitate 31](#_Toc410403178)

[2.30 Rezolvarea conflictelor De Statut 32](#_Toc410403179)

[3 Componenta CEC.SRV.BLL 34](#_Toc410403180)

[3.1 Subcomponenta BLL 35](#_Toc410403181)

[3.2 Subcomponenta Address Bll 37](#_Toc410403182)

[3.3 Subcomponenta Auditer Bll 41](#_Toc410403183)

[3.4 Subcomponenta Conflict Bll 41](#_Toc410403184)

[3.5 Subcomponenta Election Bll 45](#_Toc410403185)

[3.6 Subcomponenta Exporter Bll 46](#_Toc410403186)

[3.7 Subcomponenta Import Bll 47](#_Toc410403187)

[3.8 Subcomponenta Import Statistics Bll 49](#_Toc410403188)

[3.9 Subcomponenta Linked Region Cache 49](#_Toc410403189)

[3.10 Subcomponenta Lookup Bll 49](#_Toc410403190)

[3.11 Subcomponenta Notification Bll 57](#_Toc410403191)

[3.12 Subcomponenta Polling Station Bll 58](#_Toc410403192)

[3.13 Subcomponenta Polling Station Cache 60](#_Toc410403193)

[3.14 Subcomponenta Print Bll 60](#_Toc410403194)

[3.15 Subcomponenta Quartz Lookup Cache Data Provider 62](#_Toc410403195)

[3.16 Subcomponenta Region Cache 62](#_Toc410403196)

[3.17 Subcomponenta Statistics Bll 62](#_Toc410403197)

[3.18 Subcomponenta Street Type Cache 65](#_Toc410403198)

[3.19 Subcomponenta Street Type Code Cache 66](#_Toc410403199)

[3.20 Subcomponenta Web Lookup Cache Data Provider 66](#_Toc410403200)

[3.21 Subcomponenta Voters Bll 66](#_Toc410403201)

[3.22 Subcomponenta Pre Print Lists Exporter 72](#_Toc410403202)

[3.23 Subcomponenta Import Processer 72](#_Toc410403203)

[3.24 Subcomponenta Alegator Processer 73](#_Toc410403204)

[3.25 Subcomponenta RSA User ProcessOr 73](#_Toc410403205)

[3.26 Subcomponenta RSP Modification ProcessEr 74](#_Toc410403206)

[3.27 Subcomponenta Jobs Scheduler 74](#_Toc410403207)

[3.28 Subcomponenta Srv Repository 75](#_Toc410403208)

[3.29 Subcomponenta Auditer Repository 75](#_Toc410403209)

[3.30 Subcomponenta wEB aPI sRV Repository 76](#_Toc410403210)

[4 Componenta CEC.Web.SRV 77](#_Toc410403211)

[4.1 Subcomponenta Base Controller 77](#_Toc410403212)

[4.2 Subcomponenta Account Controller 77](#_Toc410403213)

[4.3 Subcomponenta Address Controller 80](#_Toc410403214)

[4.4 Subcomponenta Conflict Controller 82](#_Toc410403215)

[4.5 Subcomponenta Election Controller 86](#_Toc410403216)

[4.6 Subcomponenta Error Controller 87](#_Toc410403217)

[4.7 Subcomponenta Export Controller 87](#_Toc410403218)

[4.8 Subcomponenta History Controller 90](#_Toc410403219)

[4.9 Subcomponenta Home Controller 94](#_Toc410403220)

[4.10 Subcomponenta Notification Controller 94](#_Toc410403221)

[4.11 Subcomponenta Polling Station Controller 95](#_Toc410403222)

[4.12 Subcomponenta Quartz Admin Controller 96](#_Toc410403223)

[4.13 Subcomponenta Reporting Controller 97](#_Toc410403224)

[4.14 Subcomponenta Statistics Controller 98](#_Toc410403225)

[4.15 Subcomponenta Voters Controller 100](#_Toc410403226)

[4.16 Subcomponenta Lookup Controller 104](#_Toc410403227)

[5 Componenta CEC.Web.API 113](#_Toc410403228)

[5.1 Subcomponenta Home Controller 113](#_Toc410403229)

[5.2 Subcomponenta Polling Station Controller 113](#_Toc410403230)

[5.3 Subcomponenta Statistic Controller 114](#_Toc410403231)

[5.4 Subcomponenta Where To Vote Controller 114](#_Toc410403232)

[6 Componenta CEC.Web.Results.API 116](#_Toc410403233)

[6.1 Subcomponenta Home Controller 116](#_Toc410403234)

[6.2 Subcomponenta Image Route Handler 116](#_Toc410403235)

[6.3 Subcomponenta Live Election Turnout Controller 116](#_Toc410403236)

[6.4 Subcomponenta Live Election Turnout Handler 117](#_Toc410403237)

[6.5 Subcomponenta Live Results Controller 117](#_Toc410403238)

[6.6 Subcomponenta Live Results Handler 117](#_Toc410403239)

[6.7 Subcomponenta Polit Party Logo Image Handler 118](#_Toc410403240)

[6.8 Subcomponenta Live Results Logic 119](#_Toc410403241)

[6.9 Subcomponenta Sql Factory 120](#_Toc410403242)

[7 Componenta CEC.Web.Public.Modules 123](#_Toc410403243)

[7.1 Scriptul Java CEC 123](#_Toc410403244)

[7.2 Scriptul Java CEC2 124](#_Toc410403245)

[7.3 Scriptul Java Election Turnout 125](#_Toc410403246)

[7.4 Scriptul Java Live Results 126](#_Toc410403247)

[7.5 Scriptul Java Preliminaries 127](#_Toc410403248)

[7.6 Scriptul Java Register 128](#_Toc410403249)

[7.7 Scriptul Java Statistics 129](#_Toc410403250)

[8 Componenta Quartz 130](#_Toc410403251)

[8.1 Subcomponenta CEC.QuartzServer.Console.Program 130](#_Toc410403252)

[8.2 Subcomponenta CEC.QuartzServer.Core.CecQuartzServer 130](#_Toc410403253)

[8.3 Subcomponenta CEC.QuartzServer.Core.Configuration 130](#_Toc410403254)

[8.4 Subcomponenta CEC.QuartzServer.Core.JobBase 131](#_Toc410403255)

[8.5 Subcomponenta CEC.QuartzServer.Core.QuartzSecurityContext 131](#_Toc410403256)

[8.6 Subcomponenta CEC.QuartzServer.Core.QuartzServerFactory 131](#_Toc410403257)

[8.7 Subcomponenta CEC.QuartzServer.Core.SaiseDatabaseFactoryProvider 131](#_Toc410403258)

[8.8 Subcomponenta CEC.QuartzServer.Core.UnityConfig 132](#_Toc410403259)

[8.9 Subcomponenta CEC.QuartzServer.Core.UnityJobFactory 132](#_Toc410403260)

[8.10 Subcomponenta CEC.QuartzServer.Jobs.Export.SaiseExporterJob 132](#_Toc410403261)

[8.11 Subcomponenta CEC.QuartzServer.Jobs.Export.SaiseExportStageExecuter 132](#_Toc410403262)

[8.12 Subcomponenta CEC.QuartzServer.Jobs.Export.SaiseRepository 133](#_Toc410403263)

[8.13 Subcomponenta CEC.QuartzServer.Jobs.Export.UpdateConflictedSaiseVoterStageExecuter 135](#_Toc410403264)

[8.14 Subcomponenta CEC.QuartzServer.Jobs.Export.UpdateSaiseVoterStageExecuter 136](#_Toc410403265)

[8.15 Subcomponenta CEC.QuartzServer.Jobs.Export.ValidateElectionStageExecuter 136](#_Toc410403266)

[8.16 Subcomponenta CEC.QuartzServer.Jobs.Export.ValidatePollingStationsStageExecuter 136](#_Toc410403267)

[8.17 Subcomponenta CEC.QuartzServer.Jobs.Export.ValidateRegIonsStageExecuter 136](#_Toc410403268)

[8.18 Subcomponenta CEC.QuartzServer.Jobs.Import.AlegatorProcessorJob 137](#_Toc410403269)

[8.19 Subcomponenta CEC.QuartzServer.Jobs.Import.RsaUserProcessorJob 137](#_Toc410403270)

[8.20 Subcomponenta CEC.QuartzServer.Jobs.Import.RspModificationImporterJob 137](#_Toc410403271)

[8.21 Subcomponenta CEC.QuartzServer.Jobs.Import.RspModificationProcessorJob 137](#_Toc410403272)

[8.22 Subcomponenta CEC.QuartzServer.Jobs.Import.RemoveStayStatementsJob 138](#_Toc410403273)

[8.23 Subcomponenta CEC.QuartzServer.Jobs.ListsPrinting.ListsPrintingJob 138](#_Toc410403274)

[8.24 Subcomponenta CEC.QuartzServer.Jobs.Reporting.PrintOutBaseJob 138](#_Toc410403275)

[8.25 Subcomponenta CEC.QuartzServer.Jobs.Reporting.ResultsReportsPrintOutJob 139](#_Toc410403276)

[8.26 Subcomponenta CEC.QuartzServer.Jobs.Reporting.TurnOutReportsPrintOutJob 139](#_Toc410403277)

[8.27 Subcomponenta CEC.QuartzServer.Jobs.SAISE.TurnOutFixationJob 139](#_Toc410403278)

[8.28 Subcomponenta CEC.QuartzServer.Service.Program 140](#_Toc410403279)

[8.29 Subcomponenta CEC.QuartzServer.Service.QuartzService 140](#_Toc410403280)

[8.30 Subcomponenta RSPChecker.Program 140](#_Toc410403281)

[9 Componenta CEC.SRV.Domain 141](#_Toc410403282)

[9.1 Entitatea Abroad Voter Registration 141](#_Toc410403283)

[9.2 Entitatea Additional User Info 141](#_Toc410403284)

[9.3 Entitatea Address 141](#_Toc410403285)

[9.4 Entitatea Address With Count Of People 142](#_Toc410403286)

[9.5 Entitatea Doc Type Code 142](#_Toc410403287)

[9.6 Entitatea Reg Type Code 142](#_Toc410403288)

[9.7 Entitatea Sex Code 142](#_Toc410403289)

[9.8 Entitatea Street Type Code 143](#_Toc410403290)

[9.9 Entitatea Election 143](#_Toc410403291)

[9.10 Entitatea Event 143](#_Toc410403292)

[9.11 Entitatea Geo Location 143](#_Toc410403293)

[9.12 Entitatea Identity User Region View 144](#_Toc410403294)

[9.13 Entitatea Linked Region 144](#_Toc410403295)

[9.14 Entitatea Linked Regions Full Name 144](#_Toc410403296)

[9.15 Entitatea Notification 144](#_Toc410403297)

[9.16 Entitatea Notification Receiver 144](#_Toc410403298)

[9.17 Entitatea Person 145](#_Toc410403299)

[9.18 Entitatea Person Address 145](#_Toc410403300)

[9.19 Entitatea Person By Conflict 146](#_Toc410403301)

[9.20 Entitatea Person Document 146](#_Toc410403302)

[9.21 Entitatea Person Full Address 146](#_Toc410403303)

[9.22 Entitatea Person Status 147](#_Toc410403304)

[9.23 Entitatea Polling Station 147](#_Toc410403305)

[9.24 Entitatea Polling Station Statistics 147](#_Toc410403306)

[9.25 Entitatea Polling Station With Full Address 148](#_Toc410403307)

[9.26 Entitatea Problematic Data Polling Station Statistics 148](#_Toc410403308)

[9.27 Entitatea Public Administration 148](#_Toc410403309)

[9.28 Entitatea Region With Fully Qualified Name 148](#_Toc410403310)

[9.29 Entitatea Region Without Saise Id 149](#_Toc410403311)

[9.30 Entitatea SRV Identity User 149](#_Toc410403312)

[9.31 Entitatea Stay Statement 149](#_Toc410403313)

[9.32 Entitatea Street With count Of Addresses 150](#_Toc410403314)

[9.33 Entitatea Export Polling Station 150](#_Toc410403315)

[9.34 Entitatea Print Entity 150](#_Toc410403316)

[9.35 Entitatea Print Session 150](#_Toc410403317)

[9.36 Entitatea Classifier 151](#_Toc410403318)

[9.37 Entitatea Document Type 151](#_Toc410403319)

[9.38 Entitatea Election Type 151](#_Toc410403320)

[9.39 Entitatea Gender 151](#_Toc410403321)

[9.40 Entitatea Gender 151](#_Toc410403322)

[9.41 Entitatea Lookup 152](#_Toc410403323)

[9.42 Entitatea Manager Type 152](#_Toc410403324)

[9.43 Entitatea Person Address Type 152](#_Toc410403325)

[9.44 Entitatea Person Status Type 152](#_Toc410403326)

[9.45 Entitatea Region 152](#_Toc410403327)

[9.46 Entitatea Region Type 153](#_Toc410403328)

[9.47 Entitatea Street 153](#_Toc410403329)

[9.48 Entitatea Street Type 153](#_Toc410403330)

[9.49 Entitatea Alegator Data 154](#_Toc410403331)

[9.50 Entitatea Import Statistics 154](#_Toc410403332)

[9.51 Entitatea Mapping Address 154](#_Toc410403333)

[9.52 Entitatea Person Raw 155](#_Toc410403334)

[9.53 Entitatea Raw Address 155](#_Toc410403335)

[9.54 Entitatea Raw Data 155](#_Toc410403336)

[9.55 Entitatea Raw Document 156](#_Toc410403337)

[9.56 Entitatea RSA User 156](#_Toc410403338)

[9.57 Entitatea RSP Conflict Data 156](#_Toc410403339)

[9.58 Entitatea RSP Conflict Data Admin 157](#_Toc410403340)

[9.59 Entitatea RSP Modification Data 158](#_Toc410403341)

[9.60 Entitatea RSP Registration Data 159](#_Toc410403342)

[9.61 Entitatea Saise Exporter 160](#_Toc410403343)

[9.62 Entitatea Saise Exporter Stage 160](#_Toc410403344)

[9.63 Entitatea Address With Polling Station 160](#_Toc410403345)

[9.64 Entitatea Configuration Setting 161](#_Toc410403346)

[9.65 Entitatea RSP Address Mapping 161](#_Toc410403347)

# Descrierea generală a sistemului CEC RSA

## Cerințe Față de Sistem

Sistemul CEC RSA este destinat pentru trei categorii de utilizatori: *administratori, registratori, și utilizatori publici*.

În cadrul categoriei utilizatori publici sunt incluși toți cetățenii Republicii Moldova. Accesul acestora la sistem este liber, dar restricționat, ei având acces la un număr limitat de funcționalități, care sunt publice. Intrarea în sistem se face fără autentificare, o cerință de bază a sistemului fiind existența conexiunii la Internet și un navigator pentru a accesa paginile web furnizate de sistem.

Accesul administratorilor și registratorilor la sistem se efectuează prin autentificare. Administratorii au acces la toate funcționalitățile sistemului, iar registratorii – la un set restrâns de funcționalități, în dependență de atribuțiile și responsabilitățile acestora. Astfel, acești utilizatori au nevoie de un cont de utilizator cu rolul respectiv în sistem, conexiune la rețea și navigator web.

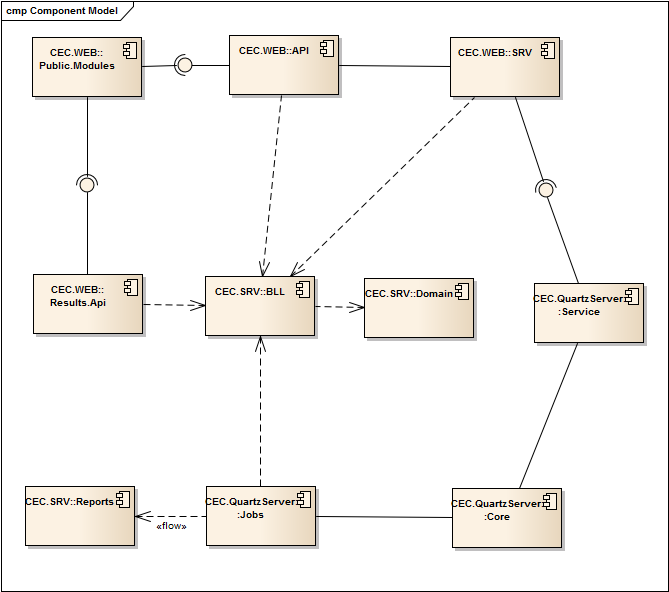
**Navigatoare compatibile**: IE (de la versiune 9.0), Mozilla Firefox, Chrome, etc.

**Sisteme de operare:** Windows

## Arhitectura generală a sistemului

Arhitectura generală a sistemului este ilustrată în **Diagrama 1.2.1**.Astfel, sistemul este compus din mai multe componente principale, precum:

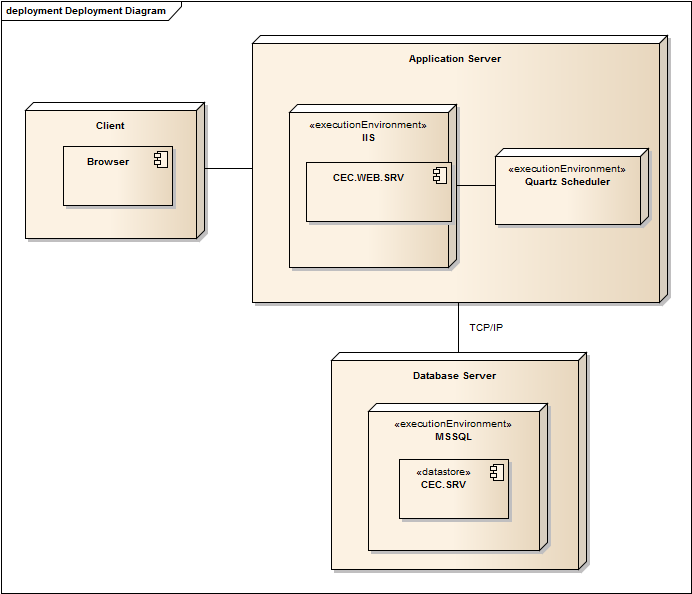
* *CEC.SRV.Domain* – componenta care se identifică cu baza de date a sistemului, având acces direct la baza de date propriu-zisă, găzduită de MS SQL Server, prin intermediul entităților care crează modelul bazei date în cadrul aplicației;
* *CEC.SRV.BLL –* componenta care interacționează cu baza de date prin intermediul modelelor componentei *CEC.SRV.Domain* pentru a recepționa și a reactualiza informația existentă în baza de date a sistemului;



**Diagrama 1.2.1 Arhitectura generală a sistemului**

* *CEC.WEB.SRV* este interfața principală cu utilizatorul și este plasată în IIS. Aceasta oferă majoritatea funcționalităților, permițând utilizatorului să gestioneze baza de date a sistemului prin intermediul componentei *CEC.SRV.BLL*;
* *CEC.WEB.Results.Api* și *CEC.WEB.API* sunt două componente care oferă servicii Web utilizatorilor. Interfața *CEC.WEB.API* difuzează funcționalități de gestionare înainte de începerea procesului de votare (precum verificarea listei electorale, aflarea locului votării, etc.), iar *CEC.WEB.Results.Api* răspunde de rezultatul votării, inclusiv efectuarea statisticilor în perioada imediat următoare procesului de votare;
* *CEC.WEB.Public.Modules –* este o componentă care oferă tuturor cetățenilor funcționalități publice, fără a fi necesară autentificarea acestora în sistem. Modulele acestui subsistem sunt dezvoltate doar cu ajutorul limbajelor javascript și html pentru a evita dependența de platformă. Astfel, *CEC.WEB.Public.Modules* interacționează cu componentele de interfață *CEC.WEB.API* și *CEC.WEB.Results.Api* pentru a accesa *CEC.SRV.BLL*;
* *QUARTZ (CEC.QuartzServer.Service, CEC.QuartzServer.Core* și *CEC.QuartzServer.Jobs)* permite executarea proceselor de lungă durată, externalizarea acestora prin expunerea pe alte servere și gestionarea lor, fie din fișiere de configurare, fie din *CEC.WEB.SRV*. De obicei, aceste procese de lungă durată au ca scop exportarea din sistem a informației în anumite formate compatibile cu alte sisteme externe;
* *CEC.SRV.Reports* oferă diverse rapoarte în baza informației furnizate de QUARTZ, în urma executării procesului de lungă durată de către *CEC.QuartzServer.Jobs.*

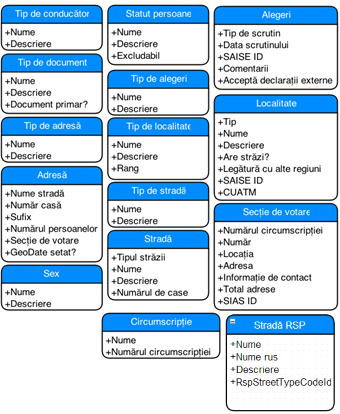
Lanivel de Deployment, structura sistemului poate fi reprezentată de **Diagrama 1.2.2**.

****

**Diagrama 1.2.2 Arhitectura generală a sistemului**

## Diagrama nomenclatoarelor sistemului

Sistemul *CEC.RSA* este format din mai multe nomenclatoare, prezentate în **Diagrama 1.3.1**.



**Diagrama 1.3.1 Nomenclatoarele sistemului**

# Cazuri de utilizare

## Nivelul general de utilizare

**Diagrama 2.1.1** ilustrează nivelul general de utilizare a aplicației RSA. Sunt prezentate funcționalitățile de bază accesibile utilizatorilor cu rol de administrator, registrator și utilizator public de Internet.



**Diagrama 2.1.1 Nivelul general de utilizare**

Astfel, utilizatorii publici de Internet au acces la vizualizarea unor date din RSA, iar utilizatorii cu rol de administrator și registrator au funcționalități extinse. Registratorii pot administra nomenclatoarele sistemului, actualiza datele din RSP, gestiona profilul, gestiona sectoarele de votare, gestiona alegătorii, extrage rapoarte și gestiona conflicte. Administratorii, adițional, pot exporta liste în SIASA și administra utilizatorii aplicației RSA.

## Actualizarea datelor din RSP

Modul de actualizare a datelor din RSP este ilustrat în **Diagrama 2.2.1**. Astfel, se extrag datele de la RSP și se lansează importul. În caz de succes se actualizează baza de date, iar în caz contrar se înregistrează conflictul, se rezolvă conflictul și se reiterează cu actualizarea bazei de date.



**Diagrama 2.2.1 Actualizarea datelor din RSP**

## Administrarea nomenclatoarelor

Modul de administrare a nomenclatoarelor de către utilizatorii cu rol de administrator și registrator este ilustrat în **Diagrama 2.3.1**. Astfel, utilizatorul are posibilitatea de adăugare, eliminare și modificare a înregistrării. De asemenea, sunt disponibile operațiile de restabilire a datelor și setare a sectoarelor de votare.



**Diagrama 2.3.1 Administrarea nomenclatoarelor**

## Adăugarea înregistrării

Adăugarea înregistrării se efectuează conform schemei redate de **Diagrama 2.4.1**. Este necesară navigarea la nomenclatorul respectiv, efectuarea unui click pe + și introducerea datelor. În continuare, este posibilă salvarea datelor sau anularea lor. În caz de anulare datele introduse se pierd, iar în caz de salvare se continuă cu validarea datelor. Dacă datele sunt valide se trece la salvarea înregistrării noi în baza de date, în caz contrar se afișează mesajul de eroare și se revine la nomenclator.



**Diagrama 2.4.1 Adăugarea înregistrării**

## Eliminarea înregistrării

Eliminarea înregistrării se efectuează conform schemei redate de **Diagrama 2.5.1**. Este necesară navigarea la nomenclatorul respectiv, selectarea înregistrării și efectuarea unui click pe *Delete*.



**Diagrama 2.5.1 Eliminarea înregistrării**

Sistemul prezintă utilizatorului un mesaj de confirmare. Dacă se confirmă eliminarea se marchează ca eliminată înregistrarea respectivă, în caz contrar se anulează operația de eliminare. În final, se revine la nomenclator.

## Modificarea înregistrării

Modificarea înregistrării se efectuează conform schemei redate de **Diagrama 2.6.1**. Este necesară navigarea la nomenclatorul respectiv, selectarea înregistrării și efectuarea unui click pe *Modifică*.



**Diagrama 2.6.1 Modificarea înregistrării**

Sistemul prezintă utilizatorului o casetă de dialog pentru modificarea câmpurilor. După modificarea parametrilor utilizatorului are posibilitatea să salveze datele sau să anuleze operația de modificare. Dacă se continuă cu *Salvează* modificările sunt salvate în baza de date, în caz contrar ele se anulează. În final, se revine la nomenclatorul respectiv.

## Restabilirea datelor

Restabilirea datelor se efectuează conform schemei redate de **Diagrama 2.7.1**. Este necesară selectarea înregistrării marcată ca eliminată și efectuarea unui click pe *Restabilire*. Sistemul elimină marcarea înregistrării și salvează acest fapt în baza de date. Astfel, înregistrarea este restabilită.



**Diagrama 2.7.1 Restabilirea datelor**

## Administrarea Utilizatorilor



**Diagrama 2.8.1 Administrarea utilizatorilor**

Modul de administrare a utilizatorilor de către utilizatorii cu rol de administrator este ilustrat în **Diagrama 2.8.1**. Astfel, utilizatorul are posibilitatea de adăugare, eliminare, modificare și modificare în grup a utilizatorilor. De asemenea, este posibilă alocarea de regiuni utilizatorilor cu rol de registrator.

## Adăugarea utilizatorilor

Adăugarea utilizatorilor se efectuează de către administrator conform schemei redate de **Diagrama 2.9.1**. Inițial, este necesară navigarea la *Utilizator Nou* sau la *Lista utilizatorilor*, urmată de efectuarea unui click pe +. Se continuă cu introducerea datelor despre utilizator. În continuare, administratorul are două posibilități: să salveze utilizatorul nou sau să anuleze operația de adăugare. În caz de anulare înregistrarea se anulează, iar în caz de salvare se continuă cu validarea datelor. Dacă datele sunt valide acestea sunt salvate în baza de date, în caz contrar se afișează mesajul de eroare. La final se revine la lista utilizatorilor.



**Diagrama 2.9.1 Adăugarea utilizatorilor**

## Alocarea regiunilor

Alocarea regiunilor se efectuează de către administrator conform schemei redate de **Diagrama 2.10.1**. Inițial, este necesară navigarea la *Lista utilizatorilor*, selectarea utilizatorului și efectuarea unui click pe *Alocare Regiuni*. Se continuă cu alocarea regiunilor în fereastra pentru alocare oferită de sistem și executarea operației de alocare. Astfel, regiunile sunt alocate utilizatorului respectiv, acesta având acces la ele. În final, se revine la lista utilizatorilor.



**Diagrama 2.10.1 Alocarea regiunilor**

## Eliminarea utilizatorilor

Eliminarea utilizatorilor se efectuează de către administrator conform schemei redate de **Diagrama 2.11.1**. Inițial, este necesară navigarea la *Lista utilizatorilor*, selectarea utilizatorului și efectuarea unui click pe *Delete*. Se continuă cu un mesaj de confirmare.



**Diagrama 2.11.1 Eliminarea utilizatorilor**

Dacă administratorul confirmă operația de eliminare se purcede la eliminarea utilizatorului, în caz contrar operația de eliminare se anulează. În final, se revine la lista utilizatorilor.

## Modificarea în grup a utilizatorilor

Modificarea în grup a utilizatorilor se efectuează de către administrator conform schemei redate de **Diagrama 2.12.1**. Inițial, este necesară navigarea la *Lista utilizatorilor*, selectarea multiplă a utilizatorilor și efectuarea unui click pe *Modificare*. Sistemul oferă un dialog pentru editare. După editarea câmpurilor sunt posibile două acțiuni: executarea modificării și închiderea dialogului. În cazul executării modificării, modificările efectuate sunt salvate, în caz contrar ele sunt anulate. În final, se revine la lista utilizatorilor.



**Diagrama 2.12.1 Modificarea în grup a utilizatorilor**

## Modificarea utilizatorilor

Modificarea utilizatorilor se efectuează de către administrator conform schemei redate de **Diagrama 2.13.1**. Inițial, este necesară navigarea la *Lista utilizatorilor*, selectarea utilizatorului și efectuarea unui click pe *Modifică*. Sistemul oferă un dialog pentru editare. După editarea câmpurilor sunt posibile două acțiuni: executarea modificării și închiderea dialogului. În cazul executării modificării, modificările efectuate sunt salvate, în caz contrar ele sunt anulate. În final, se revine la lista utilizatorilor.



**Diagrama 2.13.1 Modificarea utilizatorilor**

## Gestionarea Alegătorilor



**Diagrama 2.14.1 Gestionarea alegătorilor**

Modul de gestionare a alegătorilor de către utilizatorii cu rol de registrator este ilustrat în **Diagrama 2.14.1**. Astfel, utilizatorul are posibilitatea de modificare a statutului, înregistrare a declarației de ședere, modificare a adresei și modificare a secției de votare.

## Depunerea declarației de ședere

Depunerea declarației de ședere se efectuează conform schemei redate de **Diagrama 2.15.1**. Astfel, cetățeanul cu drept de vot depune inițial o cerere în acest sens. Registratorul preia cererea, o analizează și înregistrează declarația de ședere în sistem. La final, registratorul returnează un extras cetățeanului.



**Diagrama 2.15.1 Depunerea declarației de ședere**

## Înregistrarea declarației de ședere

Înregistrarea declarației de ședere se efectuează de către registrator conform schemei redate de **Diagrama 2.16.1**. Este necesară navigarea la lista alegătorilor, căutarea cetățeanului, efectuarea unui click pe *Declarație de ședere* și introducerea datelor. În continuare, sunt două posibilități: salvarea datelor și anularea operației de înregistrare a declarației de ședere. În caz de anulare datele se pierd, iar în caz contrar se validează datele introduse. Dacă informația este validă aceasta este salvată în baza de date, în caz contrar se afișează mesajul de eroare. În final, se revine la lista alegătorilor.



**Diagrama 2.16.1 Înregistrarea declarației de ședere**

## Modificarea statutului alegătorului

Modificarea statutului alegătorului se efectuează de către utilizator conform schemei redate de **Diagrama 2.17.1**. Este necesară navigarea la lista alegătorilor, selectarea alegătorului, efectuarea unui click pe *Modificare Statut* și introducerea datelor. În continuare, sunt două posibilități: salvarea datelor și anularea operației de modificare a statutului. În caz de anulare datele se pierd, iar în caz contrar datele sunt salvate în baza de date. În final, se revine la lista alegătorilor.



**Diagrama 2.17.1 Modificarea statutului alegătorului**

## Gestionarea Profilului utilizatorilor

Modul de gestionare a profilului utilizatorilor este ilustrat în **Diagrama 2.18.1**. Astfel, utilizatorii cu rol de registrator au posibilitatea de gestionare a profilului personal, inclusiv modificarea parolei, iar utilizatorii cu rol de administrator adițional pot gestiona profilul oricărui utilizator, inclusiv resetarea parolei.



**Diagrama 2.18.1 Gestionarea profilului utilizatorilor**

## Gestionarea profilului personal Și Modificarea Parolei

Gestionarea profilului personal și modificarea parolei se efectuează de către utilizator conform schemei redate de **Diagrama 2.19.1**. Este necesară navigarea la profilul personal, urmată de selectarea operației de modificare a datelor personale sau de modificare a parolei.

În cazul operației de modificare a datelor personale utilizatorului i se cere editarea acestora, urmată de două opțiuni posibile: salvarea modificărilor și anularea lor. Dacă se alege anularea modificărilor, modificările sunt ignorate și se revine la pagina precedentă, în caz contrar se purcede la validarea datelor. Dacă datele sunt valide se continuă cu salvarea acestora în baza de date, în caz contrar se afișează mesajul de eroare.

În cazul operației de modificare a parolei utilizatorului i se cere parola curentă și parola nouă, urmate de două opțiuni posibile: salvarea modificărilor și anularea lor. Dacă se alege anularea modificărilor, modificările sunt ignorate, în caz contrar se purcede la validarea datelor. Dacă datele sunt valide se continuă cu salvarea lor în baza de date, în caz contrar se afișează mesajul de eroare.



**Diagrama 2.19.1 Gestionarea profilului personal și modificarea parolei**

## Gestionarea profilului Utilizatorului

Gestionarea profilului utilizatorului se efectuează de către administrator conform schemei redate de **Diagrama 2.20.1**.



**Diagrama 2.20.1 Gestionarea profilului utilizatorului**

Este necesară navigarea la *Lista Utilizatorilor*, urmată de selectarea utilizatorului și modificarea parametrilor. În continuare, sunt posibile două opțiuni: salvarea modificărilor și anularea lor. Dacă se alege anularea modificărilor, modificările sunt ignorate, în caz contrar se purcede la validarea datelor. Dacă datele sunt valide se continuă cu salvarea acestora în baza de date și revenirea la *Lista Utilizatorilor*, în caz contrar se afișează mesajul de eroare și se revine la modificarea parametrilor.

## Resetarea parolei

Resetarea parolei se efectuează de către administrator conform schemei redate de **Diagrama 2.21.1**. Este necesară navigarea la *Lista Utilizatorilor*, urmată de selectarea utilizatorului, trecerea la profilul acestuia, selectarea *Resetează Parola* și modificarea parametrilor. În continuare, sunt posibile două opțiuni: salvarea modificărilor și anularea lor. Dacă se alege anularea modificărilor, modificările sunt ignorate, în caz contrar se purcede la validarea datelor. Dacă datele sunt valide se continuă cu salvarea acestora în baza de date și revenirea la profilul utilizatorului, în caz contrar se afișează mesajul de eroare și se revine la modificarea parolei.



**Diagrama 2.21.1 Resetarea parolei**

## Gestionarea Sectoarelor De Votare

Modul de gestionare a sectoarelor de votare este ilustrat în **Diagrama 2.22.1**. Astfel, utilizatorii au posibilitatea de adăugare și modificare a unui sector de votare.



**Diagrama 2.22.1 Gestionarea sectoarelor de votare**

## Adăugarea Sectoarelor De Votare

Adăugarea sectoarelor de votare se efectuează conform schemei redate de **Diagrama 2.23.1**. Este necesară selectarea regiunii, efectuarea unui click pe *Adaugă Secție* șicompletarea câmpurilor. Sunt două opțiuni posibile: salvarea secției de votare și anularea operației de adăugare. În caz de anulare datele introduse se pierd, în caz contrar se purcede la validarea datelor. Dacă datele sunt valide acestea sunt salvate în baza de date, în caz contrar se afișează mesajul de eroare și se revine la completarea câmpurilor.



**Diagrama 2.23.1 Adăugarea sectoarelor de votare**

## Modificarea Sectoarelor De Votare

Modificarea sectoarelor de votare se efectuează conform schemei redate de **Diagrama 2.24.1**. Este necesară selectarea sectorului de votare, efectuarea unui click pe *Modifică Înregistrare* șieditarea datelor. Sunt două opțiuni posibile: salvarea modificărilor și anularea operației de modificare. În caz de anulare sunt ignorate modificările și se revine la selectarea sectorului de votare, în caz contrar se purcede la validarea datelor. Dacă datele sunt valide acestea sunt salvate în baza de date, în caz contrar se afișează mesajul de eroare și se revine la modificarea datelor.



**Diagrama 2.24.1 Modificarea sectoarelor de votare**

## Generarea conflictelor

Generarea conflictelor se efectuează conform schemei redate de **Diagrama 2.25.1**. Astfel, inițial se lansează procesul de importare. Urmează unele etape de validare a datelor, descrise detaliat în continuare.

La prima etapă se verifică statutul. Dacă statutul nu e OK, se generează un conflict de statut care este transmis registratorului pentru a fi soluționat. Dacă statutul era OK sau conflictul de statut a fost soluționat de către registrator, se salvează și se trece la următoarea etapă.

La etapa a doua se verifică localitatea. Dacă localitatea nu există, se generează un conflict de lipsă localitate care este transmis administratorului pentru a fi soluționat. Dacă localitatea exista sau conflictul de lipsă localitate a fost soluționat de administrator, se trece la următoarea etapă.



**Diagrama 2.25.1 Generarea conflictelor**

La etapa a treia se verifică localitatea legată. Dacă aceasta există, se generează un conflict de localitate care este transmis registratorilor pentru a fi soluționat. Dacă localitatea legată nu exista se trece la următoarea etapă. Dacă conflictul de localitate a fost soluționat de registratori, se trece la etapa a șasea.

La etapa a patra se verifică dacă localitatea are adrese. Dacă nu, se trece la etapa a cincia, în caz contrar se trece la etapa a șasea.

La etapa a cincia se verifică dacă secția de votare nu e unică. Dacă răspunsul e afirmativ, se generează un conflict de secție de votare care este transmis spre soluționare registratorului. Dacă răspunsul era negativ sau conflictul de secție de votare a fost soluționat de registrator se salvează secția de votare și se trece la ultima etapă.

La etapa a șasea se verifică strada. Dacă codul străzii este zero, se generează un conflict StreetCode = 0, care este transmis spre soluționare administratorului. Dacă codul străzii era diferit de zero, se trece la etapa a șaptea. Dacă conflictul StreetCode = 0 a fost soluționat de administrator, se trece la etapa a opta.

La etapa a șaptea se verifică dacă strada există. În caz afirmativ, se trece la etapa a opta, în caz contrar se trece la etapa a noua.

La etapa a opta se verifică dacă adresa există. În caz negativ, se trece la etapa a noua, în caz contrar se trece la ultima etapă.

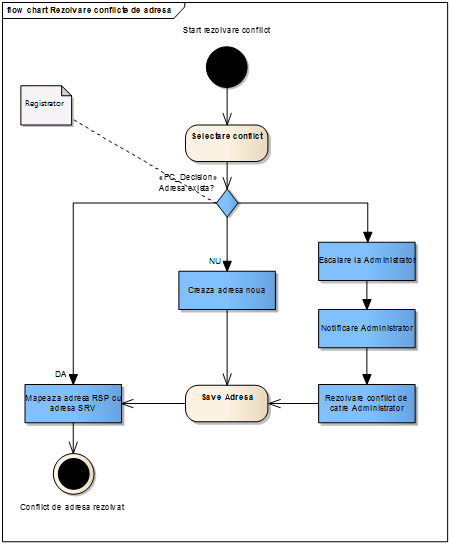
La etapa a noua se generează un conflict de adresă, care este transmis spre soluționare registratorului. Dacă conflictul de adresă este soluționat de registrator, se trece la ultima etapă.

La ultima etapă se actualizează datele personale, statutul, adresa și secția de votare.

## Rezolvarea conflictelor De Adresă

Rezolvarea conflictelor de adresă se efectuează de către registrator conform schemei redate de **Diagrama 2.26.1**. Astfel, inițial se selectează conflictul și se verifică dacă adresa există. Dacă adresa nu există, se creează și se salvează o adresă nouă. La final, se mapează adresa RSP cu adresa SRV.

Mai există o opțiune, acea de transmitere a conflictului la administrator în situația de imposibilitate a soluționării independente a conflictului. În acest caz, conflictul este escalat la administrator, care este notificat de acest fapt. Ulterior, administratorul soluționează conflictul, salvează adresa și mapează adresa RSP cu adresa SRV.

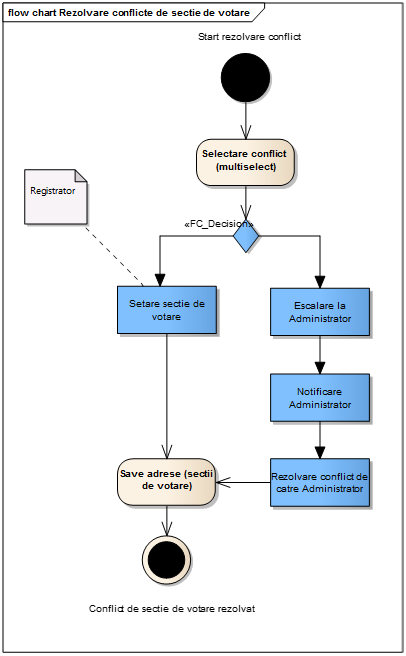


**Diagrama 2.26.1 Rezolvarea conflictelor de adresă**

## Rezolvarea conflictelor De Secție de Votare

Rezolvarea conflictelor de secție de votare se efectuează de către registrator conform schemei redate de **Diagrama 2.27.1**. Astfel, inițial se selectează conflictele. Pentru a soluționa aceste conflicte este necesară setarea în continuare a secției de votare și salvarea adreselor secțiilor de votare.

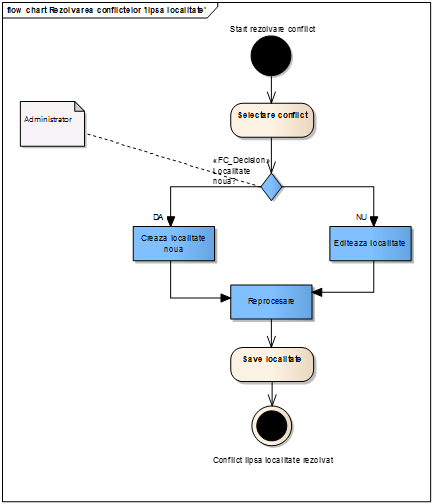
Mai există o opțiune, acea de transmitere a conflictelor la administrator în situația de imposibilitate a soluționării independente a lor. În acest caz, conflictele sunt escalate la administrator, care este notificat de acest fapt. Ulterior, administratorul soluționează conflictele și salvează adresele secțiilor de votare.



**Diagrama 2.27.1 Rezolvarea conflictelor de secție de votare**

## Rezolvarea conflictelor De Lipsă Localitate

Rezolvarea conflictelor de lipsă localitate se efectuează de către administrator conform schemei redate de **Diagrama 2.28.1**. Astfel, inițial se selectează conflictul. Sunt posibile două opțiuni: crearea unei localități noi și editarea unei localități. Administratorul decide opțiunea optimă, reieșind din cazul concret al conflictului. Ulterior, conflictul este reprocesat și localitatea este salvată. Astfel, conflictul de lipsă localitate este soluționat.

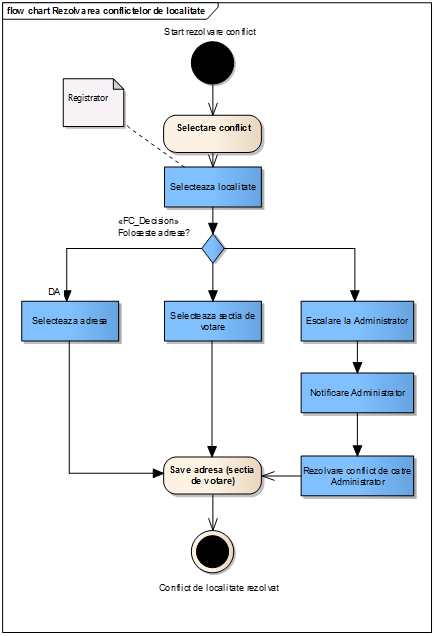


**Diagrama 2.28.1 Rezolvarea conflictelor de lipsă localitate**

## Rezolvarea conflictelor De Localitate

Rezolvarea conflictelor de localitate se efectuează de către registrator conform schemei redate de **Diagrama 2.29.1**. Astfel, inițial se selectează conflictul și localitatea. Dacă localitatea folosește adrese, se selectează adresa, în caz contrar se selectează secția de votare. La final, se salvează adresa secției de votare. În acest mod conflictul de localitate este soluționat.

Mai există o opțiune, acea de transmitere a conflictului la administrator în situația de imposibilitate a soluționării independente a lui. În acest caz, conflictul este escalat la administrator, care este notificat de acest fapt. Ulterior, administratorul soluționează conflictul și salvează adresa secției de votare.

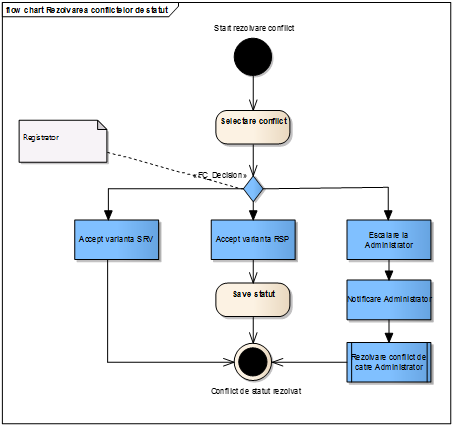


**Diagrama 2.29.1 Rezolvarea conflictelor de localitate**

## Rezolvarea conflictelor De Statut

Rezolvarea conflictelor de statut se efectuează de către registrator conform schemei redate de **Diagrama 2.30.1**. Astfel, inițial se selectează conflictul. Sunt două opțiuni posibile: acceptarea variantei SRV și acceptarea variantei RSP cu salvarea statutului. Registratorul decide opțiunea corectă, reieșind din cazul concret al conflictului.

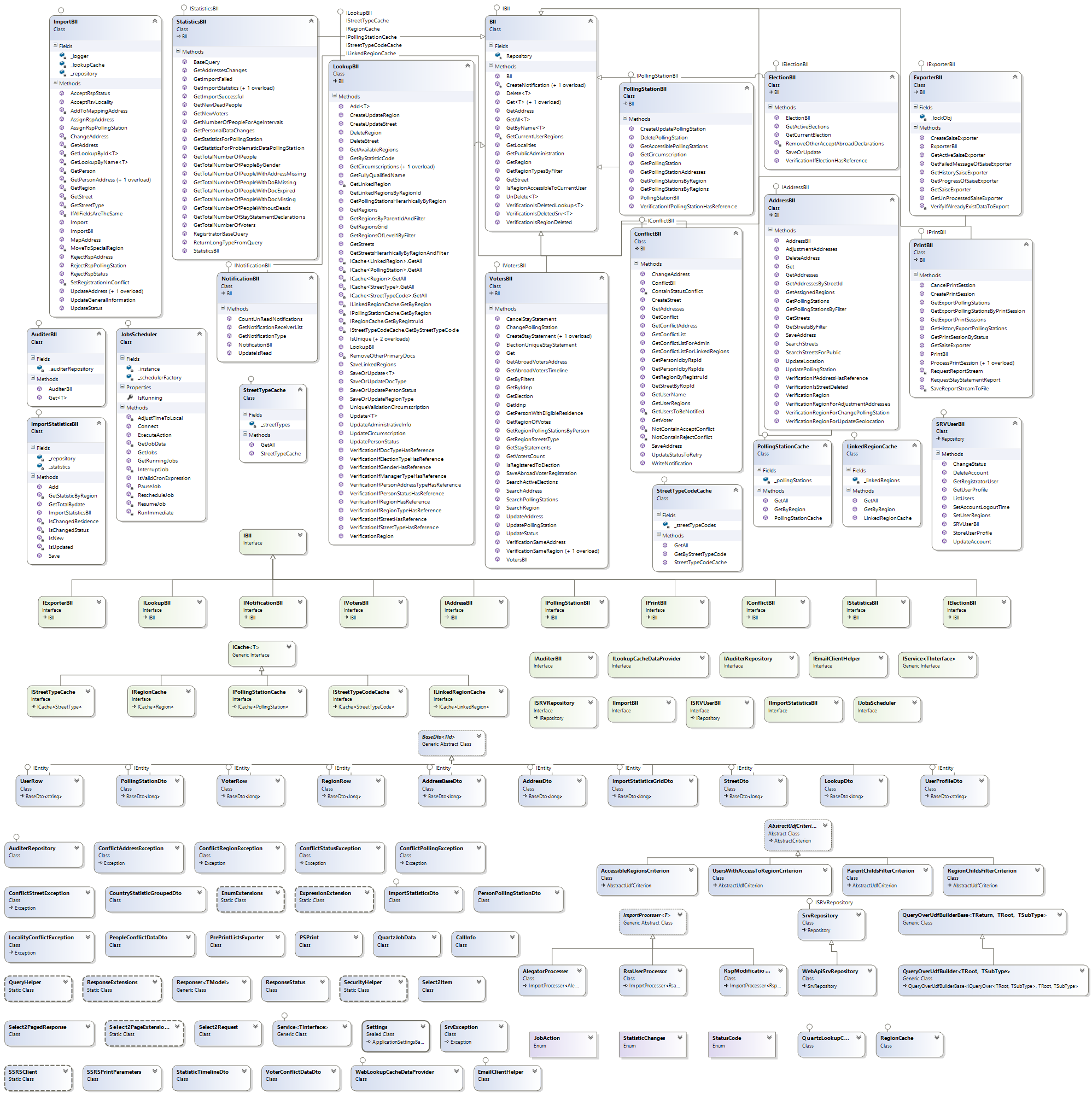
Mai există o opțiune, acea de transmitere a conflictului la administrator în situația de imposibilitate a soluționării independente a lui. În acest caz, conflictul este escalat la administrator, care este notificat de acest fapt. Ulterior, administratorul soluționează conflictul.



**Diagrama 2.30.1 Rezolvarea conflictelor de statut**

# Componenta CEC.SRV.BLL

Componenta BLL a sistemului CEC RSA este acea componentă care interacționează cu baza de date. Subcomponentele BLL sunt prezentate în următoarele subsecțiuni. Diagrama generală a claselor este prezentată în figura de mai jos.



## Subcomponenta BLL

Subcomponenta Bll conține funcționalitățile de bază ale tuturor subcomponentelor ce interacționează cu baza de date. Principalele funcționalități sunt descrise în tabelul de mai jos.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr. | Funcționalitate | Date de intrare | Date de ieșire | Descriere |
| 1. | public T Get<T>  (long id) | *T* – tipul generic al entității  *id* – numărul de identificare al entității | entitatea de tipul respectiv cu numărul de identificare respectiv | Returnează obiectul din baza de date de tipul respectiv și cu numărul de identificare respectiv. În caz de neexistență, returnează *null.* |
| 2. | public IList<T> GetAll<T>() | *T* – tipul generic al entității | lista de entități de tipul respectiv | Returnează lista tuturor obiectelor din baza de date de tipul respectiv. În caz de neexistență, se returnează o listă vidă. |
| 3. | public IList<Region> GetLocalities(int? nodeId) | *nodeId* – numărul de identificare al regiunii părinte | lista de subregiuni ale regiunii cu numărul de identificare respectiv, accesibile utilizatorului curent | Returnează lista de subregiuni ale regiunii cu numărul de identificare respectiv, accesibile utilizatorului curent. Dacă *nodeId* este *null*, atunci returnează lista de regiuni accesibile care sunt subregiuni ale unor regiuni neaccesibile pentru utilizatorul curent. |
| 4. | protected long[] GetCurrentUserRegions() |  | lista numerelor de identificare ale regiunilor accesibile utilizatorului curent (cu rol de registrator) | Returnează lista numerelor de identificare ale regiunilor accesibile utilizatorului curent (cu rol de registrator). |
| 5. | public PageResponse<T> Get<T>(PageRequest pageRequest) | *T* – tipul generic al entității, *pageRequest –* informații despre pagina cerută de la server (*PageNumber –* numărulpaginii, *PageSize* – numărul de elemente pe pagină, etc.) | pagina respectivă, încărcată cu o listă de entități (*Items*), mărimea paginii (*PageSize*), indicele primului element (*StartIndex*) și numărul total de elemente (*Total*) | Returnează o pagină încărcată cu elemente de tipul respectiv, conform cererii efectuate. |
| 6. | public void Delete<T>(long id) | *T* – tipul generic al entității,  *id –* numărul de identificare al entității |  | Elimină entitatea de tipul respectiv cu numărul de identificare respectiv din baza de date. |
| 7. | public void UnDelete<T>(long entityId) | *T* – tipul generic al entității,  *entityId –* numărul de identificare al entității |  | Restabilește entitatea de tipul respectiv cu numărul de identificare respectiv care a fost eliminată anterior din baza de date. |
| 8. | public IList<RegionType> GetRegionTypesByFilter(long regionId) | *regionId* – numărul de identificare al regiunii | lista tipurilor de regiune cu *Rank* mai mare decât *Rank*-ul tipului de regiune al regiunii cu numărul de identificare respectiv | Returnează lista tipurilor de regiune cu *Rank* mai mare decât *Rank*-ul tipului de regiune al regiunii cu numărul de identificare respectiv. |
| 9. | public PublicAdministration GetPublicAdministration(long regionId) | *regionId* – numărul de identificare al regiunii | administrația publică asociată regiunii cu numărul de identificare respectiv | Returnează administrația publică asociată regiunii cu numărul de identificare respectiv. |
| 10. | public Street GetStreet(string streetsName, long regionId, long streetTypeId) | *streetsName* – denumirea străzii,  *regionId* – numărul de identificare al regiunii,  *streetTypeId* – numărul de identificare al tipului de stradă | strada cu denumirea respectivă și tipul respectiv din regiunea dată. | Returnează strada cu denumirea respectivă și tipul respectiv din regiunea dată. |
| 11. | public IList<Region> GetRegion(string name, long regionTypeId, long? parentId) | *name* – denumirea regiunii, *regionTypeId* – numărul de identificare al tipului de regiune,  *parentId* – numărul de identificare al regiunii părinte. | lista de regiuni cu denumirea și tipul respectiv și numărul de identificare al regiunii părinte corespunzător | Returnează lista de regiuni cu denumirea și tipul respectiv și numărul de identificare al regiunii părinte corespunzător. |
| 12. | public IList<Address> GetAddress(long streetsId, long? houseNumber, string suffix) | *streetsId* – numărul de identificare al străzii,  *houseNumber* – numărul blocului,  *suffix* – sufixul numărului blocului | lista de adrese cu numărul și sufixul blocului respectiv și numărul de identificare al străzii corespunzător | Returnează lista de adrese cu numărul și sufixul blocului respectiv și numărul de identificare al străzii corespunzător. |
| 13. | public T GetByName<T>(string name) | *T* – tipul generic al entității,  *name* – denumirea entității | entitatea de tipul respectiv cu denumirea dată | Returnează entitatea de tipul respectiv cu denumirea dată. |
| 14. | public bool IsRegionAccessibleToCurrentUser(long currentRegionId) | *currentRegionId* – numărul de identificare al regiunii | *true* – dacă regiunea este accesibilă utilizatorului curent și *false* – în caz contrar | Verifică dacă regiunea este accesibilă utilizatorului curent. Dacă utilizatorul curent are rol de Administrator sau are acces la regiunea respectivă, se returnează *true*, în caz contrar – *false*. |
| 15. | public void VerificationIsRegionDeleted(long regionId) | *regionId* – numărul de identificare al regiunii |  | Se aruncă o excepție atunci când regiunea este deja eliminată. |
| 16. | protected void CreateNotification(EventTypes eventType, INotificationEntity entity, long entityId, string message) | *eventType* – tipul evenimentului;  *entity –* entitatea;  *entityId* – numărul de identificare al entității;  *message –* mesajul notificării |  | Creează o notificare cu un mesaj corespunzător și parametrii de intrare respectivi pentru toți utilizatorii cu rol de Administrator. |
| 17. | protected void CreateNotification(EventTypes eventType, INotificationEntity entity, long entityId, string message,  IEnumerable<SRVIdentityUser> users) | *eventType* – tipul evenimentului;  *entity –* entitatea;  *entityId* – numărul de identificare al entității;  *message –* mesajul notificării;  *users* – utilizatorii |  | Creează o notificare cu un mesaj corespunzător și parametrii de intrare respectivi pentru utilizatorii dați. |
| 18. | public void VerificationIsDeletedLookup<T>(long id) | *id* – numărul de identificare al entității |  | Se aruncă o excepție atunci când entitatea este deja eliminată. |
| 19. | public void SaveOrUpdate<T>  (T entity) | *T* – tipul generic al entității,  *entity –* entitatea |  | Dacă numărul de identificare al entității este zero, atunci entitatea este inserată în baza de date, în caz contrar - actualizată. |

## Subcomponenta Address Bll

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr. | Funcționalitate | Date de intrare | Date de ieșire | Descriere |
| 1. | public PageResponse<Address> Get(PageRequest pageRequest, long? regionId) | *pageRequest –* informații despre pagina cerută de la server (*PageNumber –* numărulpaginii, *PageSize* – numărul de elemente pe pagină, etc.);  *regionId* – numărul de identificare al regiunii | pagina respectivă, încărcată cu o listă de adrese (*Items*), mărimea paginii (*PageSize*), indicele primului element (*StartIndex*) și numărul total de elemente (*Total*) | Returnează o pagină încărcată cu adrese din regiunea respectivă, conform cererii efectuate. Dacă *regionId* este *null* sau regiunea dată nu este este accesibilă utilizatorului curent, se returnează o pagină vidă. |
| 2. | public IEnumerable<Street> GetStreets(long? regionId) | *regionId* – numărul de identificare al regiunii | lista de străzi din regiunea respectivă | Dacă *regionId* e nenul și corespunde unei regiuni accesibile, se returnează lista de străzi din regiunea respectivă. Dacă *regionId* e nenul și corespunde unei regiuni neaccesibile, se returnează o listă vidă de străzi. Dacă *regionId* e nul, se returnează lista tuturor străzilor. |
| 3. | public PageResponse<Street> SearchStreets(PageRequest pageRequest, long? regionId) | *pageRequest –* informații despre pagina cerută de la server (*PageNumber –* numărulpaginii, *PageSize* – numărul de elemente pe pagină, etc.);  *regionId* – numărul de identificare al regiunii | pagina cerută încărcată cu străzi din regiunea respectivă | Dacă *regionId* e nenul și corespunde unei regiuni accesibile, se returnează pagina corespunzătoare de străzi din regiunea respectivă. Dacă *regionId* e nenul și corespunde unei regiuni neaccesibile, se returnează o pagină încărcată cu o listă vidă de străzi. Dacă *regionId* e nul, se returnează pagina respectivă din lista tuturor străzilor. |
| 4. | public PageResponse<Street> SearchStreetsForPublic(PageRequest pageRequest, long? regionId) | *pageRequest –* informații despre pagina cerută de la server (*PageNumber –* numărulpaginii, *PageSize* – numărul de elemente pe pagină, etc.);  *regionId* – numărul de identificare al regiunii | pagina cerută încărcată cu străzi din regiunea respectivă | Dacă *regionId* e nenul, se returnează pagina corespunzătoare de străzi din regiunea respectivă. Dacă *regionId* e nul, se returnează pagina respectivă din lista tuturor străzilor. |
| 5. | public IEnumerable<Street> GetStreetsByFilter(string term) | *term* – o expresie care se conține în denumirea străzilor căutate | lista de străzi denumirea cărora conține expresia *term*. | Returnează lista de străzi denumirea cărora conține expresia *term*. |
| 6. | public void SaveAddress(long addressId, long streetId, int? houseNumber, string suffix, BuildingTypes buildingType, long? pollingStationId) | *addressId* – numărul de identificare al adresei;  *streetId* – numărul de identificare al străzii;  *houseNumber* – numărul blocului;  *suffix* – sufixul numărului blocului;  *buildingType* – tipul blocului (*Admin*istrativ */ Locativ*);  *pollingStationId* – numărul de identificare al secției de votare |  | Salvează adresa cu parametrii respectivi în baza de date. Dacă deja există o altă adresă cu aceeași parametri, este aruncată o excepție. |
| 7. | public void UpdateLocation(long addressId, double latitude, double longitude) | *addressId* – numărul de identificare al adresei;  *latitude* – latitudinea locației;  *longitude* – longitudinea locației |  | Reactualizează locația adresei cu datele de intrare. |
| 8. | public IEnumerable<PollingStation> GetPollingStations(long? regionId) | *regionId* – numărul de identificare al regiunii | lista de secții de votare din regiunea respectivă | Dacă *regionId* e nenul și corespunde unei regiuni accesibile, se returnează lista de secții de votare din regiunea respectivă. În celelalte cazuri este returnată o listă vidă de secții de votare. |
| 9. | public IEnumerable<PollingStation> GetPollingStationsByFilter(string term) | *term* – o expresie cu care începe numărul secțiilor de votare căutate | lista de secții de votare numărul cărora conține expresia începe cu *term*. | Returnează lista de secții de votare numărul cărora conține expresia începe cu *term*. |
| 10. | public IEnumerable<Address> GetAddressesByStreetId(long streetId) | *streetId* – numărul de identificare al străzii | lista de adrese de pe strada respectivă | Returnează lista de adrese de pe strada respectivă |
| 11. | public void DeleteAddress(long addressId) | *addressId* – numărul de identificare al adresei |  | Elimină adresa respectivă din baza de date. Dacă există o secție de votare sau o adresă de persoană asociată adresei, este aruncată o excepție. |
| 12. | public void VerificationRegion(long id, long? addressId) | *id –* numărul de identificare al regiunii;  *addressId* – numărul de identificare al adresei |  | Dacă regiunea nu are străzi sau este eliminată este aruncată o excepție. |
| 13. | public void UpdatePollingStation(long pollingStationId, IEnumerable<long> addressIds) | *pollingStationId* – numărul de identificare al secției de votare;  *addressIds* – numerele de identificare ale adreselor |  | Asignează adresele la secția de votare dată. Aruncă o excepție dacă secția de votare cu numărul de identificare respectiv nu există. |
| 14. | public PageResponse<AddressDto> GetAddresses(PageRequest pageRequest, long regionId) | *pageRequest* - pagina cerută;  *regionId* – numărul de identificare al regiunii; | pagina cerută cu extrase de informații despre adresele din regiunea respectivă | Returnează pagina cerută cu extrase de informații despre adresele din regiunea respectivă |
| 15. | public void VerificationRegionForChangePollingStation(long id) | *id* – numărul de identificare al regiunii |  | Dacă regiunea nu are străzi sau este eliminată este aruncată o excepție. |
| 16. | public void VerificationRegionForUpdateGeolocation(long id) | *id* – numărul de identificare al regiunii |  | Dacă regiunea nu are străzi sau este eliminată este aruncată o excepție. |
| 17. | public void VerificationRegionForAdjustmentAddresses(long id) | *id* – numărul de identificare al regiunii |  | Dacă regiunea nu are străzi sau este eliminată este aruncată o excepție. |
| 18. | public void VerificationIsStreetDeleted(long streetId) | *streetId* – numărul de identificare al străzii |  | Dacă strada este eliminată este aruncată o excepție. |
| 19. | public void AdjustmentAddresses(long oldAddressId, long newAddressId) | *oldAddressId* – numărul de identificare al adresei vechi;  *newAddressId* – numărul de identificare al adresei noi. |  | Toate adresele de persoane care au numărul de identificare *oldAddressId* sunt modificate cu adresa cu numărul de identificare *newAddressId.* Ulterior, adresa veche este eliminată. Dacă strada vechii adrese nu are adrese, aceasta este eliminată. De asemenea, sunt eliminate și alte entități deja neutilizabile. |
| 20. | public void VerificationIfAddressHasReference(long addressId) | *addressId* – numărul de identificare al adresei |  | Dacă există o adresă de persoană sau o secție de votare asociată adresei respective, se aruncă o excepție. |
| 21. | public IList<Region> GetAssignedRegions() |  | lista de regiuni accesibile utilizatorului curent | Returnează lista de regiuni accesibile utilizatorului curent. |
| 22. | public void DeleteAddressMappings(long addressId) | *addressId* – numărul de identificar al adresei |  | Elimină mapările de adrese asociate adresei respective. |
| 23. | public PageResponse<RspAddressMapping> ListAddressMappings(PageRequest pageRequest) | *pageRequest –* informații despre pagina cerută de la server (*PageNumber –* numărulpaginii, *PageSize* – numărul de elemente pe pagină, etc.) | pagina cerută încărcată cu mapări de adrese | Returnează lista de mapări de adrese accesibile utilizatorului autentificat. |
| 24. | public void DeleteAddressMapping(long addressMappingId) | *addressMappingId* – numărul de identificare al mapării de adrese |  | Elimină maparea de adrese respectivă. |

## Subcomponenta Auditer Bll

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr. | Funcționalitate | Date de intrare | Date de ieșire | Descriere |
| 1. | public PageResponse<T> Get<T>(PageRequest request, long enityId) | *T* – tipul generic al entității;  *pageRequest –* informații despre pagina cerută de la server (*PageNumber –* numărulpaginii, *PageSize* – numărul de elemente pe pagină, etc.);  *entityId* – numărul de identificare al entității căreia i se face audit | pagina respectivă, încărcată cu o listă de adrese (*Items*), mărimea paginii (*PageSize*), indicele primului element (*StartIndex*) și numărul total de elemente (*Total*) | Returnează o pagină încărcată cu entități cu numărul de identificare respectiv, conform cererii efectuate. |

## Subcomponenta Conflict Bll

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr. | Funcționalitate | Date de intrare | Date de ieșire | Descriere |
| 1. | public PageResponse<RspConflictData> GetConflictList(PageRequest pageRequest, ConflictStatusCode conflictCode) | *pageRequest –* informații despre pagina cerută de la server (*PageNumber –* numărulpaginii, *PageSize* – numărul de elemente pe pagină, etc.);  *conflictCode* – codul statutului conflictului | pagina respectivă, încărcată cu o listă de date despre conflicte (*Items*), mărimea paginii (*PageSize*), indicele primului element (*StartIndex*) și numărul total de elemente (*Total*) | Returnează o pagină încărcată cu date despre conflicte cu codul de statut al conflictului respectiv, conform cererii efectuate. |
| 2. | public PageResponse<RspConflictData> GetConflictListForLinkedRegions(PageRequest pageRequest, ConflictStatusCode conflictCode) | *pageRequest –* informații despre pagina cerută de la server (*PageNumber –* numărulpaginii, *PageSize* – numărul de elemente pe pagină, etc.);  *conflictCode* – codul statutului conflictului | pagina respectivă, încărcată cu o listă de date despre conflicte pentru regiunile asociate (*Items*), mărimea paginii (*PageSize*), indicele primului element (*StartIndex*) și numărul total de elemente (*Total*) | Returnează o pagină încărcată cu date despre conflicte cu codul de statut al conflictului respectiv, conform cererii efectuate. |
| 3. | public PageResponse<RspConflictDataAdmin> GetConflictListForAdmin(PageRequest pageRequest, ConflictStatusCode conflictCode) | *pageRequest –* informații despre pagina cerută de la server (*PageNumber –* numărulpaginii, *PageSize* – numărul de elemente pe pagină, etc.);  *conflictCode* – codul statutului conflictului | pagina respectivă, încărcată cu o listă de date despre conflicte pentru administrator (*Items*), mărimea paginii (*PageSize*), indicele primului element (*StartIndex*) și numărul total de elemente (*Total*) | Returnează o pagină încărcată cu date despre conflicte pentru administrator cu codul de statut al conflictului respectiv, conform cererii efectuate. |
| 4. | public PageResponse<AddressBaseDto> GetAddresses(PageRequest pageRequest, long? regionId) | *pageRequest –* informații despre pagina cerută de la server (*PageNumber –* numărulpaginii, *PageSize* – numărul de elemente pe pagină, etc.);  *regionId* – numărul de identificare al regiunii | pagina respectivă, încărcată cu o listă de date despre adrese (*Items*), mărimea paginii (*PageSize*), indicele primului element (*StartIndex*) și numărul total de elemente (*Total*) | Returnează o pagină încărcată cu date despre adrese cu codul de statut al conflictului respectiv, conform cererii efectuate. |
| 5. | public VoterConflictDataDto GetVoter(string idnp) | *idnp –* numărul de identificare al alegătorului | informații despre conflictul referitor alegătorului | Returnează informații despre conflictul referitor alegătorului. |
| 6. | public Region GetRegionByAdministrativeCode(long cuatmCode) | *cuatmCode –* numărul de identificare al regiunii în Registru | regiunea cu numărul de identificare respectiv în Registru | Returnează regiunea cu numărul de identificare respectiv în Registru. |
| 7. | public Street GetStreetByRopId(long streetCodeId) | *streetCodeId* – numărul de identificare al codului străzii | Strada cu numărul de identificare *RopId* respectiv | Returnează strada cu numărul de identificare *RopId* respectiv. |
| 8. | public long CreateStreet(string name, string description, long regionId, long streetTypeId, long? ropId, long? saiseId) | *name* – denumirea străzii;  *description* – descrierea străzii;  *regionId* – numărul de identificare al regiunii;  *streetTypeId* – numărul de identificare al tipului de stradă;  *ropId* – numărul de identificare ROP al străzii;  *saiseId* – numărul de identificare al străzii în aplicația SAISE. | Numărul de identificare al străzii create | Creează o stradă nouă cu parametrii respectivi și returnează numărul de identificare atribuit. |
| 9. | public List<long> GetPersonIdbyRspIds(List<long> conflictIds) | *conflictIds –* lista numerelor de identificare ale conflictelor de persoane | Lista numerelor de identificare ale persoanelor aflate în conflict | Returnează lista numerelor de identificare ale persoanelor aflate în conflict. În caz de neexistență a unei persoane cu numărul de identificare în lista dată, se aruncă o excepție. |
| 10. | public long GetPersonIdByConflictId(long conflictId) | *conflictId* – numărul de identificare al conflictului de persoană | Numărul de identificare al persoanei aflate în conflict | Returnează numărul de identificare al persoanei aflate în conflict. În caz de neexistență, este aruncată o excepție. |
| 11. | public void WriteNotification(ConflictStatusCode conflictStatus, RspModificationData conflictData, string notificationMessage) | *conflictStatus* – statutul conflictului;  *conflictData* – informații despre conflict;  *notificationMessage* – mesajul de notificare |  | Scrie un mesaj de notificare a conflictului cu parametrii respectivi. |
| 12. | public Address SaveAddress(long streetId, int? houseNumber, string suffix, BuildingTypes buildingType, long? pollingStationId) | *streetId* – numărul de identificare al străzii;  *houseNumber* – numărul apartamentului;  *suffix* – sufixul numărului apartamentului;  *buildingType* – tipul apartamentului;  *pollingStationId* – numărul de identificare al secției de votare | Adresa salvată | Creează o adresă nouă în baza informației date. Dacă adresa deja există, se aruncă o excepție. |
| 13. | public void UpdateStatusToRetry(long conflictId, string message, ConflictStatusCode conflictCode) | *conflictId* – numărul de identificare al conflictului;  *message –* mesajul conflictului;  *conflictCode* – codul statutului conflictului |  | Actualizează statutul conflictului pentru a indica posibilitate de a reîncerca ulterior soluționarea acestuia. |
| 14. | public RspModificationData GetConflict(long id) | *id* – numărul de identificare al conflictului | Informații despre conflictul cu numărul de identificare respectiv | Returnează informații despre conflictul cu numărul de identificare respectiv. |
| 15. | public string GetUserName() |  | Numele utilizatorului curent | Returnează numele utilizatorului curent. |
| 16. | public PageResponse<Region> GetUserRegions(PageRequest pageRequest) | *pageRequest –* informații despre pagina cerută de la server (*PageNumber –* numărulpaginii, *PageSize* – numărul de elemente pe pagină, etc.); | pagina respectivă, încărcată cu o listă de regiuni accesibile utilizatorului curent (*Items*), mărimea paginii (*PageSize*), indicele primului element (*StartIndex*) și numărul total de elemente (*Total*) | Returnează o pagină încărcată cu regiuni accesibile utilizatorului curent, conform cererii efectuate. |
| 17. | public void ChangeAddress(long addressId, long personId) | *addressId* – numărul de identificare al adresei;  *personId* – numărul de identificare al persoanei |  | Modifică adresa unde va vota persoana respectivă. |
| 18. | public RspRegistrationData GetConflictAddress(long conflictId) | *conflictId* – numărul de indentificare al conflictului | Informații despre conflictul de adresă cu numărul de identificare respectiv | Returnează informații despre conflictul de adresă cu numărul de identificare respectiv. |
| 19. | public PageResponse<RspConflictDataAdmin> GetConflictListForAdmin(PageRequest pageRequest, ConflictStatusCode[] conflictCodes) | *pageRequest –* informații despre pagina cerută de la server (*PageNumber –* numărulpaginii, *PageSize* – numărul de elemente pe pagină, etc.);  *conflictCodes* – lista de coduri ale statutelor conflictelor | pagina respectivă, încărcată cu o listă de date despre conflicte pentru administrator (*Items*), mărimea paginii (*PageSize*), indicele primului element (*StartIndex*) și numărul total de elemente (*Total*) | Returnează o pagină încărcată cu date despre conflicte pentru administrator cu codurile de statut ale conflictelor respective, conform cererii efectuate. |

## Subcomponenta Election Bll

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr. | Funcționalitate | Date de intrare | Date de ieșire | Descriere |
| 1. | public void SaveOrUpdate(long electionId, long electionTypeId, DateTime electionDate, long saiseId, string comments, bool acceptAbroadDeclaration) | *electionId –* numărul de identificare al scrutinului electoral;  *electionTypeId* – numărul de identificare al tipului scrutinului electoral;  *electionDate* – data scrutinului electoral;  *saiseId* – numărul de identificare al scrutinului electoral în aplicația SAISE;  *comments* – comentarii asociate scrutinului electoral;  *acceptAbroadDeclaration* – indică faptul dacă declarațiile din străinătate sunt acceptate |  | Dacă numărul de identificare al scrutinului electoral este zero se creează și se salvează scrutinul electoral conform informației transmise ca parametru. În caz contrar, se actualizează scrutinul electoral cu numărul de identificare respectiv conform informației transmise ca parametru. |
| 2. | public IList<Election> GetActiveElections() |  | lista de scrutine electorale active | Returnează lista de scrutine electorale active |
| 3. | public Election GetCurrentElection() |  | lista de scrutine electorale active care acceptă declarații externe. | returnează lista de scrutine electorale active care acceptă declarații externe. |
| 4. | public void VerificationIfElectionHasReference(long electionId) | *electionId* – numărul de identificare al scrutinului electoral |  | verifică dacă scrutinul respectiv are vreo legătură cu alte entități. |

## Subcomponenta Exporter Bll

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr. | Funcționalitate | Date de intrare | Date de ieșire | Descriere |
| 1. | public void CreateSaiseExporter(long electionId, bool exportAllVoters) | *electionId –* numărul de identificare al scrutinului electoral;  *exportAllVoters* – indică faptul dacă se exportă toți alegătorii |  | Creează un proces de exportare către SAISE pentru scrutinul electoral respectiv. |
| 2. | public PageResponse<SaiseExporterStage> GetSaiseExporter(PageRequest pageRequest,long? saiseExporterId) | *pageRequest –* informații despre pagina cerută de la server (*PageNumber –* numărulpaginii, *PageSize* – numărul de elemente pe pagină, etc.);  *saiseExporterId* – numărul de identificare al procesului de exportare către SAISE | pagina respectivă, încărcată cu o listă de etape ale procesului de exportare respectiv (*Items*), mărimea paginii (*PageSize*), indicele primului element (*StartIndex*) și numărul total de elemente (*Total*) | Returnează o pagină încărcată cu o listă de etape ale procesului de exportare respectiv, conform cererii efectuate. Dacă numărul de identificare al procesului de exportare către SAISE este *null* se returnează toate etapele asociate proceselor de exportare noi sau în progres. |
| 3. | public PageResponse<SaiseExporterStage> GetHistorySaiseExporter(PageRequest pageRequest, long? saiseExporterId) | *pageRequest –* informații despre pagina cerută de la server (*PageNumber –* numărulpaginii, *PageSize* – numărul de elemente pe pagină, etc.);  *saiseExporterId* – numărul de identificare al procesului de exportare către SAISE | pagina respectivă, încărcată cu o listă de etape ale procesului de exportare respectiv (*Items*), mărimea paginii (*PageSize*), indicele primului element (*StartIndex*) și numărul total de elemente (*Total*) | Returnează o pagină încărcată cu o listă de etape ale procesului de exportare respectiv, conform cererii efectuate. |
| 4. | public IEnumerable<SaiseExporter> GetUnProcessedSaiseExporter() |  | Lista de procese de exportare noi sau în progres. | Returnează lista de procese de exportare noi sau în progres. |
| 5. | public SaiseExporter GetActiveSaiseExporter() |  | Un proces de exportare nou sau în progres | Returnează un proces de exportare nou sau în progres. |
| 6. | public int GetProgressOfSaiseExporter(long saiseExporterId) | *saiseExporterId* – numărul de identificare al procesului de exportare către SAISE | Progresul în procente al procesului de exportare | Returnează progresul în procente al procesului de exportare. |
| 7. | public bool GetFailedMessageOfSaiseExporter(long saiseExporterId) | *saiseExporterId* – numărul de identificare al procesului de exportare către SAISE | Existența unei etape finisată cu eroare a procesului de exportare cu numărul de identificare respectiv. | Indică existența unei etape finisată cu eroare a procesului de exportare cu numărul de identificare respectiv. |

## Subcomponenta Import Bll

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr. | Funcționalitate | Date de intrare | Date de ieșire | Descriere |
| 1. | public Person Import(RspModificationData rawData, ref StatisticChanges status) | *rawData* – informații despre conflict | persoana cu datele actualizate în urma operației de importare;  *status –* statutul operației de importare; | Returnează persoana cu datele actualizate în urma operației de importare și setează statutul. De asemenea, în caz de apariție a unui conflict, acesta este jurnalizat. |
| 2. | public void AcceptRspStatus(RspModificationData data) | *data* – informații despre conflict |  | Este acceptat statutul persoanei din RSP. |
| 3. | public void RejectRspStatus(RspModificationData data) | *data* – informații despre conflict |  | Nu este acceptat statutul persoanei din RSP. |
| 4. | public void MapAddress(RspModificationData rawData, Address address, int appNr, string apSuffix) | *rawData* – informații despre conflict;  *address* – adresa care trebuie asociată;  *appNr* – numărul apartamentului;  *apSuffix* – sufixul numărului apartamentului |  | se asociază adresa respectivă persoanei corespunzătoare informației din *rawData*. |
| 5. | public void AssignRspAddress(RspModificationData data) | *data* – informații despre conflict |  | se asociază adresa din RSP persoanei corespunzătoare informației din *rawData*. |
| 6. | public void RejectRspAddress(RspModificationData data) | *data* – informații despre conflict |  | se renunță la adresa din RSP a persoanei corespunzătoare informației din *rawData*. |
| 7. | public void AssignRspPollingStation(RspModificationData data) | *data* – informații despre conflict |  | se asociază secția de votare din RSP persoanei corespunzătoare informației din *rawData*. |
| 8. | public void RejectRspPollingStation(RspModificationData data) | *data* – informații despre conflict |  | se renunță la secția de votare din RSP a persoanei corespunzătoare informației din *rawData*. |
| 9. | public void AcceptRsvLocality(long conflictId) | *conflictId* – numărul de identificare al conflictului |  | Se acceptă localitatea din RSV persoanei corespunzătoare conflictului cu numărul de identificare respectiv. |
| 10. | public void UpdateGeneralInformation(RspModificationData rawData, Person person, ref StatisticChanges status) | *rawData* – informații despre conflict;  *person* – persoana dată | *status –* statutul operației de actualizare a informației generale | Se actualizează informația generală despre persoană și se setează statutul. |
| 11. | public void UpdateStatus(RspModificationData rawData, Person person, ref StatisticChanges status) | *rawData* – informații despre conflict;  *person* – persoana dată | *status –* statutul operației de actualizare a statutului persoanei | Se actualizează statutul persoanei și se setează statutul operației. |
| 12. | public void UpdateAddress(RspModificationData rawData, Person person, ref StatisticChanges status) | *rawData* – informații despre conflict;  *person* – persoana dată | *status –* statutul operației de actualizare a adresei persoanei | Se actualizează adresa persoanei și se setează statutul operației. |
| 13. | public void ResolveByMappingAddress(long conflictId, long addressId, bool applyToAllEntries) | *conflictId* – numărul de identificare al conflictului;  *addressId* – numărul de identificare al adresei;  *applyToAllEntries* – indicatorul faptului dacă se aplică rezolvarea conflictului pentru toate conflictele asociate adresei mapate respective |  | Rezolvă conflictul respectiv conform mapării de adrese respective. |

## Subcomponenta Import Statistics Bll

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr. | Funcționalitate | Date de intrare | Date de ieșire | Descriere |
| 1. | public long GetTotalBydate(DateTime dateTime) | *dateTime* – data | Zero | Returnează zero |
| 2. | public void Add(long region, StatisticChanges status) | *region –* regiunea;  *status* – statutul conflictului. |  | Se actualizează datele statistice referitoare la regiunea respectivă. |
| 3. | public void Save() |  |  | Se salvează statisticile existente. |

## Subcomponenta Linked Region Cache

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr. | Funcționalitate | Date de intrare | Date de ieșire | Descriere |
| 1. | public IEnumerable<LinkedRegion> GetAll() |  | Lista tuturor regiunilor legate din Cache | Returnează lista tuturor regiunilor legate din Cache |
| 2. | public IEnumerable<LinkedRegion> GetByRegion(long id) | *id* – numărul de identificare al regiunii | Lista regiunilor legate din Cache care au legătură cu regiunea cu numărul de identificare respectiv. | Returnează lista regiunilor legate din Cache care au legătură cu regiunea cu numărul de identificare respectiv. |

## Subcomponenta Lookup Bll

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr. | Funcționalitate | Date de intrare | Date de ieșire | Descriere |
| 1. | public void Update<T>(long id, string name, string description) | *T* – tipul generic al entității;  *id* – numărul de identificare al entității;  *name* – denumirea entității;  *description* – descrierea entității. |  | Actualizează denumirea și descrierea entității respective. |
| 2. | public void Add<T>(string name, string description) | *T* – tipul generic al entității;  *name* – denumirea entității;  *description* – descrierea entității. |  | Adaugă o entitate nouă cu denumirea și descrierea respectivă. |
| 3. | public void SaveOrUpdate<T>(long? id, string name, string description) | *T* – tipul generic al entității;  *id* – numărul de identificare al entității;  *name* – denumirea entității;  *description* – descrierea entității. |  | Dacă numărul de identificare este *null,* se adaugă o entitate nouă cu denumirea și descrierea respectivă. În caz contrar, se actualizează entitatea cu numărul de identificare respectiv. |
| 4. | public void SaveOrUpdateDocType(long? id, string name, string description, bool isPrimary) | *id* – numărul de identificare al tipului de document;  *name* – denumirea tipului de document;  *description* – descrierea tipului de document;  *isPrimary* – indică dacă tipul de document este primar. |  | Dacă numărul de identificare este *null,* se adaugă un tip de document nou cu informația respectivă. În caz contrar, se actualizează tipul de document cu numărul de identificare respectiv. |
| 5. | public void SaveOrUpdatePersonStatus(long? id, string name, string description, bool isExcludable) | *id* – numărul de identificare al statutului de persoană;  *name* – denumirea statutului de persoană;  *description* – descrierea statutului de persoană;  *isExcludable* – indică dacă statutului de persoană este excludabil. |  | Dacă numărul de identificare este *null,* se adaugă un statut de persoană nou cu informația respectivă. În caz contrar, se actualizează statutul de persoană cu numărul de identificare respectiv. |
| 6. | public void SaveOrUpdateRegionType(long? id, string name, string description, byte rank) | *id* – numărul de identificare al tipului de regiune;  *name* – denumirea tipului de regiune;  *description* – descrierea tipului de regiune;  *rank* – rangul tipului de regiune. |  | Dacă numărul de identificare este *null,* se adaugă un tip de regiune nou cu informația respectivă. În caz contrar, se actualizează tipul de regiune cu numărul de identificare respectiv. |
| 7. | public void UpdatePersonStatus(long id, string name, string description, bool isExcludable) | *id* – numărul de identificare al statutului de persoană;  *name* – denumirea statutului de persoană;  *description* – descrierea statutului de persoană;  *isExcludable* – indică dacă statutului de persoană este excludabil. |  | Se actualizează statutul de persoană cu numărul de identificare respectiv cu informația transmisă ca parametru. |
| 8. | public void DeleteRegion(long id) | *id* – numărul de identificare al regiunii |  | Elimină regiunea cu numărul de identificare respectiv. Dacă există cel puțin o referință la regiunea respectivă este aruncată o excepție. |
| 9. | public PageResponse<StreetDto> GetStreets(PageRequest pageRequest, long regionId) | *pageRequest –* informații despre pagina cerută de la server (*PageNumber –* numărulpaginii, *PageSize* – numărul de elemente pe pagină, etc.);  *regionId* – numărul de identificare al regiunii | pagina respectivă, încărcată cu o listă de date despre străzile din regiunea respectivă (*Items*), mărimea paginii (*PageSize*), indicele primului element (*StartIndex*) și numărul total de elemente (*Total*) | Returnează o pagină încărcată cu o listă de date despre străzile din regiunea respectivă, conform cererii efectuate. |
| 10. | public PageResponse<Region> GetCircumscriptions(PageRequest pageRequest) | *pageRequest –* informații despre pagina cerută de la server (*PageNumber –* numărulpaginii, *PageSize* – numărul de elemente pe pagină, etc.) | pagina respectivă, încărcată cu o listă de regiuni care corespund circumscripțiilor (*Items*), mărimea paginii (*PageSize*), indicele primului element (*StartIndex*) și numărul total de elemente (*Total*) | Returnează o pagină încărcată cu o listă de regiuni care corespund circumscripțiilor, conform cererii efectuate. |
| 11. | public IEnumerable<Region> GetCircumscriptions() |  | lista de regiuni care corespund circumscripțiilor | Returnează listă de regiuni care corespund circumscripțiilor. |
| 12. | public PageResponse<RegionRow> GetRegions(PageRequest pageRequest, long parentRegionId) | *pageRequest –* informații despre pagina cerută de la server (*PageNumber –* numărulpaginii, *PageSize* – numărul de elemente pe pagină, etc.);  *parentRegionId* – numărul de identificare al regiunii părinte. | pagina respectivă, încărcată cu o listă de date despre subregiunile regiunii respective (*Items*), mărimea paginii (*PageSize*), indicele primului element (*StartIndex*) și numărul total de elemente (*Total*) | Returnează o pagină încărcată cu o listă de date despre subregiunile regiunii respective, conform cererii efectuate. |
| 13. | public PageResponse<RegionRow> GetRegionsGrid(PageRequest pageRequest) | *pageRequest –* informații despre pagina cerută de la server (*PageNumber –* numărulpaginii, *PageSize* – numărul de elemente pe pagină, etc.) | pagina respectivă, încărcată cu o listă de date despre regiuni ce au legături cu alte regiuni (*Items*), mărimea paginii (*PageSize*), indicele primului element (*StartIndex*) și numărul total de elemente (*Total*) | Returnează o pagină încărcată cu o listă de date despre regiuni ce au legături cu alte regiuni, conform cererii efectuate. |
| 14. | public void CreateUpdateStreet(long id, string name, string description, long regionId, long streetTypeId, long? ropId, long? saiseId) | *id –* numărul de identificare al străzii;  *name* – denumirea străzii;  *description* – descrierea străzii;  *regionId* – numărul de identificare al regiunii;  *streetTypeId* – numărul de identificare al tipului străzii;  *ropId –* numărul de identificare ROP;  *saiseId* – numărul de identificare al străzii în SAISE |  | Dacă numărul de identificare al străzii este zero, se creează o stradă nouă cu informația transmisă ca parametru. În caz contrar, se actualizează strada cu numărul de identificare respectiv conform informației respective. |
| 15. | public void CreateUpdateRegion(long id, string name, string description, long? parentId, long regionTypeId, bool hasStreets, long? saiseId, long? cuatm) | *id* – numărul de identificare al regiunii;  *name* – denumirea regiunii;  *description* – descrierea regiunii;  *parentId* – numărul de identificare al regiunii părinte;  *regionTypeId –* numărul de identificare al tipului regiunii;  *hasStreets* – indică dacă regiunea are străzi;  *saiseId* – numărul de identificare al regiunii în SAISE;  *cuatm* – numărul de identificare al regiunii în CUATM |  | Dacă numărul de identificare al regiunii este zero, se creează o regiune nouă cu informația transmisă ca parametru. În caz contrar, se actualizează regiunea cu numărul de identificare respectiv conform informației respective. |
| 16. | public void UpdateAdministrativeInfo(long id, string name, string surname, long regionId, long managerTypeId) | *id –* numărul de identificare al administrației publice;  *name –* numele administratorului;  *surname –* prenumele administratorului;  *regionId* – numărul de identificare al regiunii;  *managerTypeId* – numărul de identificare al tipului de conducător. |  | Dacă numărul de identificare al administrației publice este zero, se creează o administrație publică nouă cu informația transmisă ca parametru. În caz contrar, se actualizează administrația publică cu numărul de identificare respectiv conform informației respective. |
| 17. | public bool IsUnique(long id, string name, long parentId, long regionTypeId) | *id* – numărul de identificare al regiunii;  *name* – denumirea regiunii;  *parentId* – numărul de identificare al regiunii părinte;  *regionTypeId* – numărul de identificare al tipului de regiune. | Unicitatea regiunii cu parametrii transmiși | Verifică unicitatea regiunii cu parametrii transmiși. |
| 18. | public bool IsUnique(long id, long regionId, string name, long streetTypeId) | *id* – numărul de identificare al străzii;  *regionId* – numărul de identificare al regiunii;  *name* – denumirea străzii;  *streetTypeId* – numărul de identificare al tipului de stradă. | Unicitatea străzii cu parametrii transmiși | Verifică unicitatea străzii cu parametrii transmiși. |
| 19. | public bool IsUnique<T>(long? id, string name) | *T* – tipul generic al entității;  *id –* numărul de identificare al entității;  *name* – denumirea entității. | Unicitatea entității de tipul respectiv cu parametrii transmiși | Verifică unicitatea entității de tipul respectiv cu parametrii transmiși. |
| 20. | public IEnumerable<Region> GetRegionsOfLevel1ByFilter(string regionNameFilter) | *regionNameFilter* – filtrul de căutare | lista de regiuni de nivel 1 denumirea cărora conține filtrul de căutare | Returnează lista de regiuni de nivel 1 denumirea cărora conține filtrul de căutare. |
| 21. | public IEnumerable<Region> GetRegionsByParentIdAndFilter(long parentId, string regionNameFilter) | *parentId* – numărul de identificare al regiunii părinte;  *regionNameFilter* – filtrul de căutare | Lista de subregiuni al regiunii date, inclusiv regiunea respectivă (dacă are străzi), filtrată conform filtrului de căutare. | Returnează lista de subregiuni al regiunii date, inclusiv regiunea respectivă (dacă are străzi), filtrată conform filtrului de căutare. |
| 22. | public IEnumerable<PollingStation> GetPollingStationsHierarhicallyByRegion(long regionId) | *regionId* – numărul de identificare al regiunii | lista de secții de votare corespunzătoare regiunii și subregiunilor ierarhice ale regiunii respective | Returnează lista de secții de votare corespunzătoare regiunii și subregiunilor ierarhice ale regiunii respective. |
| 23. | public IEnumerable<Street> GetStreetsHierarhicallyByRegionAndFilter(long regionId, string streetNameFilter) | *regionId* – numărul de identificare al regiunii;  *streetNameFilter* – filtrul de căutare al străzii | lista de străzi corespunzătoare regiunii și subregiunilor ierarhice ale regiunii respective, denumirea cărora conțin filtrul de căutare | Returnează lista de străzi corespunzătoare regiunii și subregiunilor ierarhice ale regiunii respective, denumirea cărora conțin filtrul de căutare. |
| 24. | public void VerificationRegion(long id) | *id* – numărul de identificare al regiunii |  | Verifică dacă regiunea are străzi și este activă. Dacă regiunea nu are străzi sau este eliminată, este aruncată o excepție. |
| 25. | public void VerificationIfManagerTypeHasReference(long managerTypeId) | *managerTypeId* – numărul de identificare al tipului de conducător |  | Verifică dacă există cel puțin o referință la tipul de conducător respectiv, caz în care este aruncată o excepție. |
| 26. | public void VerificationIfPersonStatusHasReference(long personStatusId) | *personStatusId* – numărul de identificare al statutului de persoană |  | Verifică dacă există cel puțin o referință la statutul de persoană respectiv, caz în care este aruncată o excepție. |
| 27. | public void VerificationIfGenderHasReference(long genderId) | *genderId* – numărul de identificare al genului persoanei |  | Verifică dacă există cel puțin o referință la genul respectiv, caz în care este aruncată o excepție. |
| 28. | public void VerificationIfDocTypeHasReference(long docTypeId) | *docTypeId* – numărul de identificare al tipului de document al persoanei |  | Verifică dacă există cel puțin o referință la tipul de document respectiv, caz în care este aruncată o excepție. |
| 29. | public void VerificationIfElectionTypeHasReference(long electionTypeId) | *electionTypeId* – numărul de identificare al tipului de scrutin electoral |  | Verifică dacă există cel puțin o referință la tipul de scrutin electoral respectiv, caz în care este aruncată o excepție. |
| 30. | public void VerificationIfPersonAddressTypeHasReference(long personAddressId) | *personAddressId* – numărul de identificare al adresei de persoană |  | Verifică dacă există cel puțin o referință la adresa de persoană respectivă, caz în care este aruncată o excepție. |
| 31. | public void VerificationIfRegionTypeHasReference(long regionTypeId) | *regionTypeId* – numărul de identificare al tipului de regiune |  | Verifică dacă există cel puțin o referință la tipul de regiune respectiv, caz în care este aruncată o excepție. |
| 32. | public void VerificationIfRegionHasReference(long regionId) | *regionId –* numărul de identificare al regiunii |  | Verifică dacă există cel puțin o referință la regiunea respectivă, caz în care este aruncată o excepție. |
| 33. | public void VerificationIfStreetTypeHasReference(long streetTypeId) | *streetTypeId* – numărul de identificare al tipului de stradă |  | Verifică dacă există cel puțin o referință la tipul de stradă respectiv, caz în care este aruncată o excepție. |
| 34. | public void VerificationIfStreetHasReference(long streetId) | *streetId* – numărul de identificare al străzii |  | Verifică dacă există cel puțin o referință la strada respectivă, caz în care este aruncată o excepție. |
| 35. | public void UpdateCircumscription(long id, int? circumscriptionNumber) | *id –* numărul de identificare al regiunii;  *circumscriptionNumber* – numărul circumscripției |  | Actualizează numărul circumscripției al regiunii cu numărul de identificare respectiv. |
| 36. | public bool UniqueValidationCircumscription(long regionId, int? circumscription) | *regionId –* numărul de identificare al regiunii;  *circumscription* – numărul circumscripției | unicitatea circumscripției | Verifică unicitatea circumscripției pentru regiunea respectivă. |
| 37. | public void DeleteStreet(long streetId) | *streetId* – numărul de identificare al străzii |  | Elimină strada cu numărul de identificare respectiv. Dacă există o adresă care face referință la strada respectivă, este aruncată o excepție. |
| 38. | public PageResponse<RegionWithFullyQualifiedName> GetAvailableRegions(PageRequest pageRequest) | *pageRequest –* informații despre pagina cerută de la server (*PageNumber –* numărulpaginii, *PageSize* – numărul de elemente pe pagină, etc.) | pagina respectivă, încărcată cu o listă de date despre regiuni împreună cu denumirea completă (*Items*), mărimea paginii (*PageSize*), indicele primului element (*StartIndex*) și numărul total de elemente (*Total*) | Returnează o pagină încărcată cu o listă de date despre regiuni împreună cu denumirea completă, conform cererii efectuate. |
| 39. | public IList<LinkedRegionsFullName> GetLinkedRegionsByRegionId(long regionId) | *regionId* – numărul de identificare al regiunii | Lista de denumiri complete ale regiunilor legate ce au legătură cu regiunea cu numărul de identificare respectiv. | Returnează lista de denumiri complete ale regiunilor legate ce au legătură cu regiunea cu numărul de identificare respectiv. |
| 40. | public void SaveLinkedRegions(long regionId, long[] linkedRegionIds) | *regionId –* numărul de identificare al regiunii;  *linkedRegionIds* – numerele de identificare ale regiunilor legate cu regiunea dată. |  | Creează o legătură între regiunea dată și lista de regiuni dată ca parametru prin intermediul numerelor de identificare. |
| 41. | public RegionWithFullyQualifiedName GetFullyQualifiedName(long regionId) | *regionId –* numărul de identificare al regiunii | Regiunea împreună cu denumirea completă | Returnează regiunea împreună cu denumirea completă. |
| 42. | IEnumerable<StreetType> ICache<StreetType>.GetAll() |  | Lista tuturor tipurilor de străzi | Returnează lista tuturor tipurilor de străzi. |
| 43. | IEnumerable<Region> IRegionCache.GetByRegistruId(long getAdministrativeCode) | *getAdministrativeCode –* numărul de identificare al regiunii în Registru | Lista de regiuni cu numărul de identificare în Registru respectiv | Returnează lista de regiuni cu numărul de identificare în Registru respectiv. |
| 44. | public Region GetByStatisticCode(long statisticCode) | *statisticCode –* numărul de identificare pentru statistică al regiunii | Regiunea cu numărul de identificare statistic respectiv | Returnează regiunea cu numărul de identificare statistic respectiv. |
| 45. | IEnumerable<Region> ICache<Region>.GetAll() |  | Lista tuturor regiunilor | Returnează lista tuturor regiunilor. |
| 46. | IEnumerable<PollingStation> IPollingStationCache.GetByRegion(long regionId) | *regionId –* numărul de identificare al regiunii | Lista secțiilor de votare asociate regiunii respective | Returnează lista secțiilor de votare asociate regiunii respective. |
| 47. | IEnumerable<PollingStation> ICache<PollingStation>.GetAll() |  | Lista tuturor secțiilor de votare | Returnează lista tuturor secțiilor de votare. |
| 48. | StreetTypeCode IStreetTypeCodeCache.GetByStreetTypeCode(long id) | *id –* numărul de identificare al codului tipului de stradă | Codul tipului de stradă cu numărul de identificare respectiv | Returnează codul tipului de stradă cu numărul de identificare respectiv. |
| 49. | IEnumerable<StreetTypeCode> ICache<StreetTypeCode>.GetAll() |  | Lista tuturor codurilor tipului de stradă | Returnează lista tuturor codurilor tipului de stradă. |
| 50. | IEnumerable<LinkedRegion> ILinkedRegionCache.GetByRegion(long id) | *id –* numărul de identificare al regiunii | Lista de regiuni legate care au legătură cu regiunea respectivă. | Returnează lista de regiuni legate care au legătură cu regiunea respectivă. |
| 51. | IEnumerable<LinkedRegion> ICache<LinkedRegion>.GetAll() |  | Lista tuturor regiunilor legate. | Returnează lista tuturor regiunilor legate. |

## Subcomponenta Notification Bll

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr. | Funcționalitate | Date de intrare | Date de ieșire | Descriere |
| 1. | public IList<string> GetNotificationType() |  | Lista de tipuri de entități ale evenimentelor notificărilor | Returnează lista de tipuri de entități ale evenimentelor notificărilor. |
| 2. | public int CountUnReadNotifications() |  | Numărul de notificări necitite asociate utilizatorului curent | Returnează numărul de notificări necitite asociate utilizatorului curent. |
| 3. | public void UpdateIsRead(long notificationId) | *notificationId* – numărul de identificare al notificării |  | Marchează ca citită notificarea respectivă. |
| 4. | public PageResponse<NotificationReceiver> GetNotificationReceiverList(PageRequest pageRequest) | *pageRequest –* informații despre pagina cerută de la server (*PageNumber –* numărulpaginii, *PageSize* – numărul de elemente pe pagină, etc.) | pagina respectivă, încărcată cu o listă de receptori de notificări asociați utilizatorului curent (*Items*), mărimea paginii (*PageSize*), indicele primului element (*StartIndex*) și numărul total de elemente (*Total*) | Returnează o pagină încărcată cu o listă de receptori de notificări asociați utilizatorului curent, conform cererii efectuate. |

## Subcomponenta Polling Station Bll

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr. | Funcționalitate | Date de intrare | Date de ieșire | Descriere |
| 1. | public IEnumerable<Address> GetPollingStationAddresses(long regionId) | *regionId* – numărul de identificare al regiunii. | Lista de adrese asociate regiunii respective | Returnează lista de adrese accesibile asociate regiunii respective. |
| 2. | public void CreateUpdatePollingStation(long id, long regionId, string number, string location, long addressId, string contactInfo, long? saiseId, PollingStationTypes pollingStationType) | *id* – numărul de identificare al secției de votare;  *regionId* – numărul de identificare al regiunii;  *number* – numărul secției de votare;  *location –* localitatea secției de votare;  *addressId* – numărul de identificare al adresei;  *contactInfo* – informații de contact;  *saiseId* – numărul de identificare al secției de votare în SAISE;  *pollingStationType* – tipul secției de votare |  | Dacă numărul de identificare al secției de votare este zero, se creează o secție de votare nouă cu informația primită ca parametru și toți utilizatorii care au acces la regiunea respectivă sunt notificați. În caz contrar, se actualizează secția de votare respectivă cu informația primită ca parametru. |
| 3. | public void DeletePollingStation(long pollingStationId) | *pollingStationId –* numărul de identificare al secției de votare |  | Dacă există cel puțin o adresă care face referință la secția de votare respectivă, se aruncă o excepție. În caz contrar, se elimină secția de votare cu notificarea utilizatorilor care au acces la regiunea asociată secției de votare. |
| 4. | public PageResponse<PollingStationDto> GetPollingStation(PageRequest pageRequest, long regionId) | *pageRequest –* informații despre pagina cerută de la server (*PageNumber –* numărulpaginii, *PageSize* – numărul de elemente pe pagină, etc.);  *regionId* – numărul de identificare al regiunii | pagina respectivă, încărcată cu informații despre secțiile de votare accesibile din regiunea respectivă (*Items*), mărimea paginii (*PageSize*), indicele primului element (*StartIndex*) și numărul total de elemente (*Total*) | Returnează o pagină încărcată cu informații despre secțiile de votare accesibile din regiunea respectivă, conform cererii efectuate. |
| 5. | public IList<PollingStation> GetAccessiblePollingStations() |  | Lista de secții de votare accesibile utilizatorului curent | Returnează lista de secții de votare accesibile utilizatorului curent. |
| 6. | public PageResponse<PollingStation> GetPollingStationsByRegions(PageRequest pageRequest, long[] selectedRegion) | *pageRequest –* informații despre pagina cerută de la server (*PageNumber –* numărulpaginii, *PageSize* – numărul de elemente pe pagină, etc.);  *selectedRegion* – lista numerelor de identificare ale regiunilor | pagina respectivă, încărcată cu secțiile de votare din circumscripțiile corespunzătoare regiunilor date (*Items*), mărimea paginii (*PageSize*), indicele primului element (*StartIndex*) și numărul total de elemente (*Total*) | Returnează o pagină încărcată cu secțiile de votare din circumscripțiile corespunzătoare regiunilor date, conform cererii efectuate. |
| 7. | public IList<PollingStation> GetPollingStationsByRegion(long[] selectedRegions) | *selectedRegion* – lista numerelor de identificare ale regiunilor | Lista secțiilor de votare din circumscripțiile corespunzătoare regiunilor date | Returnează lista secțiilor de votare din circumscripțiile corespunzătoare regiunilor date. |
| 8. | public void VerificationIfPollingStationHasReference(long pollingStationId) | *pollingStationId* – numărul de identificare al secției de votare |  | Verifică dacă există cel puțin o referință la secția de votare respectivă. În caz afirmativ, este aruncată o excepție. |

## Subcomponenta Polling Station Cache

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr. | Funcționalitate | Date de intrare | Date de ieșire | Descriere |
| 1. | public IEnumerable<PollingStation> GetAll() |  | Lista tuturor secțiilor de votare din Cache | Returnează lista tuturor secțiilor de votare din Cache. |
| 2. | public IEnumerable<PollingStation> GetByRegion(long regionId) | *regionId* – numărul de identificare al regiunii | Lista de secții de votare din Cache asociate regiunii respective | Returnează lista de secții de votare din Cache asociate regiunii respective. |

## Subcomponenta Print Bll

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr. | Funcționalitate | Date de intrare | Date de ieșire | Descriere |
| 1. | public void CreatePrintSession(long electionId, IEnumerable<long> pollingStationIds) | *electionId* – numărul de identificare al scrutinului electoral;  *pollingStationIds* – lista numerelor de identificare ale secțiilor de votare |  | Creează o sesiune de tipărire corespunzătoare scrutinului electoral și secțiilor de votare respective. |
| 2. | public void CancelPrintSession(long printSessionId) | *printSessionId –* numărul de identificare al sesiunii de tipărire |  | Anulează sesiunea de tipărire cu numărul de identificare respectiv. |
| 3. | public void ProcessPrintSession(SSRSPrintParameters printParams) | *printParams –* parametrii sesiunii de tipărire |  | Procesează o sesiune de tipărire în așteptare cu parametrii respectivi. |
| 4. | public IList<PrintSession> GetPrintSessionByStatus(PrintStatus status) | *status –* statutul sesiunii de tipărire | Lista de sesiuni de tipărire care au statutul respectiv | Returnează lista de sesiuni de tipărire care au statutul respectiv. |
| 5. | public PageResponse<ExportPollingStation> GetExportPollingStations(PageRequest pageRequest, long? printSessionId) | *pageRequest –* informații despre pagina cerută de la server (*PageNumber –* numărulpaginii, *PageSize* – numărul de elemente pe pagină, etc.);  *printSessionId* – numărul de identificare al sesiunii de tipărire | pagina respectivă, încărcată cu o listă de secții de votare pregătite pentru exportare de sesiunea de tipărire respectivă (*Items*), mărimea paginii (*PageSize*), indicele primului element (*StartIndex*) și numărul total de elemente (*Total*) | Returnează o pagină încărcată cu secții de votare pregătite pentru exportare de sesiunea de tipărire respectivă, conform cererii efectuate. Dacă *printSessionId* este *null,* acesta se preinițializează cu numărul de identificare al primei sesiuni de tipărire aflate în progres. |
| 6. | public PageResponse<ExportPollingStation> GetHistoryExportPollingStations(PageRequest pageRequest, long? printSessionId) | *pageRequest –* informații despre pagina cerută de la server (*PageNumber –* numărulpaginii, *PageSize* – numărul de elemente pe pagină, etc.);  *printSessionId* – numărul de identificare al sesiunii de tipărire | pagina respectivă, încărcată cu o listă de secții de votare pregătite pentru exportare de sesiunea de tipărire respectivă (*Items*), mărimea paginii (*PageSize*), indicele primului element (*StartIndex*) și numărul total de elemente (*Total*) | Returnează o pagină încărcată cu secții de votare pregătite pentru exportare de sesiunea de tipărire respectivă, conform cererii efectuate. |
| 7. | public PageResponse<PrintSession> GetExportPrintSessions(PageRequest pageRequest) | *pageRequest –* informații despre pagina cerută de la server (*PageNumber –* numărulpaginii, *PageSize* – numărul de elemente pe pagină, etc.) | pagina respectivă, încărcată cu o listă de sesiuni de tipărire (*Items*), mărimea paginii (*PageSize*), indicele primului element (*StartIndex*) și numărul total de elemente (*Total*) | Returnează o pagină încărcată cu sesiuni de tipărire, conform cererii efectuate. |
| 8. | public PageResponse<SaiseExporter> GetSaiseExporter(PageRequest pageRequest) | *pageRequest –* informații despre pagina cerută de la server (*PageNumber –* numărulpaginii, *PageSize* – numărul de elemente pe pagină, etc.) | pagina respectivă, încărcată cu o listă de procese de exportare către SAISE (*Items*), mărimea paginii (*PageSize*), indicele primului element (*StartIndex*) și numărul total de elemente (*Total*) | Returnează o pagină încărcată cu procese de exportare către SAISE, conform cererii efectuate. |
| 9. | public byte[] RequestStayStatementReport(SSRSPrintParameters ssrsParameters, long stayStatementId) | *ssrsParameters* – parametrii sesiunii de tipărire;  *stayStatementId* – numărul de identificare al declarației de ședere | raportul declarației de ședere conform parametrilor dați. | Returnează raportul declarației de ședere conform parametrilor dați. |
| 10. | public IList<ExportPollingStation> GetExportPollingStationsByPrintSession(long printSessionId) | *printSessionId* – numărul de identificare al sesiunii de tipărire | Lista de secții de votare pregătite de exportare de sesiunea de tipărire respectivă | Returnează lista de secții de votare pregătite de exportare de sesiunea de tipărire respectivă |

## Subcomponenta Quartz Lookup Cache Data Provider

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr. | Funcționalitate | Date de intrare | Date de ieșire | Descriere |
| 1. | public IRegionCache RegionCache |  | O instanță de Cache de Regiuni | Returnează o instanță de Cache de Regiuni |
| 2. | public IPollingStationCache PollingStationCache |  | O instanță de Cache de Secții de Votare | Returnează o instanță de Cache de Secții de Votare |
| 3. | public IStreetTypeCache StreetTypeCache |  | O instanță de Cache de Tipuri de Străzi | Returnează o instanță de Cache de Tipuri de Străzi |
| 4. | public IStreetTypeCodeCache StreetTypeCodeCache |  | O instanță de Cache de Coduri de Tipuri de Străzi | Returnează o instanță de Cache de Coduri de Tipuri de Străzi |
| 5. | public ILinkedRegionCache LinkedRegionCache |  | O instanță de Cache de Regiuni Legate | Returnează o instanță de Cache de Regiuni Legate |

## Subcomponenta Region Cache

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr. | Funcționalitate | Date de intrare | Date de ieșire | Descriere |
| 1. | public IEnumerable<Region> GetAll() |  | Lista tuturor regiunilor din Cache | Returnează lista tuturor regiunilor din Cache |
| 2. | public IEnumerable<Region> GetByRegistruId(long AdministrativeCode) | *AdministrativeCode* – numărul de identificare al regiunii în Registru | Lista regiunilor din Cache cu numărul de iden-tificare respectiv în Registru | Returnează lista regiunilor din Cache cu numărul de identificare respectiv în Registru |
| 3. | public Region GetByStatisticCode(long statisticCode) | *statisticCode* – numărul de identificare statistic al regiunii | Regiunea din Cache cu numărul de identificare statistic respectiv | Returnează regiunea din Cache cu numărul de identificare statistic respectiv |

## Subcomponenta Statistics Bll

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr. | Funcționalitate | Date de intrare | Date de ieșire | Descriere |
| 1. | public IQueryOver<Person,Person> BaseQuery() |  | Interogarea generală asupra bazei de date a persoanelor pentru utilizatorii cu rol de Administrator | Returnează interogarea generală asupra bazei de date a persoanelor pentru utilizatorii cu rol de Administrator. |
| 2. | public IQueryOver<Person,Person> RegisrtratorBaseQuery(IQueryOver<Person,Person> query) |  | Interogarea generală asupra bazei de date a persoanelor pentru utilizatorii cu rol de Registrator | Returnează interogarea generală asupra bazei de date a persoanelor pentru utilizatorii cu rol de Registrator. |
| 3. | public long ReturnLongTypeFromQuery(IQueryOver<Person, Person> query) | *query* – interogarea asupra bazei de date a persoanelor | Numărul de înregistrări returnate de interogare | Returnează numărul de înregistrări returnate de interogare. |
| 4. | public long GetTotalNumberOfPeople() |  | Numărul total de persoane din regiunile accesibile utilizatorului curent | Returnează numărul total de persoane din regiunile accesibile utilizatorului curent. |
| 5. | public long GetTotalNumberOfPeopleWithoutDeads() |  | Numărul total de persoane vii din regiunile accesibile utilizatorului curent | Returnează numărul total de persoane vii din regiunile accesibile utilizatorului curent. |
| 6. | public long GetTotalNumberOfVoters() |  | Numărul total de alegători din regiunile accesibile utilizatorului curent | Returnează numărul total de alegători din regiunile accesibile utilizatorului curent. |
| 7. | public long GetTotalNumberOfStayStatementDeclarations() |  | Numărul total de declarații de ședere din regiunile accesibile utilizatorului curent | Returnează numărul total de declarații de ședere din regiunile accesibile utilizatorului curent. |
| 8. | public long GetTotalNumberOfPeopleByGender(GenderTypes gender) | *gender* – genul persoanei | Numărul total de persoane de genul respectiv din regiunile accesibile utilizatorului curent | Returnează numărul total de persoane de genul respectiv din regiunile accesibile utilizatorului curent. |
| 9. | public long GetTotalNumberOfPeopleWithDoBMissing() |  | Numărul total de persoane cu data de naștere lipsă din regiunile accesibile utilizatorului curent | Returnează numărul total de persoane cu data de naștere lipsă din regiunile accesibile utilizatorului curent. |
| 10. | public long GetTotalNumberOfPeopleWithAddressMissing() |  | Numărul total de persoane cu adresa lipsă din regiunile accesibile utilizatorului curent | Returnează numărul total de persoane cu adresa lipsă din regiunile accesibile utilizatorului curent. |
| 11. | public long GetTotalNumberOfPeopleWithDocMissing() |  | Numărul total de persoane cu documentul lipsă din regiunile accesibile utilizatorului curent | Returnează numărul total de persoane cu documentul lipsă din regiunile accesibile utilizatorului curent. |
| 12. | public long GetTotalNumberOfPeopleWithDocExpired() |  | Numărul total de persoane cu documentul expirat din regiunile accesibile utilizatorului curent | Returnează numărul total de persoane cu documentul expirat din regiunile accesibile utilizatorului curent. |
| 13. | public long GetNumberOfPeopleForAgeIntervals(int ageInterval) | *ageInterval* – categoria de vârstă (0-5) | Numărul total de persoane din categoria de vârstă respectivă din regiunile accesibile utilizatorului curent | Returnează numărul total de persoane din categoria de vârstă respectivă din regiunile accesibile utilizatorului curent. |
| 14. | public long GetNewVoters() |  | Zero | Returnează zero |
| 15. | public long GetNewDeadPeople() |  | Zero | Returnează zero |
| 16. | public long GetPersonalDataChanges() |  | Zero | Returnează zero |
| 17. | public long GetAddressesChanges() |  | Zero | Returnează zero |
| 18. | public long GetImportSuccessful() |  | Zero | Returnează zero |
| 19. | public long GetImportFailed() |  | Zero | Returnează zero |
| 20. | public PageResponse<PollingStationStatistics> GetStatisticsForPollingStation(PageRequest pageRequest) | *pageRequest –* informații despre pagina cerută de la server (*PageNumber –* numărulpaginii, *PageSize* – numărul de elemente pe pagină, etc.) | pagina respectivă, încărcată cu statistici despre secțiile de votare (*Items*), mărimea paginii (*PageSize*), indicele primului element (*StartIndex*) și numărul total de elemente (*Total*) | Returnează o pagină încărcată cu statistici despre secțiile de votare, conform cererii efectuate. |
| 21. | public PageResponse<ProblematicDataPollingStationStatistics> GetStatisticsForProblematicDataPollingStation(PageRequest pageRequest) | *pageRequest –* informații despre pagina cerută de la server (*PageNumber –* numărulpaginii, *PageSize* – numărul de elemente pe pagină, etc.) | pagina respectivă, încărcată cu statistici despre secțiile de votare problematice (*Items*), mărimea paginii (*PageSize*), indicele primului element (*StartIndex*) și numărul total de elemente (*Total*) | Returnează o pagină încărcată cu statistici despre secțiile de votare problematice, conform cererii efectuate. |
| 22. | public PageResponse<ImportStatisticsGridDto> GetImportStatistics(PageRequest pageRequest, long? regionId) | *pageRequest –* informații despre pagina cerută de la server (*PageNumber –* numărulpaginii, *PageSize* – numărul de elemente pe pagină, etc.);  *regionId* – numărul de identificare al regiunii | pagina respectivă, încărcată cu informații despre statisticile referitoare la importare (*Items*), mărimea paginii (*PageSize*), indicele primului element (*StartIndex*) și numărul total de elemente (*Total*) | Returnează o pagină încărcată cu informații despre statisticile referitoare la importare, conform cererii efectuate. |
| 23. | public ImportStatisticsDto GetImportStatistics(DateTime importDataTime, long regionId) | *importDataTime –* dataimportării;  *regionId* – numărul de identificare al regiunii. | Informații despre statisticile la importare | Returnează informații despre statisticile la importare. |

## Subcomponenta Street Type Cache

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr. | Funcționalitate | Date de intrare | Date de ieșire | Descriere |
| 1. | public IEnumerable<StreetType> GetAll() |  | Lista tipurilor de străzi din Cache | Returnează lista tipurilor de străzi din Cache. |

## Subcomponenta Street Type Code Cache

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr. | Funcționalitate | Date de intrare | Date de ieșire | Descriere |
| 1. | public IEnumerable<StreetTypeCode> GetAll() |  | Lista codurilor tipurilor de străzi din Cache | Returnează lista codurilor tipurilor de străzi din Cache. |
| 2. | public StreetTypeCode GetByStreetTypeCode(long id) | *id* – numărul de identificare al codului tipului de străzi | Codul tipului de străzi cu numărul de identificare respectiv | Returnează codul tipului de străzi cu numărul de identificare respectiv. |

## Subcomponenta Web Lookup Cache Data Provider

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr. | Funcționalitate | Date de intrare | Date de ieșire | Descriere |
| 1. | public IRegionCache RegionCache |  | Cache de Regiuni | Transformă instanța internă LookupBll într-un Cache de Regiuni. |
| 2. | public IPollingStationCache PollingStationCache |  | Cache de Secții de Votare | Transformă instanța internă LookupBll într-un Cache de Secții de Votare. |
| 3. | public IStreetTypeCache StreetTypeCache |  | Cache de Tipuri de Străzi | Transformă instanța internă LookupBll într-un Cache de Tipuri de Străzi. |
| 4. | public IStreetTypeCodeCache StreetTypeCodeCache |  | Cache de Coduri de Tipuri de Străzi | Transformă instanța internă LookupBll într-un Cache de Coduri de Tipuri de Străzi. |
| 5. | public ILinkedRegionCache LinkedRegionCache |  | Cache de Regiuni Legate | Transformă instanța internă LookupBll într-un Cache de Regiuni Legate. |

## Subcomponenta Voters Bll

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr. | Funcționalitate | Date de intrare | Date de ieșire | Descriere |
| 1. | public PageResponse<Person> Get(PageRequest pageRequest) | *pageRequest –* informații despre pagina cerută de la server (*PageNumber –* numărulpaginii, *PageSize* – numărul de elemente pe pagină, etc.) | pagina respectivă, încărcată cu o listă de persoane cu adresă eligibilă (*Items*), mărimea paginii (*PageSize*), indicele primului element (*StartIndex*) și numărul total de elemente (*Total*) | Returnează o pagină încărcată cu o listă de persoane cu adresă eligibilă, conform cererii efectuate. |
| 2. | public PageResponse<Person> GetIdnp(PageRequest pageRequest, string idnp) | *pageRequest –* informații despre pagina cerută de la server (*PageNumber –* numărulpaginii, *PageSize* – numărul de elemente pe pagină, etc.);  *idnp* – numărul de identificare al persoanei | pagina respectivă, încărcată cu o listă de persoane cu IDNP respectiv (*Items*), mărimea paginii (*PageSize*), indicele primului element (*StartIndex*) și numărul total de elemente (*Total*) | Returnează o pagină încărcată cu o listă de persoane cu IDNP respectiv, conform cererii efectuate. |
| 3. | public PageResponse<VoterRow> GetByFilters(PageRequest pageRequest, long? regionId, long? pollingStationId) | *pageRequest –* informații despre pagina cerută de la server (*PageNumber –* numărulpaginii, *PageSize* – numărul de elemente pe pagină, etc.);  *regionId* – numărul de identificare al regiunii;  *pollingStationId* – numărul de identificare al secției de votare | pagina respectivă, încărcată cu o listă de date despre alegătorii din secția de votare și regiunea respectivă (*Items*), mărimea paginii (*PageSize*), indicele primului element (*StartIndex*) și numărul total de elemente (*Total*) | Returnează o pagină încărcată cu o listă de date despre alegătorii din secția de votare și regiunea respectivă, conform cererii efectuate. |
| 4. | public Person GetByIdnp(string idnp) | *idnp* – numărul de identificare al persoanei | Persoana cu numărul de identificare respectiv | Returnează persoana cu numărul de identificare respectiv. |
| 5. | public int GetVotersCount(PersonAddress address) | *address* – adresa persoanei | Numărul de persoane pe adresa respectivă | Returnează numărul de persoane pe adresa respectivă. |
| 6. | public void UpdateStatus(long personId, long statusId, string confirmation) | *personId* – numărul de identificare al persoanei;  *statusId –* numărul de identificare al statutului;  *confirmation* – confirmarea. |  | Actualizează statutul persoanei respective și notifică utilizatorii care au acces la persoana respectivă. |
| 7. | public PageResponse<Region> SearchRegion(PageRequest pageRequest) | *pageRequest –* informații despre pagina cerută de la server (*PageNumber –* numărulpaginii, *PageSize* – numărul de elemente pe pagină, etc.) | pagina respectivă, încărcată cu o listă de regiuni accesibile utilizatorului curent (*Items*), mărimea paginii (*PageSize*), indicele primului element (*StartIndex*) și numărul total de elemente (*Total*) | Returnează o pagină încărcată cu o listă de regiuni accesibile utilizatorului curent, conform cererii efectuate. |
| 8. | public PageResponse<Election> SearchActiveElections(PageRequest pageRequest) | *pageRequest –* informații despre pagina cerută de la server (*PageNumber –* numărulpaginii, *PageSize* – numărul de elemente pe pagină, etc.) | pagina respectivă, încărcată cu o listă de scrutine electorale active (*Items*), mărimea paginii (*PageSize*), indicele primului element (*StartIndex*) și numărul total de elemente (*Total*) | Returnează o pagină încărcată cu o listă de scrutine electorale active, conform cererii efectuate. |
| 9. | public PageResponse<PollingStation> SearchPollingStations(PageRequest pageRequest, long? regionId) | *pageRequest –* informații despre pagina cerută de la server (*PageNumber –* numărulpaginii, *PageSize* – numărul de elemente pe pagină, etc.);  *regionId* – numărul de identificare al regiunii. | pagina respectivă, încărcată cu o listă de secții de votare din regiunea respectivă (*Items*), mărimea paginii (*PageSize*), indicele primului element (*StartIndex*) și numărul total de elemente (*Total*) | Returnează o pagină încărcată cu o listă de secții de votare din regiunea respectivă, conform cererii efectuate. |
| 10. | public PageResponse<AddressWithPollingStation> SearchAddress(PageRequest pageRequest, long? regionId) | *pageRequest –* informații despre pagina cerută de la server (*PageNumber –* numărulpaginii, *PageSize* – numărul de elemente pe pagină, etc.);  *regionId* – numărul de identificare al regiunii. | pagina respectivă, încărcată cu o listă de adrese cu secții de votare din regiunea respectivă (*Items*), mărimea paginii (*PageSize*), indicele primului element (*StartIndex*) și numărul total de elemente (*Total*) | Returnează o pagină încărcată cu o listă de adrese cu secții de votare din regiunea respectivă, conform cererii efectuate. |
| 11. | public PageResponse<StayStatement> GetStayStatements(PageRequest pageRequest) | *pageRequest –* informații despre pagina cerută de la server (*PageNumber –* numărulpaginii, *PageSize* – numărul de elemente pe pagină, etc.) | pagina respectivă, încărcată cu o listă de declarații de ședere (*Items*), mărimea paginii (*PageSize*), indicele primului element (*StartIndex*) și numărul total de elemente (*Total*) | Returnează o pagină încărcată cu o listă de declarații de ședere, conform cererii efectuate. |
| 12. | public long CreateStayStatement(long id, long personId, long addressId, int? apNumber, string apSuffix, long electionId) | *id –* numărul de identificare al declarației de ședere;  *personId* – numărul de identificare al persoanei;  *addressId* – numărul de identificare al adresei;  *apNumber* – numărul apartamentului;  *apSuffix –* sufixul numărului apartamentului;  *electionId –* numărul de identificare al scrutinului electoral. | Numărul de identificare al noii declarații de ședere | Creează o nouă declarație de ședere, setează noua adresă de votare pentru persoana respectivă, notifică utilizatorii care au acces la regiunea adresei de bază și a adresei declarate și returnează numărul de identificare al noii declarații de ședere. |
| 13. | public long CreateStayStatement(long personId, long pollingStationId, long electionId) | *personId* – numărul de identificare al persoanei;  *pollingStationId* – numărul de identificare al secției de votare;  *electionId –* numărul de identificare al scrutinului electoral. | Numărul de identificare al noii declarații de ședere | Creează o nouă declarație de ședere, setează noua secție de votare pentru persoana respectivă, notifică utilizatorii care au acces la regiunea adresei de bază și a adresei declarate și returnează numărul de identificare al noii declarații de ședere. |
| 14. | public bool ElectionUniqueStayStatement(long personId, long electionId) | *personId* – numărul de identificare al persoanei;  *electionId –* numărul de identificare al scrutinului electoral. | Unicitatea declarației de ședere | Verifică unicitatea declarației de ședere. |
| 15. | public void CancelStayStatement(long stayStatementId) | *stayStatementId –* numărul de identificare al declarației de ședere |  | Anulează declarația de ședere și notifică utilizatorii regiunii adresei de bază și adresei declarate. Este aruncată o excepție în cazul când utilizatorul curent nu are acces la adresa declarată. |
| 16. | public void UpdateAddress(long personId, long addressId, int? apNumber, string apSufix) | *personId* – numărul de identificare al persoanei;  *addressId* – numărul de identificare al adresei;  *apNumber* – numărul apartamentului;  *apSuffix –* sufixul numărului apartamentului. |  | Actualizează adresa de votare a persoanei respective și notifică utilizatorii care au acces la regiunea adresei de bază și adresei declarate. Este aruncată o excepție dacă adresei noi nu-i este asociată o secție de votare. |
| 17. | public Person GetPersonWithEligibleResidence(long personId) | *personId* – numărul de identificare al persoanei | Persoana cu numărul de identificare respectiv | Returnează persoana cu numărul de identificare respectiv. Dacă regiunea adresei de votare nu are străzi sau persoana are depusă o declarație de ședere, este aruncată o excepție. |
| 18. | public IList<Election> GetElection() |  | Lista de scrutine electorale active. | Returnează lista de scrutine electorale active. |
| 19. | public bool VerificationSameAddress(long personAddressId, long declaredAddressId) | *personAddressId –* numărul de identificare al adresei persoanei;  *declaredAddressId* – numărul de identificare al adresei declarate | Indicatorul faptului că adresa persoanei și adresa declarată a persoanei se află în aceeași regiune | Verifică dacă adresa persoanei și adresa declarată a persoanei se află în aceeași regiune. |
| 20. | public bool VerificationSameRegion(long personAddressId, long regionId) | *personAddressId –* numărul de identificare al adresei persoanei;  *regionId* – numărul de identificare al regiunii | Indicatorul faptului că adresa persoanei se află în regiunea dată. | Verifică dacă adresa persoanei se află în regiunea dată. |
| 21. | public long SaveAbroadVoterRegistration(long personId, string abroadAddress, string residenceAddress, string abroadAddresCountry, double abroadAddressLat, double abroadAddressLong, string email, string ipAddress) | *personId* – numărul de identificare al persoanei;  *abroadAddress –* adresa din străinătate;  *residenceAddress* – adresa de reședință;  *abroadAddresCountry* – țara aflării;  *abroadAddressLat* – latitudinea adresei din străinătate;  *abroadAddressLong* – longitudinea adresei din străinătate;  *email* – adresa de e-mail a persoanei;  *ipAddress* – adresa IP a calculatorului persoanei. | Numărul de identificare al declarației de ședere peste hotare | Înregistrează șederea peste hotare și returnează numărul de identificare al declarației de ședere create. |
| 22. | public List<AbroadVoterRegistration> GetAbroadVotersAddress() |  | Lista declarațiilor de ședere peste hotare care au informații despre latitudinea și longitudinea poziției geolocale. | Returnează lista declarațiilor de ședere peste hotare care au informații despre latitudinea și longitudinea poziției geolocale. |
| 23. | public IList<CountryStatisticGroupedDto> GetRegionOfVotes(string country = null) | *country* – țara (opțional) | Statistica numărului de declarații de ședere peste hotare pe fiecare țară | Returnează statistica numărului de declarații de ședere peste hotare pe fiecare țară. |
| 24. | public IList<StatisticTimelineDto> GetAbroadVotersTimeline() |  | Statistica numărului de declarații de ședere peste hotare pe dată | Returnează statistica numărului de declarații de ședere peste hotare pe dată. |
| 25. | public bool IsRegisteredToElection(long id) | *id* – numărul de identificare al persoanei | Indicatorul faptului înregistrării persoanei pentru scrutinul electoral peste hotare | Verifică dacă persoana dată s-a înregistrat pentru scrutinul electoral peste hotare. |
| 26. | public void UpdatePollingStation(long personId, long pollingStationId) | *personId* – numărul de identificare al persoanei;  *pollingStationId* – numărul de identificare al secției de votare |  | Actualizează persoana respectivă cu informația despre secția de votare dată. |
| 27. | public IList<PollingStation> GetRegionPollingStationsByPerson() |  | Lista de secții de votare din regiunile accesibile utilizatorului curent | Returnează lista de secții de votare din regiunile accesibile utilizatorului curent. |
| 28. | public void VerificationSameRegion(long[] peopleIds) | *peopleIds –* numerele de identificare ale persoanelor |  | Verifică dacă persoanele date sunt din aceeași regiune. Dacă nu sunt toate din aceeași regiune, sau cel puțin una din persoane are declarație de ședere, sau regiunea adresei persoanei date are străzi, este aruncată o excepție. |
| 29. | public void ChangePollingStation(long pollingStationId, List<long> peopleIds) | *peopleIds –* numerele de identificare ale persoanelor |  | Mută persoanele date la secția de votare respectivă. |
| 30. | public RegionStreetsType GetRegionStreetsType() |  | Tipul regiunilor (cu străzi, fără străzi, mixt) | Returnează tipul regiunilor (cu străzi, fără străzi, mixt). |

## Subcomponenta Pre Print Lists Exporter

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr. | Funcționalitate | Date de intrare | Date de ieșire | Descriere |
| 1. | public void Process(VotersListPrintingParameters printParams) | *printParams –* parametrii exportatorului de liste în cadrul sesiunii de tipărire |  | Procesarea exportării listelor în cadrul sesiunii de tipărire, conform parametrilor transmiși. |

## Subcomponenta Import Processer

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr. | Funcționalitate | Date de intrare | Date de ieșire | Descriere |
| 1. | public T Get(long id) | *T* – tipul generic al entității;  *id* – numărul de identificare al entității | entitatea de tipul respectiv cu numărul de identificare dat. | Returnează entitatea de tipul respectiv cu numărul de identificare dat. |
| 2. | public void Delete(T rawData) | *T* – tipul generic al entității;  *rawData* – entitatea |  | Elimină entitatea respectivă. |
| 3. | public void Retry(T rawData, string message) | *T* – tipul generic al entității;  *rawData* – entitatea;  *message* - mesajul |  | Setează entitatea pentru reprocesarea ulterioară. |
| 4. | public void ProcessAllNew() |  |  | Procesează toate entitățile noi |
| 5. | public void ProcessNew(params RawDataStatus[] statuses) | *statuses –* statutele entităților |  | Procesează toate entitățile de statut respectiv |
| 6. | public void Process(T rawData) | *rawData* – entitatea; |  | Procesează entitatea respectivă |
| 7. | protected IEnumerable<T> GetRawByStatus(params RawDataStatus[] statuses) | *statuses –* statutele entităților | Un subset de entități de tipul dat și statutul respectiv | Returnează un subset de entități de tipul dat și statutul respectiv |
| 8. | public int CountRawByStatus(params RawDataStatus[] statuses) | *statuses –* statutele entităților | Numărul tuturor entităților de statut respectiv | Numără toate entitățile de statut respectiv |

## Subcomponenta Alegator Processer

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr. | Funcționalitate | Date de intrare | Date de ieșire | Descriere |
| 1. | protected override void ProcessInternal(AlegatorData rawData) | *rawData –* informații despre alegător |  | Procesarea importării unui alegător. Dacă alegătorul cu numărul de identitate respectiv este deja în sistem, se aruncă o excepție. |
| 2. | protected override void NotifySuccess(AlegatorData rawData) | *rawData –* informații despre alegător |  | Notificarea derulării cu succes a operației de importare a alegătorului prin eliminarea informațiilor despre alegătorul importat. |
| 3. | protected override void NotifyFailure(AlegatorData rawData) | *rawData –* informații despre alegător |  | Notificarea derulării cu eroare a operației de importare a alegătorului. |

## Subcomponenta RSA User ProcessOr

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr. | Funcționalitate | Date de intrare | Date de ieșire | Descriere |
| 1. | protected override void ProcessInternal(RsaUser rawData) | *rawData –* informații despre utilizatorul din RSA |  | Procesarea importării unui utilizator. Dacă utilizatorul respectiv este deja în sistem, se aruncă o excepție. |
| 2. | protected override void NotifySuccess(RsaUser rawData) | *rawData –* informații despre utilizatorul din RSA |  | Notificarea derulării cu succes a operației de importare a utilizatorului prin eliminarea informațiilor despre utilizatorul importat. |
| 3. | protected override void NotifyFailure(RsaUser rawData) | *rawData –* informații despre utilizatorul din RSA |  | Notificarea derulării cu eroare a operației de importare a utilizatorului. |

## Subcomponenta RSP Modification ProcessEr

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr. | Funcționalitate | Date de intrare | Date de ieșire | Descriere |
| 1. | protected override void ProcessInternal(RspModificationData rawData) | *rawData –* informații despre modificări în RSP |  | Procesarea importării unei RSP modificări. |
| 2. | protected override void NotifySuccess(RspModificationData rawData) | *rawData –* informații despre modificări în RSP |  | Notificarea derulării cu succes a operației de importare a RSP modificării. |
| 3. | protected override void NotifyFailure(RspModificationData rawData) | *rawData –* informații despre modificări în RSP |  | Notificarea derulării cu eroare a operației de importare a RSP modificării. |

## Subcomponenta Jobs Scheduler

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr. | Funcționalitate | Date de intrare | Date de ieșire | Descriere |
| 1. | public void Connect() |  |  | Instanțiază un planificator de procese de lungă durată. |
| 2. | public bool IsRunning |  | Indicatorul faptului că planificatorul este în execuție | Verifică dacă planificatorul este în execuție. |
| 3. | public List<QuartzJobData> GetJobs() |  | Lista de informații despre procese | Returnează lista de informații despre procese. |
| 4. | public IList<QuartzJobData> GetRunningJobs() |  | Lista de informații despre procesele în execuție | Returnează lista de informații despre procesele în execuție. |
| 5. | public void ExecuteAction(JobAction action, string groupName, string jobName, string triggerName, string triggerGroupName, string cronExpression) | *action* – acțiunea care se execută (întrerupere, executare, pauză, repornire, replanificare);  *groupName* – numele grupului;  *jobName* – denumirea sarcinii;  *triggerName* – denumirea declanșatorului;  *triggerGroupName* – denumirea grupului de declanșatoare;  *cronExpression* – expresia CRON |  | Execută acțiunea respectivă cu parametrii dați. |
| 6. | public bool IsValidCronExpression(string cronExpression) | *cronExpression* – expresia CRON | Indicatorul faptului că expresia CRON este validă | Verifică dacă expresia CRON este validă. |

## Subcomponenta Srv Repository

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr. | Funcționalitate | Date de intrare | Date de ieșire | Descriere |
| 1. | public IQueryOver<T,T> QueryOver<T>() | *T* – tipul generic al entității | Interogarea la tabela bazei de date corespunzătoare tipului entității | Returnează interogarea la tabela bazei de date corespunzătoare tipului entității. |
| 2. | public IQueryOver<T, T> QueryOver<T>(  Expression<Func<T>> alias) | *T* – tipul generic al entității;  *alias* – pseudonim | Interogarea la tabela bazei de date corespunză-toare tipului entității la care este asociat un pseudonim | Returnează interogarea la tabela bazei de date corespunzătoare tipului enti-tății la care este asociat un pseudonim. |
| 3. | public T UnDelete<T>(long id) | *T* – tipul generic al entității;  *id* – numărul de identificare al entității | Entitatea restabilită | Restabilește entitatea de tipul respectiv cu numărul de identificare dat și o returnează. Dacă entitatea nu există, sau nu este eliminată, sau nu a fost eliminată de utilizatorul curent (care nu este administrator), este aruncată o excepție. |
| 4. | protected override ISession Session |  | Sesiunea curentă | Returnează sesiunea curentă |

## Subcomponenta Auditer Repository

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr. | Funcționalitate | Date de intrare | Date de ieșire | Descriere |
| 1. | public PageResponse<T> Get<T>(PageRequest request, long enityId) | *T* – tipul generic al entității, *pageRequest –* informații despre pagina cerută de la server (*PageNumber –* numărulpaginii, *PageSize* – numărul de elemente pe pagină, etc.);  *enityId* – numărul de identificare al entității | pagina respectivă, încărcată cu o listă de audite al entităților cu numărul de identificare dat (*Items*), mărimea paginii (*PageSize*), indicele primului element (*StartIndex*) și numărul total de elemente (*Total*) | Returnează o pagină încărcată cu audite al entităților cu numărul de identificare dat, conform cererii efectuate. |

## Subcomponenta wEB aPI sRV Repository

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr. | Funcționalitate | Date de intrare | Date de ieșire | Descriere |
| 1. | protected override ISession Session |  | Sesiunea curentă | Returnează sesiunea curentă |

# Componenta CEC.Web.SRV

## Subcomponenta Base Controller

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr. | Funcționalitate | Date de intrare | Date de ieșire | Descriere |
| 1. | protected override IAsyncResult BeginExecuteCore(AsyncCallback callback, object state) | *callback* – acțiunea;  *state* - starea | Acțiunea asincronă de începere a execuției acțiunii în contextul controlului curent. | Începerea execuției asincrone a acțiunii date în contextul controlului curent, cu presetarea apriori a culturii curente și a altor proprietăți specifice. |
| 2. | public ActionResult GetFile(string fn) | *fn* – denumirea fișierului de export | Acțiunea de returnare a fișierului de exportare cu denumirea respectivă. | Returnează acțiunea de returnare a fișierului de exportare cu denumirea respectivă. |
| 3. | protected ActionResult ExportGridData(JqGridRequest request, ExportType exportType,  string dataSetName, Type gridModelType, Func<JqGridRequest, JqGridJsonResult> readDataAction) | *request* – informații despre pagina tabelei cerute de la server;  *exportType* – tipul de exportare;  *dataSetName* – denumirea setului de date;  *gridModelType* – tipul modelului de tabel;  *readDataAction* – funcția care returnează setul de date exportate în baza cererii. | Acțiunea în format JSON de exportare în fișier a unei tabele de date în baza cererii efectuate și a parametrilor transmiși. | Returnează acțiunea în format JSON de exportare în fișier a unei tabele de date în baza cererii efectuate și a parametrilor transmiși. |

## Subcomponenta Account Controller

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr. | Funcționalitate | Date de intrare | Date de ieșire | Descriere |
| 1. | public UserManager <SRVIdentityUser> UserManager |  | Managerul de utilizator | Returnează managerul de utilizator. |
| 2. | public ActionResult Login(string returnUrl) | *returnUrl* – adresa url returnată | Pagina de autentificare | Returnează pagina de autentificare, setând adresa URL. |
| 3. | public async Task<ActionResult> Login(LoginViewModel model, string returnUrl) | *returnUrl* – adresa url returnată;  *model* – modelul paginii de autentificare | Pagina de autentificare | Returnează pagina de autentificare, setând adresa URL și modelul de autentificare. |
| 4. | public ActionResult Register() |  | Pagina de înregistrare | Returnează pagina de înregistrare. |
| 5. | public async Task<ActionResult> Register(RegisterViewModel model) | *model* – modelul paginii de înregistrare | Pagina de înregistrare | Returnează pagina de înregistrare, conform modelului dat. |
| 6. | public ActionResult ChangePassword() |  | Pagina de modificare a parolei | Returnează pagina de modificare a parolei. |
| 7. | public async Task<ActionResult> ChangePassword(ChangePasswordViewModel model) | *model* – modelul paginii de modificare a parolei | Pagina de modificare a parolei | Returnează pagina de modificare a parolei, conform modelului dat. |
| 8. | public ActionResult ResetPassword(string userId) | *userId* – numărul de identificare al utilizatorului | Pagina de resetare a parolei a utilizatorului respectiv | Returnează pagina de resetare a parolei a utilizatorului respectiv. |
| 9. | public async Task<ActionResult> ResetPassword(ResetPasswordViewModel model) | *model* – modelul paginii de resetare a parolei | Pagina de resetare a parolei a utilizatorului corespunzător modelului | Returnează pagina de resetare a parolei a utilizatorului corespunzător modelului. |
| 10. | public ActionResult LogOff() |  | Acțiunea de delogare | Returnează acțiunea de delogare |
| 11. | public ActionResult Users() |  | Pagina de vizualizare a utilizatorilor | Returnează pagina de vizualizare a utilizatorilor |
| 12. | public ActionResult CreateUser() |  | Pagina de creare a unui utilizator | Returnează pagina de creare a unui utilizator |
| 13. | public async Task<ActionResult> CreateUser(CreateUserViewModel model) | *model* – modelul paginii de creare a unui utilizator | Pagina de creare a unui utilizator | Returnează pagina de creare a unui utilizator, conform modelului respectiv. |
| 14. | public JqGridJsonResult ListUsersAjax(JqGridRequest request) | *request* – informații despre pagina tabelei cerute de la server | Pagina tabelei cerută de la server, încărcată cu o listă de utilizatori, conform cererii. | Returnează pagina tabelei cerută de la server, încărcată cu o listă de utilizatori, conform cererii. |
| 15. | public ActionResult ExportUsers(JqGridRequest request, ExportType exportType) | *request* – informații despre pagina tabelei cerute de la server | Acțiunea de exportare a paginii de utilizatori, conform cererii. | Returnează acțiunea de exportare a paginii de utilizatori, conform cererii. |
| 16. | public ActionResult UpdateUser(string userId) | *userId* – numărul de identificare al utilizatorului | Acțiunea de actualizare a utilizatorului respectiv. | Returnează acțiunea de actualizare a utilizatorului respectiv. |
| 17. | public ActionResult UpdateUser(UpdateUserModel model) | *model* – modelul paginii de actualizare a unui utilizator | Pagina de actualizare a unui utilizator | Returnează pagina de actualizare a unui utilizator, conform modelului respectiv. |
| 18. | public ActionResult UpdateAccounts() |  | Pagina de actualizare a conturilor de utilizator | Returnează pagina de actualizare a conturilor de utilizator. |
| 19. | public ActionResult UpdateAccounts(MultiSelectAccountEdit model) | *model* – modelul paginii de actualizare a conturilor de utilizator | Pagina de actualizare a conturilor de utilizator | Returnează pagina de actualizare a conturilor de utilizator, conform modelului dat. |
| 20. | public ActionResult DeleteUser(string userId) | *userId* – numărul de identificare al utilizatorului | Acțiunea de eliminare a utilizatorului respectiv | Returnează acțiunea de eliminare a utilizatorului respectiv. |
| 21. | public ActionResult SetRegionsForUser(string userId) | *userId* – numărul de identificare al utilizatorului | Pagina de alocare a regiunilor utilizatorului respectiv | Returnează pagina de alocare a regiunilor utilizatorului respectiv. |
| 22. | public ActionResult SetRegions(AllocateRegionModel model) | *model* – modelul paginii de alocare a regiunilor utilizatorului corespunzător modelului. | Pagina de alocare a regiunilor utilizatorului corespunzător modelului. | Returnează pagina de alocare a regiunilor utilizatorului corespunzător modelului. |
| 23. | public ActionResult SelectRoles() |  | Acțiunea de selectare a rolurilor | Returnează acțiunea de selectare a rolurilor. |
| 24. | public ActionResult AccountStatuses() |  | Acțiunea de vizualizare a statutelor conturilor de utilizator | Returnează acțiunea de vizualizare a statutelor conturilor de utilizator. |
| 25. | public ActionResult AccountStatusesForSearch() |  | Acțiunea de căutare a statutelor conturilor de utilizator | Returnează acțiunea de căutare a statutelor conturilor de utilizator. |
| 26. | public ActionResult ViewProfile(string userId) | *userId* – numărul de identificare al utilizatorului | Pagina de vizualizare a profilului utilizatorului respectiv | Returnează pagina de vizualizare a profilului utilizatorului respectiv. |
| 27. | public ActionResult UpdateProfile(UserProfileModel viewModel) | *viewModel* – modelul paginii de actualizare a profilului utilizatorului corespunzător modelului. | Pagina de actualizare a profilului utilizatorului corespunzător modelului. | Returnează pagina de actualizare a profilului utilizatorului corespunzător modelului. |
| 28. | public UserProfileModel GetUserProfile(string userId) | *userId* – numărul de identificare al utilizatorului | Modelul profilului utilizatorului cu numărul de identificare respectiv. | Returnează modelul profilului utilizatorului cu numărul de identificare respectiv. |

## Subcomponenta Address Controller

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr. | Funcționalitate | Date de intrare | Date de ieșire | Descriere |
| 1. | public ActionResult Buildings() |  | Pagina de vizualizare a apartamentelor | Returnează pagina de vizualizare a apartamentelor. |
| 2. | public JqGridJsonResult ListAddressesBuildingsAjax(JqGridRequest request, long? regionId) | *request* – informații despre pagina tabelei cerute de la server;  *regionId* – numărul de identificare al regiunii. | Pagina tabelei cerută de la server, încărcată cu o listă de date despre adrese, conform cererii. | Returnează pagina tabelei cerută de la server, încărcată cu o listă de date despre adrese, conform cererii. |
| 3. | public ActionResult ExportAddresses(JqGridRequest request, ExportType exportType, long? regionId) | *request* – informații despre pagina tabelei cerute de la server;  *exportType* – tipul de export;  *regionId* – numărul de identificare al regiunii. | Acțiunea de exportare a adreselor din regiunea respectivă, conform cererii efectuate. | Returnează acțiunea de exportare a adreselor din regiunea respectivă, conform cererii efectuate. |
| 4. | public ActionResult SaveBuildingAddress(long regionId, long? addressId) | *regionId* – numărul de identificare al regiunii;  *addressId* – numărul de identificare al adresei | Acțiunea de salvare a adresei în regiunea respectivă. | Returnează acțiunea de salvare a adresei în regiunea respectivă. |
| 5. | public ActionResult SaveBuildingAddress(UpdateBuildingAddressModel model) | *model* – modelul paginii de salvare a adresei corespunzătoare modelului. | Pagina de salvare a adresei corespunzătoare modelului. | Returnează pagina de salvare a adresei corespunzătoare modelului. |
| 6. | public ActionResult ChangePollingStation(long regionId) | *regionId* – numărul de identificare al regiunii | Pagina de modificare a secției de votare pentru regiunea respectivă. | Returnează pagina de modificare a secției de votare pentru regiunea respectivă. |
| 7. | public ActionResult ChangePollingStation(ChangePollingStationForAddressModel model) | *model* – modelul acțiunii de modificare a secției de votare. | Acțiunea de modificare a secției de votare pentru regiunea respectivă, conform modelului. | Returnează acțiunea de modificare a secției de votare pentru regiunea respectivă, conform modelului. |
| 8. | public void DeleteAddress(long addressId) | *addressId* – numărul de identificare al adresei |  | Eliminarea adresei respective. |
| 9. | public ActionResult UpdateGeolocation(long id) | *id* – numărul de identificare al adresei | Acțiunea de actualizare a poziției geolocale a adresei respective. | Returnează acțiunea de actualizare a poziției geolocale a adresei respective. |
| 10. | public ActionResult UpdateGeolocation(EditGeolocationModel model) | *model* – modelul acțiunii de modificare a poziției geolocale a adresei. | Acțiunea de modificare a poziției geolocale a adresei corespunzătoare modelului. | Returnează acțiunea de modificare a poziției geolocale a adresei corespunzătoare modelului. |
| 11. | public ActionResult GetAddressesBy(string term) | *term* – filtrul de căutare a adresei | Acțiunea în format Json de căutare a adresei după filtrul dat. | Returnează acțiunea în format Json de căutare a adresei după filtrul dat. |
| 12. | public ActionResult SelectBuildingType() |  | Acțiunea de selectare a tipului de bloc. | Returnează acțiunea de selectare a tipului de bloc. |
| 13. | public ActionResult GetPollingStationsByFilter(long? regionId) | *regionId* – numărul de identificare al regiunii | Acțiunea de extragere a secțiilor de votare din regiunea respectivă. | Returnează acțiunea de extragere a secțiilor de votare din regiunea respectivă. |
| 14. | public void UnDeleteAddress(long id) | *id* – numărul de identificare al adresei |  | Restabilirea adresei respective. |
| 15. | public ActionResult GetLocation(long regionId, long adressId) | *regionId* – numărul de identificare al regiunii;  *addressId* – numărul de identificare al adresei | Pozițiile geolocale ale regiunii și adresei, serializate în format JSON. | Returnează pozițiile geolo-cale ale regiunii și adresei, serializate în format JSON. |
| 16. | public JsonResult GetStreets(Select2Request request, long? regionId) | *request* – informații despre pagina tabelei cerute de la server.  *regionId* – numărul de identificare al regiunii. | Pagina de străzi din regiunea dată, serializată în format JSON, conform cererii efectuate. | Returnează pagina de străzi din regiunea dată, serializată în format JSON, conform cererii efectuate. |
| 17. | public JsonResult GetStreetsName(long id) | *id* – numărul de identificare al străzii | Strada cu numărul de identificare respectiv, în format JSON. | Returnează strada cu numărul de identificare respectiv, în format JSON. |
| 18. | public long GetRegionByPollingStation(long pollingStationId) | *pollingStationId* – numărul de identificare al secției de votare. | Numărul de identificare al regiunii corespunzătoare secției de votare. | Returnează numărul de identificare al regiunii corespunzătoare secției de votare. |
| 19. | public ActionResult AdjustmentAddresses(long addressId) | *addressId* – numărul de identificare al adresei | Acțiunea de ajustare a adresei date cu cea curentă. | Returnează acțiunea de ajustare a adresei date cu cea curentă. |
| 20. | public ActionResult AdjustmentAddresses(AdjustmentAddressesModel model) | *model* – modelul acțiunii de ajustare a adreselor. | Acțiunea de ajustare a adreselor conform modelului dat. | Returnează acțiunea de ajustare a adreselor conform modelului dat. |
| 21. | public ActionResult RspAddressMappings() |  | Acțiunea de mapare a adreselor. | Returnează acțiunea de mapare a adreselor. |
| 22. | public JqGridJsonResult ListRspAddressMappingsAjax(JqGridRequest request) | *request* – informații despre pagina tabelei cerute de la server. | Pagina tabelei cerută de la server, încărcată cu o listă de mapări de adrese, conform cererii. | Returnează pagina tabelei cerută de la server, încărcată cu o listă de mapări de adrese, conform cererii. |
| 23. | public void DeleteRspAddressMappings(long addressMappingId) | *addressMappingId* – numărul de identificare al mapării de adrese. |  | Elimină maparea de adrese respectivă. |
| 24. | public ActionResult ExportRspAddressMappings(JqGridRequest request, ExportType exportType) | *request* – informații despre pagina tabelei cerute de la server;  *exportType* – tipul de export. | Acțiunea de exportare a mapărilor de adrese, conform cererii efectuate. | Returnează acțiunea de exportare a mapărilor de adrese, conform cererii efectuate. |

## Subcomponenta Conflict Controller

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr. | Funcționalitate | Date de intrare | Date de ieșire | Descriere |
| 1. | public ActionResult Index() |  | Pagina principală de rezolvare a conflictelor | Returnează pagina principală de rezolvare a conflictelor. |
| 2. | public ActionResult LoadGridList(ConflictStatusCode statusCode) | *statusCode* – Codul statutului conflictului | Pagina de încărcare a tabelei de date, corespunzătoare codului statutului conflictului | Pagina de încărcare a tabelei de date, corespunzătoare codului statutului conflictului. |
| 3. | public ActionResult StatusListConflictAjax(JqGridRequest request) | *request* – informații despre pagina tabelei cerute de la server. | Pagina de conflicte de statut, serializată în format JSON, conform cererii efectuate. | Returnează pagina de conflicte de statut, serializată în format JSON, conform cererii efectuate. |
| 4. | public ActionResult ResolveStatusConflict(long conflictId, bool keepRSVVersion = false) | *conflictId –* numărul de identificare al conflictului;  *keepRSVVersion* – indică păstrarea versiunii din RSV. | Acțiunea de soluționare a conflictului de statut. | Returnează acțiunea de soluționare a conflictului de statut. |
| 5. | public ActionResult AddressListConflictAjax(JqGridRequest request) | *request* – informații despre pagina tabelei cerute de la server. | Pagina de conflicte de adresă, serializată în format JSON, conform cererii efectuate. | Returnează pagina de conflicte de adresă, serializată în format JSON, conform cererii efectuate. |
| 6. | public ActionResult ResolveAddressConflict(long conflictId) | *conflictId –* numărul de identificare al conflictului; | Acțiunea de soluționare a conflictului de adresă. | Returnează acțiunea de soluționare a conflictului de adresă. |
| 7. | public ActionResult LoadAddressGridList(long? regionId) | *regionId* – numărul de identificare al regiunii | Acțiunea de încărcare a conflictelor de adresă corespunzătoare regiunii. | Returnează acțiunea de încărcare a conflictelor de adresă corespunzătoare regiunii. |
| 8. | public ActionResult ListAddressesAjax(JqGridRequest request, long? regionId) | *request* – informații despre pagina tabelei cerute de la server;  *regionId* – numărul de identificare al regiunii. | Pagina de adrese din regiunea dată, serializată în format JSON, conform cererii efectuate. | Returnează pagina de adrese din regiunea dată, serializată în format JSON, conform cererii efectuate. |
| 9. | public ActionResult MapAddressConflict(long conflictId, long addressId, bool applyToAllEntries) | *conflictId* – numărul de identificare al conflictului;  *addressId* – numărul de identificare al adresei;  *applyToAllEntries* – indicatorul faptului că se aplică soluționarea conflictului pentru toate conflictele asociate adresei respective. | Acțiunea de soluționare a conflictului de adrese mapate. | Returnează acțiunea de soluționare a conflictului de adrese mapate. |
| 10. | public ActionResult CreateNewAddress(long conflictId, long regionId) | *conflictId* – numărul de identificare al conflictului;  *regionId* – numărul de identificare al regiunii. | Acțiunea de creare a unei noi adrese în regiunea respectivă în baza conflictului dat. | Returnează acțiunea de creare a unei noi adrese în regiunea respectivă în baza conflictului dat. |
| 11. | public ActionResult CreateNewAddress(CreateAddressModel model) | *model* – modelul acțiunii de creare a unei noi adrese. | Acțiunea de creare a unei noi adrese în baza modelului dat. | Returnează acțiunea de creare a unei noi adrese în baza modelului dat. |
| 12. | public ActionResult RejectAddressConflict(List<long> conflictIds, ConflictStatusCode conflictStatus) | *conflictIds* – lista numerelor de iden-tificare a conflic-telor de adresă;  *conflictStatus* – codul statutului conflictului. | Acțiunea de respingere a conflictelor de adresă. | Returnează acțiunea de respingere a conflictelor de adresă. |
| 13. | public ActionResult RejectAddressConflictSave(RejectConflict model) | *model* – modelul acțiunii de respingere a unei noi adrese. | Acțiunea de respingere a conflictului de adresă, conform modelului. | Returnează acțiunea de respingere a conflictului de adresă, conform modelului. |
| 14. | public ActionResult AddressNoStreetListConflictAjax(JqGridRequest request) | *request* – informații despre pagina tabelei cerute de la server. | Pagina de conflicte de adresă fără stradă, serializată în format JSON, conform cererii efectuate. | Returnează pagina de conflicte de adresă fără stradă, serializată în format JSON, conform cererii efectuate. |
| 15. | public ActionResult ResolveAddressNoStreetConflict(List<long> conflictIds) | *conflictIds* – lista numerelor de iden-tificare ale conflic-telor de adresă fără stradă. | Acțiunea de soluționare a conflictelor de adresă fără stradă. | Returnează acțiunea de soluționare a conflictelor de adresă fără stradă. |
| 16. | public ActionResult MapAddressNoStreetConflict(ResolveAddressNoStreetConflictModel model) | *model* – modelul acțiunii de asignare a conflictului de adresă fără stradă. | Acțiunea de asignare a conflictului de adresă fără stradă conform modelului. | Returnează acțiunea de asignare a conflictului de adresă fără stradă conform modelului. |
| 17. | public ActionResult RegionListConflictAjax(JqGridRequest request) | *request* – informații despre pagina tabelei cerute de la server. | Pagina de conflicte de regiune, serializată în format JSON, conform cererii efectuate. | Returnează pagina de conflicte de regiune, serializată în format JSON, conform cererii efectuate. |
| 18. | public ActionResult ResolveRegionConflict(long conflictId) | *conflictId* –numărul de identificare a conflictului de regiune. | Acțiunea de soluționare a conflictului de regiune. | Returnează acțiunea de soluționare a conflictului de regiune. |
| 19. | public ActionResult StreetListConflictAjax(JqGridRequest request) | *request* – informații despre pagina tabelei cerute de la server. | Pagina de conflicte de stradă, serializată în format JSON, conform cererii efectuate. | Returnează pagina de conflicte de stradă, serializată în format JSON, conform cererii efectuate. |
| 20. | public ActionResult ResolveStreetConflict(long conflictId) | *conflictId* – numărul de identifi-care a conflictului de stradă. | Acțiunea de soluționare a conflictului de stradă. | Returnează acțiunea de soluționare a conflictului de stradă. |
| 21. | public ActionResult LocalityConflictAjax(JqGridRequest request) | *request* – informații despre pagina tabelei cerute de la server. | Pagina de conflicte de localitate, serializată în format JSON, conform cererii efectuate. | Returnează pagina de conflicte de localitate, serializată în format JSON, conform cererii efectuate. |
| 22. | public ActionResult ResolveLocalityConflict(long conflictId) | *conflictId* – numărul de identificare a conflictului de localitate. | Acțiunea de soluționare a conflictului de localitate. | Returnează acțiunea de soluționare a conflictului de localitate. |
| 23. | public ActionResult ChangeAddress(long conflictId, long addressId) | *conflictId* – numărul de identificare al conflictului;  *addressId* – numărul de identificare al adresei. | Acțiunea de modificare a adresei corespunzător conflictului dat. | Returnează acțiunea de modificare a adresei corespunzător conflictului dat. |
| 24. | public ActionResult ChangePollingStation(long conflictId, long pollingStationId) | *conflictId* – numărul de identificare al conflictului;  *pollingStationId* – numărul de identificare al secției de votare. | Acțiunea de modificare a secției de votare corespunzător conflictului dat. | Returnează acțiunea de modificare a secției de votare corespunzător conflictului dat. |
| 25. | public JsonResult GetUserRegions(Select2Request request) | *request* – informații despre pagina tabelei cerute de la server. | Pagina de regiuni accesibile utilizatorului curent, serializată în format JSON, conform cererii efectuate. | Returnează pagina de regiuni accesibile utilizatorului curent, serializată în format JSON, conform cererii efectuate. |
| 26. | public JsonResult CheckRegionForStreets(long regionId) | *regionId* – numărul de identificare al regiunii. | Indicatorul faptului că regiunea respectivă are străzi, serializat în format JSON. | Returnează indicatorul faptului că regiunea respectivă are străzi, serializat în format JSON. |
| 27. | public ActionResult LoadPollingStationsList() |  | Acțiunea de încărcare a secțiilor de votare | Returnează acțiunea de încărcare a secțiilor de votare. |
| 28. | public ActionResult GetViewConflict(long conflictId, ConflictStatusCode conflictType) | *conflictId* – numărul de identificare al conflictului;  *conflictType* – codul statutului conflictului. | Pagina de vizualizare a conflictului, corespunzătoare tipului de conflict. | Returnează pagina de vizualizare a conflictului, corespunzătoare tipului de conflict. |
| 29. | public ActionResult SelectSource() |  | Acțiunea de selectare a sursei. | Returnează acțiunea de selectare a sursei. |
| 30. | public ActionResult SelectStatus() |  | Acțiunea de selectare a statutului. | Returnează acțiunea de selectare a statutului. |

## Subcomponenta Election Controller

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr. | Funcționalitate | Date de intrare | Date de ieșire | Descriere |
| 1. | public ActionResult Index() |  | Pagina principală de gestionare a scrutinelor electorale | Returnează pagina principală de gestionare a scrutinelor electorale. |
| 2. | public JqGridJsonResult ListElectionsAjax(JqGridRequest request) | *request* – informații despre pagina tabelei cerute de la server. | Pagina de scrutine electorale, serializată în format JSON, conform cererii efectuate. | Returnează pagina de scrutine electorale, serializată în format JSON, conform cererii efectuate. |
| 3. | public ActionResult ExportElections(JqGridRequest request, ExportType exportType) | *request* – informații despre pagina tabelei cerute de la server;  *exportType* – tipul de exportare | Acțiunea de exportare a paginii respective de scrutine electorale. | Returnează acțiunea de exportare a paginii respective de scrutine electorale. |
| 4. | public ActionResult CreateUpdate(long? electionId) | *electionId* – numărul de identi-ficare al scrutinului electoral | Acțiunea de creare sau actualizare a scrutinului elec-toral cu numărul de identificare res-pectiv. | Returnează acțiunea de creare sau actualizare a scrutinului electoral cu numărul de identificare respectiv. |
| 5. | public ActionResult CreateUpdate(UpdateElectionModel model) | *model* – modelul acțiunii de creare sau actualizare a scrutinului electoral. | Acțiunea de creare sau actualizare a scrutinului electoral corespunzător modelului dat. | Returnează acțiunea de creare sau actualizare a scrutinului electoral corespunzător modelului dat. |
| 6. | public void Delete(long id) | *id* – numărul de identificare al scru-tinului electoral |  | Eliminarea scrutinului electo-ral respectiv. |
| 7. | public void UnDelete(long id) | *id* – numărul de identificare al scru-tinului electoral |  | Restabilirea scrutinului elec-toral respectiv. |

## Subcomponenta Error Controller

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr. | Funcționalitate | Date de intrare | Date de ieșire | Descriere |
| 1. | public ViewResult NotFound(string aspxerrorpath) |  | Pagina de eroare *NotFoundHttpException* | Returnează pagina de eroare *NotFoundHttpException.* |

## Subcomponenta Export Controller

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr. | Funcționalitate | Date de intrare | Date de ieșire | Descriere |
| 1. | public ActionResult ExportList() |  | Acțiunea de exportare a unei liste. | Returnează acțiunea de exportare a unei liste. |
| 2. | public ActionResult HistoryExportList() |  | Acțiunea de ex-portare a istori-cului. | Returnează acțiunea de exportare a istoricului. |
| 3. | public ActionResult HistoryExportRsaToSaise() |  | Acțiunea de ex-portare a istori-cului din RSA către SAISE. | Returnează acțiunea de exportare a istoricului din RSA către SAISE. |
| 4. | public ActionResult ExportRsaToSaise() |  | Acțiunea de ex-portare din RSA către SAISE. | Returnează acțiunea de exportare din RSA către SAISE. |
| 5. | public ActionResult ExportRsaToSaise(ExportRsaToSaiseModel model) | *model* – modelul acțiunii de ex-portare din RSA către SAISE. | Acțiunea de ex-portare din RSA către SAISE, conform modelului dat. | Returnează acțiunea de exportare din RSA către SAISE, conform modelului dat. |
| 6. | public void CancelPrintSession() |  |  | Anularea sesiunii de tipărire. |
| 7. | public ActionResult GetPollingStationbyRegions(Select2Request request, long[] regionsId) | *request* – informații despre pagina tabelei cerute de la server;  *regionsId* – lista numerelor de identificare ale regiunilor. | Pagina de secții de votare din regiunea dată, serializată în format JSON, conform cererii efectuate. | Returnează pagina de secții de votare din regiunea dată, serializată în format JSON, conform cererii efectuate. |
| 8. | public ActionResult ListPollingStationAjax(JqGridRequest request, long? printSessionId) | *request* – informații despre pagina tabelei cerute de la server;  *printSessionId* –numărul de identificare al sesiunii de tipărire. | Pagina de secții de votare destinate exportării în sesiunea de tipărire respectivă, serializată în format JSON, conform cererii efectuate. | Returnează pagina de secții de votare destinate exportării în sesiunea de tipărire respectivă, serializată în format JSON, conform cererii efectuate. |
| 9. | public ActionResult ListHistoryPollingStationAjax(JqGridRequest request, long? printSessionId) | *request* – informații despre pagina tabelei cerute de la server;  *printSessionId* –numărul de identificare al sesiunii de tipărire. | Pagina de secții de votare din istoric destinate exportării în sesiunea de tipărire respectivă, serializată în format JSON, conform cererii efectuate. | Returnează pagina de secții de votare din istoric destinate exportării în sesiunea de tipărire respectivă, serializată în format JSON, conform cererii efectuate. |
| 10. | public ActionResult ListPrintSessionsAjax(JqGridRequest request) | *request* – informații despre pagina tabelei cerute de la server. | Pagina de sesiuni de tipărire pentru exportare, serializată în format JSON, conform cererii efectuate. | Returnează pagina de sesiuni de tipărire pentru exportare, serializată în format JSON, conform cererii efectuate. |
| 11. | public ActionResult ListSaiseExporterAjax(JqGridRequest request) | *request* – informații despre pagina tabelei cerute de la server. | Pagina de procese de exportare către SAISE, serializată în format JSON, conform cererii efectuate. | Returnează pagina de procese de exportare către SAISE, serializată în format JSON, conform cererii efectuate. |
| 12. | public ActionResult ListSaiseExporterStageAjax(JqGridRequest request, long? saiseExporterId) | *request* – informații despre pagina tabelei cerute de la server;  *saiseExporterId* – numărul de identificare al exportatorului către SAISE. | Pagina de etape ale procesului de exportare către SAISE dat, serializată în format JSON, conform cererii efectuate. | Returnează pagina de etape ale procesului de exportare către SAISE dat, serializată în format JSON, conform cererii efectuate. |
| 13. | public ActionResult ListHistorySaiseExporterStageAjax(JqGridRequest request, long? saiseExporterId) | *request* – informații despre pagina tabelei cerute de la server;  *saiseExporterId* – numărul de identificare al exportatorului către SAISE. | Pagina de etape din istoric ale procesului de exportare către SAISE dat, serializată în format JSON, conform cererii efectuate. | Returnează pagina de etape din istoric ale procesului de exportare către SAISE dat, serializată în format JSON, conform cererii efectuate. |
| 14. | public void CreatePrintSession(long electionId, long[] regionsId, long[] pollingStationsId) | *electionId* – numărul de identi-ficare al scrutinului electoral;  *regionsId* – lista numerelor de identificare ale regiunilor;  *pollingStationsId* – lista numerelor de identificare ale secțiilor de votare. |  | Creează o sesiune de tipărire pentru scrutinul electoral respectiv, regiunile și secțiile de votare date. |
| 15. | public ActionResult SelectPrintStatus() |  | Acțiunea de selec-tare a statutului de tipărire. | Returnează acțiunea de selec-tare a statutului de tipărire. |
| 16. | public ActionResult SelectStageStatus() |  | Acțiunea de selec-tare a statutului etapei exportatorului către SAISE. | Returnează acțiunea de selec-tare a statutului etapei exportatorului către SAISE. |
| 17. | public ActionResult SelectExporterStatus() |  | Acțiunea de selec-tare a statutului exportatorului către SAISE. | Returnează acțiunea de selec-tare a statutului exportatorului către SAISE. |
| 18. | public ActionResult GetProgressOfSaiseExporter(long saiseExporterId) | *saiseExporterId* – numărul de identi-ficare al exporta-torului către SAISE | Acțiunea de returnare în format JSON a progresului și a erorilor. | Returnează acțiunea de returnare în format JSON a progresului și a erorilor. |
| 19. | public ActionResult GetProgressOfPrintSession(long printSessionId) | *printSessionId* – numărul de identi-ficare al sesiunii de tipărire. | Acțiunea de returnare în format JSON a progresului, a erorilor și secțiilor de votare finisate. | Returnează acțiunea de returnare în format JSON a progresului, a erorilor și secțiilor de votare finisate. |

## Subcomponenta History Controller

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr. | Funcționalitate | Date de intrare | Date de ieșire | Descriere |
| 1. | public ActionResult ManagerTypeHistory (long id) | *id –* numărul de identificare al tipu-lui de conducător | Acțiunea de vizualizare a istoricului tipului de conducător cu numărul de identificare respectiv. | Returnează acțiunea de vizualizare a istoricului tipului de conducător cu numărul de identificare respectiv. |
| 2. | public JqGridJsonResult ListManagerTypeHistoryAjax(JqGridRequest request, long id) | *request* – informații despre pagina tabelei cerute de la server;  *id* – numărul de identificare al tipului de conducător | Pagina de audite ale tipului de conducător cu numărul de identificare respectiv, serializată în format JSON, conform cererii efectuate. | Returnează pagina de audite ale tipului de conducător cu numărul de identificare respectiv, serializată în format JSON, conform cererii efectuate. |
| 3. | public ActionResult GenderHistory(long id) | *id –* numărul de identificare al genului | Acțiunea de vizualizare a istoricului genului cu numărul de identificare respectiv. | Returnează acțiunea de vizualizare a istoricului genului cu numărul de identificare respectiv. |
| 4. | public JqGridJsonResult ListGenderHistoryAjax(JqGridRequest request, long id) | *request* – informații despre pagina tabelei cerute de la server;  *id –* numărul de identificare al genului | Pagina de audite ale genului cu numărul de identificare respectiv, serializată în format JSON, conform cererii efectuate. | Returnează pagina de audite ale genului cu numărul de identificare respectiv, serializată în format JSON, conform cererii efectuate. |
| 5. | public ActionResult ElectionTypeHistory(long id) | *id –* numărul de identificare al tipului de scrutin electoral | Acțiunea de vizualizare a istoricului tipului de scrutin elec-toral cu numărul de identificare respectiv. | Returnează acțiunea de vizualizare a istoricului tipului de scrutin electoral cu numărul de identificare respectiv. |
| 6. | public JqGridJsonResult ListElectionTypeHistoryAjax(JqGridRequest request, long id) | *request* – informații despre pagina tabelei cerute de la server;  *id –* numărul de identificare al tipului de scrutin electoral | Pagina de audite ale tipului de scrutin electoral cu numărul de identificare res-pectiv, serializată în format JSON, conform cererii efectuate. | Returnează pagina de audite ale tipului de scrutin electoral cu numărul de identificare respectiv, serializată în format JSON, conform cererii efectuate. |
| 7. | public ActionResult ElectionHistory(long id) | *id –* numărul de identificare al scrutinului electoral | Acțiunea de vizualizare a istoricului scruti-nului electoral cu numărul de iden-tificare respectiv. | Returnează acțiunea de vizualizare a istoricului scrutinului electoral cu numărul de identificare respectiv. |
| 8. | public JqGridJsonResult ListElectionHistoryAjax(JqGridRequest request, long id) | *request* – informații despre pagina tabelei cerute de la server;  *id –* numărul de identificare al scrutinului electoral | Pagina de audite ale scrutinului electoral cu numă-rul de identificare respectiv, seriali-zată în format JSON, conform cererii efectuate. | Returnează pagina de audite ale scrutinului electoral cu numărul de identificare respectiv, serializată în format JSON, conform cererii efectuate. |
| 9. | public ActionResult StreetTypeHistory(long id) | *id –* numărul de identificare al tipului de stradă | Acțiunea de vizualizare a istoricului tipului de stradă cu numărul de iden-tificare respectiv. | Returnează acțiunea de vizualizare a istoricului tipului de stradă cu numărul de identificare respectiv. |
| 10. | public JqGridJsonResult ListStreetTypeHistoryAjax(JqGridRequest request, long id) | *request* – informații despre pagina tabelei cerute de la server;  *id –* numărul de identificare al tipului de stradă | Pagina de audite ale tipului de stradă cu numărul de identificare respectiv, seriali-zată în format JSON, conform cererii efectuate. | Returnează pagina de audite ale tipului de stradă cu numărul de identificare respectiv, serializată în format JSON, conform cererii efectuate. |
| 11. | public ActionResult PersonStatusHistory(long id) | *id –* numărul de identificare al statutului de persoană | Acțiunea de vizualizare a istoricului statu-tului de persoană cu numărul de identificare res-pectiv. | Returnează acțiunea de vizualizare a istoricului statutului de persoană cu numărul de identificare respectiv. |
| 12. | public JqGridJsonResult ListPersonStatusHistoryAjax(JqGridRequest request, long id) | *request* – informații despre pagina tabelei cerute de la server;  *id –* numărul de identificare al statutului de per-soană | Pagina de audite ale statutului de persoană cu nu-mărul de identi-ficare respectiv, serializată în for-mat JSON, con-form cererii efec-tuate. | Returnează pagina de audite ale statutului de persoană cu numărul de identificare respectiv, serializată în format JSON, conform cererii efectuate. |
| 13. | public ActionResult DocumentTypeHistory(long id) | *id –* numărul de identificare al tipului de document | Acțiunea de vizualizare a istoricului tipului de document cu numărul de identificare respectiv. | Returnează acțiunea de vizualizare a istoricului tipului de document cu numărul de identificare respectiv. |
| 14. | public JqGridJsonResult ListDocumentTypeHistoryAjax(JqGridRequest request, long id) | *request* – informații despre pagina tabelei cerute de la server;  *id –* numărul de identificare al tipului de document | Pagina de audite ale tipului de document cu nu-mărul de identi-ficare respectiv, serializată în for-mat JSON, con-form cererii efec-tuate. | Returnează pagina de audite ale tipului de document cu numărul de identificare respectiv, serializată în format JSON, conform cererii efectuate. |
| 15. | public ActionResult RegionTypeHistory(long id) | *id –* numărul de identificare al tipului de regiune | Acțiunea de vizualizare a istoricului tipului de regiune cu numărul de identificare respectiv. | Returnează acțiunea de vizualizare a istoricului tipului de regiune cu numărul de identificare respectiv. |
| 16. | public JqGridJsonResult ListRegionTypeHistoryAjax(JqGridRequest request, long id) | *request* – informații despre pagina tabelei cerute de la server;  *id –* numărul de identificare al tipului de regiune | Pagina de audite ale tipului de regiune cu nu-mărul de identi-ficare respectiv, serializată în for-mat JSON, con-form cererii efec-tuate. | Returnează pagina de audite ale tipului de regiune cu numărul de identificare respectiv, serializată în format JSON, conform cererii efectuate. |
| 17. | public ActionResult RegionHistory(long id) | *id –* numărul de identificare al regiunii | Acțiunea de vizualizare a istoricului regiunii cu numărul de identificare respectiv. | Returnează acțiunea de vizualizare a istoricului regiunii cu numărul de identificare respectiv. |
| 18. | public JqGridJsonResult ListRegionHistoryAjax(JqGridRequest request, long id) | *request* – informații despre pagina tabelei cerute de la server;  *id –* numărul de identificare al regiunii | Pagina de audite ale regiunii cu nu-mărul de identi-ficare respectiv, serializată în for-mat JSON, con-form cererii efec-tuate. | Returnează pagina de audite ale regiunii cu numărul de identificare respectiv, serializată în format JSON, conform cererii efectuate. |
| 19. | public ActionResult StreetHistory(long id) | *id –* numărul de identificare al străzii | Acțiunea de vizualizare a istoricului străzii cu numărul de identificare respectiv. | Returnează acțiunea de vizualizare a istoricului străzii cu numărul de identificare respectiv. |
| 20. | public JqGridJsonResult ListStreetHistoryAjax(JqGridRequest request, long id) | *request* – informații despre pagina tabelei cerute de la server;  *id –* numărul de identificare al străzii | Pagina de audite ale străzii cu nu-mărul de identi-ficare respectiv, serializată în for-mat JSON, con-form cererii efec-tuate. | Returnează pagina de audite ale străzii cu numărul de identificare respectiv, serializată în format JSON, conform cererii efectuate. |
| 21. | public ActionResult PollingStationHistory(long id) | *id –* numărul de identificare al secției de votare | Acțiunea de vizualizare a istoricului secției de votare cu numărul de identificare respectiv. | Returnează acțiunea de vizualizare a istoricului secției de votare cu numărul de identificare respectiv. |
| 22. | public JqGridJsonResult ListPollingStationHistoryAjax(JqGridRequest request, long id) | *request* – informații despre pagina tabelei cerute de la server;  *id –* numărul de identificare al secției de votare | Pagina de audite ale secției de votare cu nu-mărul de identi-ficare respectiv, serializată în for-mat JSON, con-form cererii efec-tuate. | Returnează pagina de audite ale secției de votare cu numărul de identificare respectiv, serializată în format JSON, conform cererii efectuate. |
| 23. | public ActionResult AddressHistory(long id) | *id –* numărul de identificare al adresei | Acțiunea de vizualizare a istoricului adresei cu numărul de identificare respectiv. | Returnează acțiunea de vizualizare a istoricului adresei cu numărul de identificare respectiv. |
| 24. | public JqGridJsonResult ListAddressHistoryAjax(JqGridRequest request, long id) | *request* – informații despre pagina tabelei cerute de la server;  *id –* numărul de identificare al adresei | Pagina de audite ale adresei cu nu-mărul de identi-ficare respectiv, serializată în for-mat JSON, con-form cererii efec-tuate. | Returnează pagina de audite ale adresei cu numărul de identificare respectiv, seri-alizată în format JSON, conform cererii efectuate. |
| 25. | public ActionResult PersonAddressTypeHistory(long id) | *id –* numărul de identificare al tipului de adresă de persoană | Acțiunea de vizualizare a istoricului tipului de adresă de persoană cu nu-mărul de iden-tificare respectiv. | Returnează acțiunea de vizualizare a istoricului tipului de adresă de persoană cu numărul de identificare respectiv. |
| 26. | public JqGridJsonResult ListPersonAddressTypeHistoryAjax(JqGridRequest request, long id) | *request* – informații despre pagina tabelei cerute de la server;  *id –* numărul de identificare al tipului de adresă de persoană | Pagina de audite ale tipului de adresă de persoa-nă cu numărul de identificare res-pectiv, serializată în format JSON, conform cererii efectuate. | Returnează pagina de audite ale tipului de adresă de persoană cu numărul de identificare respectiv, seri-alizată în format JSON, conform cererii efectuate. |
| 27. | public ActionResult ConfigurationSettingHistory(long id) | *id* – numărul de identificare al parametrului de configurare | Acțiunea de vizu-alizare a istori-cului parametrului de configurare cu numărul de identificare respectiv. | Returnează acțiunea de vizualizare a istoricului parametrului de configurare cu numărul de identificare respectiv. |
| 28. | public JqGridJsonResult ListConfigurationSettingHistoryAjax(JqGridRequest request, long id) | *request* – informații despre pagina tabelei cerute de la server;  *id –* numărul de identificare al parametrului de configurare | Pagina de audite ale parametrului de configurare cu numărul de identi-ficare respectiv, serializată în for-mat JSON, con-form cererii efec-tuate. | Returnează pagina de audite ale parametrului de configurare cu numărul de identificare respectiv, seri-alizată în format JSON, conform cererii efectuate. |

## Subcomponenta Home Controller

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr. | Funcționalitate | Date de intrare | Date de ieșire | Descriere |
| 1. | public ActionResult Index() |  | Pagina principală | Returnează pagina principală. |
| 2. | public ActionResult About() |  | Pagina descriptivă | Returnează pagina descrip-tivă. |
| 3. | public ActionResult Contact() |  | Pagina de contact | Returnează pagina de contact. |
| 4. | public ActionResult KeepAlive() |  | Acțiune de menținere în viață prin păstrarea unui conținut OK. | Returnează o acțiune de menținere în viață prin păstrarea unui conținut OK. |

## Subcomponenta Notification Controller

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr. | Funcționalitate | Date de intrare | Date de ieșire | Descriere |
| 1. | public ActionResult Index() |  | Pagina principală | Returnează pagina principală. |
| 2. | public ActionResult SelectNotificationType() |  | Acțiunea de selectare a unui tip de notificare | Returnează acțiunea de selectare a unui tip de notificare. |
| 3. | public JqGridJsonResult ListNotificationAjax(JqGridRequest request) | *request* – informații despre pagina tabelei cerute de la server | Pagina de recep-tori de notificări, serializată în for-mat JSON, con-form cererii efec-tuate. | Returnează pagina de receptori de notificări, seri-alizată în format JSON, conform cererii efectuate. |
| 4. | public ActionResult ExportNotifications(JqGridRequest request, ExportType exportType) | *request* – informații despre pagina tabelei cerute de la server;  *exportType* – tipul de exportare | Acțiunea de exportare a paginii de receptori de notificări, con-form cererii efec-tuate. | Returnează acțiunea de exportare a paginii de receptori de notificări, con-form cererii efectuate. |
| 5. | public ActionResult GetCountNotification() |  | Numărul de notificări necitite, serializat în format JSON | Returnează numărul de notificări necitite, serializat în format JSON. |
| 6. | public void UpdateIsRead(long notificationId) | *notificationId* – numărul de iden-tificare al notificării |  | Marchează ca citită notificarea cu numărul de identificare respectiv. |

## Subcomponenta Polling Station Controller

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr. | Funcționalitate | Date de intrare | Date de ieșire | Descriere |
| 1. | public ActionResult Index() |  | Pagina principală | Returnează pagina principală. |
| 2. | public ActionResult CreateUpdatePollingStation(long regionId, long? pollingStationId) | *regionId* - numărul de identificare al regiunii;  *pollingStationId* - numărul de iden-tificare al secției de votare | Acțiunea de creare sau actualizare a secției de votare cu numărul de identificare res-pectiv din regi-unea cu numărul de identificare dat. | Returnează acțiunea de creare sau actualizare a secției de votare cu numărul de identificare res-pectiv din regi-unea cu numărul de identificare dat. |
| 3. | public ActionResult CreateUpdatePollingStation(UpdatePollingStationModel model) | *model* – modelul acțiunii de creare sau actualizare a secției de votare | Acțiunea de creare sau actualizare a secției de votare, corespunzător modelului dat. | Returnează acțiunea de creare sau actualizare a secției de votare, corespunzător mode-lului dat. |
| 4. | public void DeletePollingStation(long pollingStationId) | *pollingStationId* - numărul de iden-tificare al secției de votare |  | Eliminarea secției de votare cu numărul de identificare respectiv. |
| 5. | public void UnDeletePollingStation(long id) | *id* - numărul de identificare al sec-ției de votare |  | Restabilirea secției de votare cu numărul de identificare respectiv. |
| 6. | public JqGridJsonResult ListPollingStationAjax(JqGridRequest request, long? regionId) | *request* – informații despre pagina tabelei cerute de la server;  *regionId* - numărul de identificare al regiunii. | Pagina de secții de votare, serializată în format JSON, conform cererii efectuate. | Returnează pagina de secții de votare, serializată în format JSON, conform cererii efectuate. |
| 7. | public JsonResult GetAddressName(long id) | *id* – numărul de identificare al adresei | Denumirea completă a adresei cu numărul de identificare respectiv, serializat în format JSON. | Returnează denumirea completă a adresei cu numărul de identificare respectiv, serializat în format JSON. |
| 8. | public ActionResult ExportPollingStations(JqGridRequest request, ExportType exportType, long? regionId) | *request* – informații despre pagina tabelei cerute de la server;  *exportType* – tipul de exportare;  *regionId* - numărul de identificare al regiunii. | Acțiunea de exportare a paginii de secții de votare din regiunea respectivă, con-form cererii efec-tuate. | Returnează acțiunea de exportare a paginii de secții de votare din regiunea respectivă, conform cererii efectuate. |

## Subcomponenta Quartz Admin Controller

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr. | Funcționalitate | Date de intrare | Date de ieșire | Descriere |
| 1. | public ActionResult Index() |  | Pagina principală | Returnează pagina principală. |
| 2. | public ActionResult ListQuartzJobs(JqGridRequest request) | *request* – informații despre pagina tabelei cerute de la server. | Pagina de procese de lungă durată, serializată în format JSON, conform cererii efectuate. | Returnează pagina de procese de lungă durată, serializată în format JSON, conform cererii efectuate. |
| 3. | public ActionResult ListRunningJobs(JqGridRequest request) | *request* – informații despre pagina tabelei cerute de la server. | Pagina de procese de lungă durată în execuție, seriali-zată în format JSON, conform cererii efectuate. | Returnează pagina de procese de lungă durată în execuție, serializată în format JSON, conform cererii efectuate. |
| 4. | public ActionResult PerformAction(QuartzJobActionModel model) | *model* – modelul acțiunii de execu-tare a unui proces de lungă durată | Acțiunea de executare a unui proces de lungă durată | Returnează acțiunea de executare a unui proces de lungă durată. |
| 5. | public ActionResult JobActions(QuartzScheduledJobsGridModel jobModel) | *jobModel* – modelul planificatorului proceselor de lungă durată. | Acțiunea de ges-tionare a proce-selor de lungă du-rată, conform mo-delului planifica-torului. | Returnează acțiunea de ges-tionare a proceselor de lungă durată, conform modelului planificatorului. |
| 6. | public ActionResult InterruptRunningJob(QuartzRunningJobsGridModel jobModel) | *jobModel* – modelul planificatorului proceselor de lungă durată. | Acțiunea de între-rupere a execuției procesului de lungă durată aflat în execuție, conform mode-lului planificato-rului. | Returnează acțiunea de între-rupere a execuției procesului de lungă durată aflat în execuție, conform modelului planificatorului. |
| 7. | public ActionResult GetQuartzJobs() |  | Pagina de procese de lungă durată, serializată în format JSON. | Returnează pagina de procese de lungă durată, serializată în format JSON. |
| 8. | public ActionResult GetRunningJobs() |  | Pagina de procese de lungă durată în execuție, serializată în format JSON. | Returnează pagina de procese de lungă durată în execuție, serializată în format JSON. |

## Subcomponenta Reporting Controller

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr. | Funcționalitate | Date de intrare | Date de ieșire | Descriere |
| 1. | public ActionResult ListPrinting() |  | Acțiunea de tipărire a unei liste electorale. | Returnează acțiunea de tipărire a unei liste electorale. |
| 2. | public ActionResult ListPrinting(ListPrintingModel model) | *model* – modelul acțiunii de tipărire a unei liste electorale | Acțiunea de tipărire a unei liste electorale, conform modelului dat. | Returnează acțiunea de tipărire a unei liste electorale, conform modelului dat. |
| 3. | public ActionResult GetPollingStationbyRegions(Select2Request request, long[] regionsId) | *request* – informații despre pagina cerută de la server;  *regionsId* – lista numerelor de identificare ale regiunilor | Pagina de secții de votare din regiu-nile respective, se-rializată în format JSON, conform cererii efectuate. | Returnează pagina de secții de votare din regiunile respective, serializată în for-mat JSON, conform cererii efectuate. |
| 4. | public JsonResult GetCircumscriptions(Select2Request request) | *request* – informații despre pagina cerută de la server | Pagina de regiuni corespunzătoare circumscripțiilor, serializată în format JSON, conform cererii efectuate. | Returnează pagina de regiuni corespunzătoare circumscrip-țiilor, serializată în format JSON, conform cererii efectuate. |
| 5. | public ActionResult PollingStationsBorders() |  | Acțiunea de vizualizare a hota-relor secțiilor de votare | Returnează acțiunea de vizualizare a hotarelor secțiilor de votare. |
| 6. | public ActionResult PollingStationsBorders(PollingStationBordersModel model) | *model* – modelul acțiunii de vizualizare a hotarelor secțiilor de votare | Acțiunea de vizualizare a hota-relor secțiilor de votare, conform modelului dat. | Returnează acțiunea de vizualizare a hotarelor secțiilor de votare, conform modelului dat. |

## Subcomponenta Statistics Controller

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr. | Funcționalitate | Date de intrare | Date de ieșire | Descriere |
| 1. | public ActionResult LoadInfoTab() |  | Pagina de infor-mații statististice | Returnează pagina de infor-mații statististice. |
| 2. | public ActionResult LoadAgeDistributionTab() |  | Pagina distribuției pe vârste | Returnează pagina distribuției pe vârste. |
| 3. | public ActionResult LoadProblematicDataTab() |  | Pagina de date problematice | Returnează pagina de date problematice. |
| 4. | public ActionResult LoadStatisticsFromImportTab() |  | Pagina cu statistici la importare | Returnează pagina cu statistici la importare. |
| 5. | public ActionResult LoadStatisticsForPollingStation() |  | Pagina cu statistici pe secții de votare | Returnează Pagina cu statistici pe secții de votare. |
| 6. | public JqGridJsonResult ListStatisticsPollingStationsAjax(JqGridRequest request) | *request* – informații despre pagina tabelei cerută de la server | Pagina de statistici pe secții de votare, serializată în format JSON, conform cererii efectuate. | Returnează pagina de statistici pe secții de votare, serializată în format JSON, conform cererii efectuate. |
| 7. | public ActionResult ExportStatisticsPollingStations(JqGridRequest request, ExportType exportType) | *request* – informații despre pagina tabelei cerute de la server;  *exportType* – tipul de exportare. | Acțiunea de exportare a paginii de statistici pe secții de votare, conform cererii efectuate. | Returnează acțiunea de exportare a paginii de statistici pe secții de votare, conform cererii efectuate. |
| 8. | public ActionResult GetTotalPeople() |  | Numărul total de persoane, serializat în format JSON | Returnează numărul total de persoane, serializat în format JSON. |
| 9. | public ActionResult GetTotalNumberOfPeopleWithoutDeads() |  | Numărul total de persoane vii, serializat în format JSON | Returnează numărul total de persoane vii, serializat în format JSON. |
| 10. | public ActionResult GetTotalNumberOfVoters() |  | Numărul total de alegători, seriali-zat în format JSON | Returnează numărul total de alegători, serializat în format JSON. |
| 11. | public ActionResult GetTotalNumberOfStayStatementDeclarations() |  | Numărul total de declarații de ședere, serializat în format JSON | Returnează numărul total de declarații de ședere, serializat în format JSON. |
| 12. | public ActionResult GetTotalNumberOfPeopleByGender(GenderTypes gender) | *gender* – genul | Numărul total de persoane de genul respectiv, seriali-zat în format JSON | Returnează numărul total de persoane de genul respectiv, serializat în format JSON. |
| 13. | public ActionResult GetTotalNumberOfPeopleWithDoBMissing() |  | Numărul total de persoane cu data de naștere lipsă, serializat în format JSON | Returnează numărul total de persoane cu data de naștere lipsă, serializat în format JSON. |
| 14. | public ActionResult GetTotalNumberOfPeopleWithAddressMissing() |  | Numărul total de persoane cu adresa lipsă, serializat în format JSON | Returnează numărul total de persoane cu adresa lipsă, serializat în format JSON. |
| 15. | public ActionResult GetTotalNumberOfPeopleWithDocMissing() |  | Numărul total de persoane cu documentul lipsă, serializat în format JSON | Returnează numărul total de persoane cu documentul lipsă, serializat în format JSON. |
| 16. | public ActionResult GetTotalNumberOfPeopleWithDocExpired() |  | Numărul total de persoane cu documentul expirat, serializat în format JSON | Returnează numărul total de persoane cu documentul expirat, serializat în format JSON. |
| 17. | public ActionResult GetNumberOfPeopleForAgeIntervals() |  | Numărul de persoane în fiecare categorie de vârstă, serializat în format JSON | Returnează numărul de persoane în fiecare categorie de vârstă, serializat în format JSON. |
| 18. | public ActionResult GetNewVoters() |  | Numărul de alegători noi, serializat în format JSON | Returnează numărul de alegători noi, serializat în format JSON. |
| 19. | public ActionResult GetNewDeadPeople() |  | Numărul de persoane decedate noi, serializat în format JSON | Returnează numărul de persoane decedate noi, serializat în format JSON. |
| 20. | public ActionResult GetPersonalDataChanges() |  | Numărul de date personale modi-ficate, serializat în format JSON | Returnează numărul de date personale modificate, seria-lizat în format JSON. |
| 21. | public ActionResult GetAddressesChanges() |  | Numărul de adrese modificate, serializat în format JSON | Returnează numărul de adrese modificate, serializat în format JSON. |
| 22. | public ActionResult GetImportSuccessful() |  | Numărul de im-portări reușite, serializat în for-mat JSON | Returnează numărul de im-portări reușite, serializat în format JSON. |
| 23. | public ActionResult GetImportFailed() |  | Numărul de im-portări eșuate, serializat în for-mat JSON | Returnează numărul de im-portări eșuate, serializat în format JSON. |
| 24. | public JqGridJsonResult ListProblematicDataPollingStationAjax(JqGridRequest request) | *request* – informații despre pagina tabelei cerută de la server | Pagina de statistici despre datele problematice pe secțiile de votare, serializată în format JSON, conform cererii efectuate. | Returnează pagina de statistici despre datele problematice pe secțiile de votare, serializată în format JSON, conform cererii efectuate. |
| 25. | public JqGridJsonResult ImporterListAjax(JqGridRequest request, long? regionId) | *request* – informații despre pagina tabelei cerută de la server;  *regionId* – numărul de identificare al regiunii. | Pagina de statistici la importare pentru regiunea respectivă, seria-lizată în format JSON, conform cererii efectuate. | Returnează pagina de statistici la importare pentru regiunea respectivă, seria-lizată în format JSON, conform cererii efectuate. |
| 26. | public ActionResult GetImportStatistics(string importDataTime, long regionId) | *importDataTime* – data importării;  *regionId* – numărul de identificare al regiunii. | Acțiunea de vizu-alizare a statis-ticilor la importare pentru data și re-giunea respectivă. | Returnează acțiunea de vizu-alizare a statisticilor la importare pentru data și regiunea respectivă. |

## Subcomponenta Voters Controller

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr. | Funcționalitate | Date de intrare | Date de ieșire | Descriere |
| 1. | public ActionResult Index() |  | Pagina principală | Returnează pagina principală. |
| 2. | public ActionResult StayStatements() |  | Pagina de decla-rații de ședere | Returnează pagina de decla-rații de ședere. |
| 3. | public JqGridJsonResult ListVotersAjax(JqGridRequest request, long? regionId, long? pollingStationId) | *request* – informații despre pagina tabelei cerută de la server;  *regionId* – numărul de identificare al regiunii;  *pollingStationId –* numărul de identi-ficare al secției de votare. | Pagina de alegă-tori din secția de votare și regiunea respectivă, seria-lizată în format JSON, conform cererii efectuate. | Returnează pagina de alegă-tori din secția de votare și regiunea respectivă, seria-lizată în format JSON, conform cererii efectuate. |
| 4. | public ActionResult ExportVoters(JqGridRequest request, ExportType exportType, long? regionId, long? pollingStationId) | *request* – informații despre pagina tabelei cerute de la server;  *exportType* – tipul de exportare;  *regionId* – numărul de identificare al regiunii;  *pollingStationId –* numărul de identi-ficare al secției de votare. | Acțiunea de exportare a paginii de alegători din secția de votare și regiunea respec-tivă, conform ce-rerii efectuate. | Returnează acțiunea de exportare a paginii de alegă-tori din secția de votare și regiunea respectivă, conform cererii efectuate. |
| 5. | public ActionResult SelectVoterStatus() |  | Acțiunea de selec-tare a unui tip de statut de persoană | Returnează acțiunea de selec-tare a unui tip de statut de persoană. |
| 6. | public ActionResult SelectVoterGender() |  | Acțiunea de selec-tare a unui gen | Returnează acțiunea de selec-tare a unui gen. |
| 7. | public ActionResult SearchVotersByIdnpAjax(JqGridRequest request, string idnp) | *request* – informații despre pagina tabelei cerută de la server;  *idnp* – numărul de identitate al alegătorului. | Pagina de alegă-tori cu numărul de identitate respec-tiv, serializată în format JSON, conform cererii efectuate. | Returnează pagina de alegă-tori cu numărul de identitate respectiv, serializată în format JSON, conform cererii efectuate. |
| 8. | public JqGridJsonResult ListStayStatementsAjax(JqGridRequest request) | *request* – informații despre pagina tabelei cerută de la server. | Pagina de declarații de ședere, serializată în format JSON, conform cererii efectuate. | Returnează pagina de declarații de ședere, serializată în format JSON, conform cererii efectuate. |
| 9. | public ActionResult ExportStayStatements(JqGridRequest request, ExportType exportType) | *request* – informații despre pagina tabelei cerute de la server;  *exportType* – tipul de exportare. | Acțiunea de ex-portare a paginii de declarații de ședere, conform cererii efectuate. | Returnează acțiunea de exportare a paginii de declarații de ședere, conform cererii efectuate. |
| 10. | public ActionResult CreateStayStatement(long personId) | *personId* – numărul de identificare al persoanei. | Acțiunea de creare a unei declarații de ședere pentru per-soana respectivă. | Returnează acțiunea de creare a unei declarații de ședere pentru persoana respectivă. |
| 11. | public ActionResult CreateStayStatement(StayStatementModel model) | *model* – modelul acțiunii de creare a unei declarații de ședere | Acțiunea de creare a unei declarații de ședere pentru per-soana corespun-zătoare modelului. | Returnează acțiunea de creare a unei declarații de ședere pentru persoana corespun-zătoare modelului. |
| 12. | public bool GetRegionStreetsType (long regionId) | *regionId* – numărul de identificare al regiunii | Indicatorul faptu-lui că regiunea respectivă are străzi. | Verifică dacă regiunea respectivă are străzi. |
| 13. | public ActionResult SetStatus(long personId) | *personId* – numărul de identificare al persoanei. | Acțiunea de setare a statutului per-soanei respective. | Returnează acțiunea de setare a statutului persoanei respective. |
| 14. | public ActionResult SetStatus(UpdateVotersStatusModel model) | *model* – modelul acțiunii de setare a statutului. | Acțiunea de setare a statutului per-soanei corespun-zătoare modelului. | Returnează acțiunea de setare a statutului persoanei cores-punzătoare modelului. |
| 15. | public JsonResult GetRegions(Select2Request request) | *request* – informații despre pagina tabelei cerută de la server. | Pagina de regiuni, serializată în format JSON, conform cererii efectuate. | Returnează pagina de regiuni, serializată în format JSON, conform cererii efectuate. |
| 16. | public JsonResult GetActiveElections(Select2Request request) | *request* – informații despre pagina tabelei cerută de la server. | Pagina de scrutine electorale active, serializată în format JSON, conform cererii efectuate. | Returnează pagina de scrutine electoral active, serializată în format JSON, conform cererii efectuate. |
| 17. | public JsonResult GetPollingStations(Select2Request request, long? regionId) | *request* – informații despre pagina tabelei cerută de la server;  *regionId* – numărul de identificare al regiunii. | Pagina de secții de votare din regiu-nea dată, seriali-zată în format JSON, conform cererii efectuate. | Returnează pagina de secții de votare din regiunea dată, serializată în format JSON, conform cererii efectuate. |
| 18. | public ActionResult SelectElectionInstance() |  | Acțiunea de selectare a scrutinului electoral. | Returnează acțiunea de selectare a scrutinului electoral. |
| 19. | public JsonResult GetAddresses(Select2Request request, long? regionId) | *request* – informații despre pagina tabelei cerută de la server;  *regionId* – numărul de identificare al regiunii. | Pagina de adrese din regiunea dată, serializată în format JSON, conform cererii efectuate. | Returnează pagina de adrese din regiunea dată, serializată în format JSON, conform cererii efectuate. |
| 20. | public ActionResult StayStatementView(long id) | *id* – numărul de identificare al de-clarației de ședere | Acțiunea de vizu-alizare a decla-rației de ședere cu numărul de iden-tificare respectiv. | Returnează acțiunea de vizu-alizare a declarației de ședere cu numărul de identificare respectiv. |
| 21. | public ActionResult SelectAddressType() |  | Acțiunea de selectare a tipului de adresă. | Returnează acțiunea de selectare a tipului de adresă. |
| 22. | public ActionResult UpdateAddress(long personId) | *personId* – numărul de identificare al regiunii. | Acțiunea de actua-lizare a adresei persoanei respec-tive. | Returnează acțiunea de actualizare a adresei per-soanei respective. |
| 23. | public ActionResult UpdateAddress(UpdateAddressModel model) | *model* – modelul acțiunii de actuali-zare a adresei. | Acțiunea de actua-lizare a adresei persoanei cores-punzătoare mode-lului. | Returnează acțiunea de actua-lizare a adresei persoanei corespunzătoare modelului. |
| 24. | public ActionResult ChangePollingStationForm(long[] peopleIds) | *peopleIds* – lista numerelor de iden-tificare ale persoa-nelor | Pagina de modi-ficare a secției de votare pentru persoanele respec-tive, din aceeași regiune. | Returnează pagina de modi-ficare a secției de votare pentru persoanele respective, din aceeași regiune. |
| 25. | public ActionResult ChangePollingStation(ChangePollingStationModel model) | *model* – modelul acțiunii de modifi-care a secției de votare. | Acțiunea de modi-ficare a secției de votare pentru persoanele cores-punzătoare mode-lului. | Returnează acțiunea de modi-ficare a secției de votare pentru persoanele cores-punzătoare modelului. |
| 26. | public void CancelStayStatement(long stayStatementId) | *stayStatementId –* numărul de identi-ficare al declarației de ședere |  | Anularea declarației de ședere cu numărul de identificare respectiv. |
| 27. | public ActionResult StayStatementPrinting(long stayStatementId) | *stayStatementId –* numărul de identi-ficare al declarației de ședere | Acțiunea de tipă-rire a declarației de ședere. | Returnează acțiunea de tipă-rire a declarației de ședere. |
| 28. | public ActionResult StayStatementReport(long stayStatementId) | *stayStatementId –* numărul de identi-ficare al declarației de ședere | Acțiunea de rapor-tare a declarației de ședere. | Returnează acțiunea de raportare a declarației de ședere. |
| 29. | public JsonResult GetPollingStationName(long id) | *id* – numărul de identificare al secț-iei de votare | Denumirea com-pletă a secției de votare, serializată în format JSON. | Returnează denumirea com-pletă a secției de votare, serializată în format JSON. |
| 30. | public JsonResult GetElectionsName(long id) | *id* – numărul de identificare al scru-tinului electoral | Denumirea com-pletă a scrutinului electoral, seriali-zată în format JSON. | Returnează denumirea com-pletă a scrutinului electoral, serializată în format JSON. |
| 31. | public JsonResult GetRegionName(long id) | *id* – numărul de identificare al regiunii | Denumirea com-pletă a regiunii, serializată în for-mat JSON. | Returnează denumirea com-pletă a regiunii, serializată în format JSON. |
| 32. | public JsonResult GetPollingStationsName(long id) | *id* – numărul de identificare al secț-iei de votare | Denumirea com-pletă a secției de votare, serializată în format JSON. | Returnează denumirea com-pletă a secției de votare, serializată în format JSON. |
| 33. | public ActionResult RspChecking() |  | Acțiunea de veri-ficare a persoanei direct în RSP. | Returnează acțiunea de veri-ficare a persoanei direct în RSP. |
| 34. | public ActionResult RspChecking(string idnp) |  | Acțiunea de veri-ficare a persoanei direct în RSP în baza numărului de identitate al per-soanei, serializată în format JSON. | Returnează acțiunea de veri-ficare a persoanei direct în RSP în baza numărului de identitate al persoanei, serializată în format JSON. |

## Subcomponenta Lookup Controller

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr. | Funcționalitate | Date de intrare | Date de ieșire | Descriere |
| 1. | public ActionResult StreetTypes() |  | Acțiunea de vizualizare a tipurilor de străzi | Returnează acțiunea de vizualizare a tipurilor de străzi. |
| 2. | public ActionResult RegionTypes() |  | Acțiunea de vizu-alizare a tipurilor de regiuni | Returnează acțiunea de vizualizare a tipurilor de regiuni. |
| 3. | public ActionResult ManagerTypes() |  | Acțiunea de vizu-alizare a tipurilor de conducători | Returnează acțiunea de vizualizare a tipurilor de conducători. |
| 4. | public ActionResult PersonStatus() |  | Acțiunea de vizu-alizare a statutelor de persoană | Returnează acțiunea de vizualizare a statutelor de persoană. |
| 5. | public ActionResult Genders() |  | Acțiunea de vizu-alizare a genurilor | Returnează acțiunea de vizualizare a genurilor. |
| 6. | public ActionResult Streets() |  | Acțiunea de vizu-alizare a străzilor | Returnează acțiunea de vizualizare a străzilor. |
| 7. | public ActionResult DocumentTypes() |  | Acțiunea de vizu-alizare a tipurilor de documente | Returnează acțiunea de vizualizare a tipurilor de documente. |
| 8. | public ActionResult ElectionTypes() |  | Acțiunea de vizu-alizare a tipurilor de scrutine electorale | Returnează acțiunea de vizualizare a tipurilor de scrutine electorale. |
| 9. | public ActionResult RegionsTree() |  | Acțiunea de vizu-alizare a regiu-nilor în formă arborescentă. | Returnează acțiunea de vizualizare a regiunilor în formă arborescentă. |
| 10. | public ActionResult RegionsGrid() |  | Acțiunea de vizu-alizare a regiu-nilor în formă tabelară. | Returnează acțiunea de vizualizare a regiunilor în formă tabelară. |
| 11. | public ActionResult PersonAddressTypes() |  | Acțiunea de vizu-alizare a tipurilor de adrese de persoană. | Returnează acțiunea de vizualizare a tipurilor de adrese de persoană. |
| 12. | public ActionResult Circumscriptions() |  | Acțiunea de vizu-alizare a circum-scripțiilor. | Returnează acțiunea de vizualizare a circumscrip-țiilor. |
| 13. | public ActionResult CreateUpdateManagerType(long? managerTypeId) | *managerTypeId* – numărul de iden-tificare al tipului de conducător. | Acțiunea de creare sau actualizare a tipului de condu-cător respectiv. | Returnează acțiunea de creare sau actualizare a tipului de conducător respectiv. |
| 14. | public ActionResult CreateUpdateManagerType(UpdateLookupModel model) | *model* – modelul acțiunii de creare sau actualizare a tipului de condu-cător. | Acțiunea de creare sau actualizare a tipului de condu-cător corespunză-tor modelului. | Returnează acțiunea de creare sau actualizare a tipului de conducător corespunzător modelului. |
| 15. | public void DeleteManagerType(long id) | *id* – numărul de identificare al tipu-lui de conducător. |  | Eliminarea tipului de conducător cu numărul de identificare respectiv. |
| 16. | public void UnDeleteManagerType(long id) | *id* – numărul de identificare al tipu-lui de conducător. |  | Restabilirea tipului de conducător cu numărul de identificare respectiv. |
| 17. | public ActionResult CreateUpdateStreetType(long? streetTypeId) | *streetTypeId* – numărul de iden-tificare al tipului de stradă. | Acțiunea de creare sau actualizare a tipului de stradă respectiv. | Returnează acțiunea de creare sau actualizare a tipului de stradă respectiv. |
| 18. | public ActionResult CreateUpdateStreetType(UpdateLookupModel model) | *model* – modelul acțiunii de creare sau actualizare a tipului de stradă. | Acțiunea de creare sau actualizare a tipului de stradă corespunzător modelului. | Returnează acțiunea de creare sau actualizare a tipului de stradă corespunzător modelului. |
| 19. | public void DeleteStreetType(long id) | *id* – numărul de identificare al tipu-lui de stradă. |  | Eliminarea tipului de stradă cu numărul de identificare respectiv. |
| 20. | public void UnDeleteStreetType(long id) | *id* – numărul de identificare al tipu-lui de stradă. |  | Restabilirea tipului de stradă cu numărul de identificare respectiv. |
| 21. | public ActionResult CreateUpdateRegionType(long? regionTypeId) | *regionTypeId* – numărul de iden-tificare al tipului de regiune. | Acțiunea de creare sau actualizare a tipului de regiune respectiv. | Returnează acțiunea de creare sau actualizare a tipului de regiune respectiv. |
| 22. | public ActionResult CreateUpdateRegionType(UpdateRegionTypesModel model) | *model* – modelul acțiunii de creare sau actualizare a tipului de regiune. | Acțiunea de creare sau actualizare a tipului de regiune corespunzător modelului. | Returnează acțiunea de creare sau actualizare a tipului de regiune corespunzător modelului. |
| 23. | public void DeleteRegionType(long id) | *id* – numărul de identificare al tipu-lui de regiune. |  | Eliminarea tipului de regiune cu numărul de identificare respectiv. |
| 24. | public void UnDeleteRegionType(long id) | *id* – numărul de identificare al tipu-lui de regiune. |  | Restabilirea tipului de regiune cu numărul de identificare respectiv. |
| 25. | public ActionResult CreateUpdatePersonStatus(long? personStatusId) | *personStatusId* – numărul de iden-tificare al statutului de persoană. | Acțiunea de creare sau actualizare a statutului de per-soană respectiv. | Returnează acțiunea de creare sau actualizare a statutului de persoană respectiv. |
| 26. | public ActionResult CreateUpdatePersonStatus(UpdatePersonStatusModel model) | *model* – modelul acțiunii de creare sau actualizare a statutului de persoană. | Acțiunea de creare sau actualizare a statutului de per-soană corespun-zător modelului. | Returnează acțiunea de creare sau actualizare a statutului de persoană corespunzător modelului. |
| 27. | public void DeletePersonStatus(long id) | *id* – numărul de identificare al statutului de persoană. |  | Eliminarea statutului de persoană cu numărul de identificare respectiv. |
| 28. | public void UnDeletePersonStatus(long id) | *id* – numărul de identificare al statutului de persoană. |  | Restabilirea statutului de persoană cu numărul de identificare respectiv. |
| 29. | public ActionResult CreateUpdateGender(long? genderId) | *genderId* – numărul de identificare al genului. | Acțiunea de creare sau actualizare a genului. | Returnează acțiunea de creare sau actualizare a genului. |
| 30. | public ActionResult CreateUpdateGender(UpdateLookupModel model) | *model* – modelul acțiunii de creare sau actualizare a genului. | Acțiunea de creare sau actualizare a genului corespun-zător modelului. | Returnează acțiunea de creare sau actualizare a genului corespunzător modelului. |
| 31. | public void DeleteGender(long id) | *id* – numărul de identificare al genului. |  | Eliminarea genului cu numărul de identificare respectiv. |
| 32. | public void UnDeleteGender(long id) | *id* – numărul de identificare al genului. |  | Restabilirea genului cu numărul de identificare respectiv. |
| 33. | public ActionResult CreateUpdateStreet(long regionId, long? streetId) | *regionId* – numărul de identificare al regiunii;  *streetId* – numărul de identificare al străzii | Acțiunea de creare sau actualizare a străzii în regiunea respectivă. | Returnează acțiunea de creare sau actualizare a străzii în regiunea respectivă. |
| 34. | public ActionResult CreateUpdateStreet(UpdateStreetModel model) | *model* – modelul acțiunii de creare sau actualizare a străzii. | Acțiunea de creare sau actualizare a străzii corespun-zătoare modelului. | Returnează acțiunea de creare sau actualizare a străzii corespunzătoare modelului. |
| 35. | public void DeleteStreet(long streetId) | *streetId* – numărul de identificare al străzii. |  | Eliminarea străzii cu numărul de identificare respectiv. |
| 36. | public void UnDeleteStreet(long id) | *id* – numărul de identificare al străzii. |  | Restabilirea străzii cu numărul de identificare respectiv. |
| 37. | public ActionResult CreateUpdatePersonAddressType(long? personAddressTypeId) | *personAddressTypeId* – numărul de identificare al tipului de adresă de persoană | Acțiunea de creare sau actualizare a tipului de adresă de persoană. | Returnează acțiunea de creare sau actualizare a tipului de adresă de persoană. |
| 38. | public ActionResult CreateUpdatePersonAddressType(UpdateLookupModel model) | *model* – modelul acțiunii de creare sau actualizare a tipului de adresă de persoană. | Acțiunea de creare sau actualizare a tipului de adresă de persoană corespunzător modelului. | Returnează acțiunea de creare sau actualizare a tipului de adresă de persoană cores-punzător modelului. |
| 39. | public ActionResult CreateUpdateDocType(long? docTypeId) | *docTypeId* – numărul de identificare al tipului de document | Acțiunea de creare sau actualizare a tipului de document. | Returnează acțiunea de creare sau actualizare a tipului de document. |
| 40. | public ActionResult CreateUpdateDocType(UpdateDocumentTypesModel model) | *model* – modelul acțiunii de creare sau actualizare a tipului de document. | Acțiunea de creare sau actualizare a tipului de document corespunzător modelului. | Returnează acțiunea de creare sau actualizare a tipului de document corespunzător modelului. |
| 41. | public void DeleteDocType(long id) | *streetId* – numărul de identificare al tipului de document. |  | Eliminarea tipului de document cu numărul de identificare respectiv. |
| 42. | public void UnDeleteDocType(long id) | *id* – numărul de identificare al tipului de document. |  | Restabilirea tipului de document cu numărul de identificare respectiv. |
| 43. | public JqGridJsonResult ListElectionTypesAjax(JqGridRequest request) | *request* – informații despre pagina tabelei cerute de la server | Pagina tabelei cerută de la server, încărcată cu o listă de tipuri de scrutine electorale, conform cererii. | Returnează pagina tabelei cerută de la server, încărcată cu o listă de tipuri de scrutine electorale, conform cererii. |
| 44. | public ActionResult ExportElectionTypes(JqGridRequest request, ExportType exportType) | *request* – informații despre pagina tabelei cerute de la server;  *exportType* – tipul de exportare. | Acțiunea de exportare a paginii de tipuri de scrutine electorale, conform cererii. | Returnează acțiunea de exportare a paginii de tipuri de scrutine electorale, conform cererii. |
| 45. | public ActionResult CreateUpdateElectionType(long? electionTypeId) | *electionTypeId* – numărul de identificare al tipului de scrutin electoral | Acțiunea de creare sau actualizare a tipului de scrutin electoral. | Returnează acțiunea de creare sau actualizare a tipului de scrutin electoral. |
| 46. | public ActionResult CreateUpdateElectionType(UpdateLookupModel model) | *model* – modelul acțiunii de creare sau actualizare a tipului de scrutin electoral. | Acțiunea de creare sau actualizare a tipului de scrutin electoral corespunzător modelului. | Returnează acțiunea de creare sau actualizare a tipului de scrutin electoral corespunzător modelului. |
| 47. | public void DeleteElectionType(long id) | *id* – numărul de identificare al tipului de scrutin electoral. |  | Eliminarea tipului de scrutin electoral cu numărul de identificare respectiv. |
| 48. | public void UnDeleteElectionType(long id) | *id* – numărul de identificare al tipului de scrutin electoral. |  | Restabilirea tipului de scrutin electoral cu numărul de identificare respectiv. |
| 49. | public JqGridJsonResult ListPersonAddressTypesAjax(JqGridRequest request) | *request* – informații despre pagina tabelei cerute de la server | Pagina tabelei cerută de la server, încărcată cu o listă de tipuri de adresă de persoană, conform cererii. | Returnează pagina tabelei cerută de la server, încărcată cu o listă de tipuri de adresă de persoană, conform cererii. |
| 50. | public ActionResult ExportPersonAddressTypes(JqGridRequest request, ExportType exportType) | *request* – informații despre pagina tabelei cerute de la server;  *exportType* – tipul de exportare. | Acțiunea de exportare a paginii de tipuri de adresă de persoană, conform cererii. | Returnează acțiunea de exportare a paginii de tipuri de adresă de persoană, conform cererii. |
| 51. | public void DeletePersonAddressType(long id) | *id* – numărul de identificare al tipului de adresă de persoană. |  | Eliminarea tipului de adresă de persoană cu numărul de identificare respectiv. |
| 52. | public void UnDeletePersonAddressType(long id) | *id* – numărul de identificare al tipului de adresă de persoană. |  | Restabilirea tipului de adresă de persoană cu numărul de identificare respectiv. |
| 53. | public JqGridJsonResult ListStreetTypeAjax(JqGridRequest request) | *request* – informații despre pagina tabelei cerute de la server | Pagina tabelei cerută de la server, încărcată cu o listă de tipuri de străzi, conform cererii. | Returnează pagina tabelei cerută de la server, încărcată cu o listă de tipuri de străzi, conform cererii. |
| 54. | public ActionResult ExportStreetTypes(JqGridRequest request, ExportType exportType) | *request* – informații despre pagina tabelei cerute de la server;  *exportType* – tipul de exportare. | Acțiunea de exportare a paginii de tipuri de străzi, conform cererii. | Returnează acțiunea de exportare a paginii de tipuri de străzi, conform cererii. |
| 55. | public JqGridJsonResult ListManagerTypesAjax(JqGridRequest request) | *request* – informații despre pagina tabelei cerute de la server | Pagina tabelei cerută de la server, încărcată cu o listă de tipuri de conducători, conform cererii. | Returnează pagina tabelei cerută de la server, încărcată cu o listă de tipuri de conducători, conform cererii. |
| 56. | public ActionResult ExportManagerTypes(JqGridRequest request, ExportType exportType) | *request* – informații despre pagina tabelei cerute de la server;  *exportType* – tipul de exportare. | Acțiunea de exportare a paginii de tipuri de conducători, conform cererii. | Returnează acțiunea de exportare a paginii de tipuri de conducători, conform cererii. |
| 57. | public JqGridJsonResult ListCircumscriptionsAjax(JqGridRequest request) | *request* – informații despre pagina tabelei cerute de la server | Pagina tabelei cerută de la server, încărcată cu o listă de regiuni ce reprezintă circumscripții, conform cererii. | Returnează pagina tabelei cerută de la server, încărcată cu o listă de regiuni ce reprezintă circumscripții, conform cererii. |
| 58. | public ActionResult ExportCircumscriptions(JqGridRequest request, ExportType exportType) | *request* – informații despre pagina tabelei cerute de la server;  *exportType* – tipul de exportare. | Acțiunea de exportare a paginii de regiuni ce reprezintă circumscripții, conform cererii. | Returnează acțiunea de exportare a paginii de regiuni ce reprezintă circumscripții, conform cererii. |
| 59. | public JqGridJsonResult ListGendersAjax(JqGridRequest request) | *request* – informații despre pagina tabelei cerute de la server | Pagina tabelei cerută de la server, încărcată cu o listă de genuri, conform cererii. | Returnează pagina tabelei cerută de la server, încărcată cu o listă de genuri, conform cererii. |
| 60. | public ActionResult ExportGenders(JqGridRequest request, ExportType exportType) | *request* – informații despre pagina tabelei cerute de la server;  *exportType* – tipul de exportare. | Acțiunea de exportare a paginii de genuri, conform cererii. | Returnează acțiunea de exportare a paginii de genuri, conform cererii. |
| 61. | public JqGridJsonResult ListRegionTypesAjax(JqGridRequest request) | *request* – informații despre pagina tabelei cerute de la server | Pagina tabelei cerută de la server, încărcată cu o listă de tipuri de regiuni, conform cererii. | Returnează pagina tabelei cerută de la server, încărcată cu o listă de tipuri de regiuni, conform cererii. |
| 62. | public ActionResult ExportRegionTypes(JqGridRequest request, ExportType exportType) | *request* – informații despre pagina tabelei cerute de la server;  *exportType* – tipul de exportare. | Acțiunea de exportare a paginii de tipuri de regiuni, conform cererii. | Returnează acțiunea de exportare a paginii de tipuri de regiuni, conform cererii. |
| 63. | public JqGridJsonResult ListPersonStatusAjax(JqGridRequest request) | *request* – informații despre pagina tabelei cerute de la server | Pagina tabelei cerută de la server, încărcată cu o listă de statute de persoană, conform cererii. | Returnează pagina tabelei cerută de la server, încărcată cu o listă de statute de persoană, conform cererii. |
| 64. | public ActionResult ExportPersonStatuses(JqGridRequest request, ExportType exportType) | *request* – informații despre pagina tabelei cerute de la server;  *exportType* – tipul de exportare. | Acțiunea de exportare a paginii de statute de persoană, conform cererii. | Returnează acțiunea de exportare a paginii de statute de persoană, conform cererii. |
| 65. | public JqGridJsonResult ListDocTypesAjax(JqGridRequest request) | *request* – informații despre pagina tabelei cerute de la server | Pagina tabelei cerută de la server, încărcată cu o listă de tipuri de document, conform cererii. | Returnează pagina tabelei cerută de la server, încărcată cu o listă de tipuri de document, conform cererii. |
| 66. | public ActionResult ExportDocTypes(JqGridRequest request, ExportType exportType) | *request* – informații despre pagina tabelei cerute de la server;  *exportType* – tipul de exportare. | Acțiunea de exportare a paginii de tipuri de document, conform cererii. | Returnează acțiunea de exportare a paginii de tipuri de document, conform cererii. |
| 67. | public JqGridJsonResult ListTreeViewRegionsAjax(int? nodeid) | *nodeid* – numărul de identificare a nodului regiune din arbore | Lista subregiunilor nodului respectiv, serializată în format JSON. | Returnează lista subregiunilor nodului respectiv, serializată în format JSON. |
| 68. | public JqGridJsonResult ListStreetsAjax(JqGridRequest request, long? regionId) | *request* – informații despre pagina tabelei cerute de la server;  *regionId* – numărul de identificare al regiunii. | Pagina tabelei cerută de la server, încărcată cu o listă de străzi din regiunea respectivă, conform cererii. | Returnează pagina tabelei cerută de la server, încărcată cu o listă de străzi din regiunea respectivă, conform cererii. |
| 69. | public ActionResult ExportStreets(JqGridRequest request, ExportType exportType, long? regionId) | *request* – informații despre pagina tabelei cerute de la server;  *exportType* – tipul de exportare;  *regionId* – numărul de identificare al regiunii. | Acțiunea de exportare a paginii de străzi din regiunea respectivă, conform cererii. | Returnează acțiunea de exportare a paginii de străzi din regiunea respectivă, conform cererii. |
| 70. | public JqGridJsonResult ListRegionsTreeAjax(JqGridRequest request, long? parentId) | *request* – informații despre pagina tabelei cerute de la server;  *parentId* – numărul de identificare al regiunii părinte. | Pagina tabelei cerută de la server, încărcată cu o listă de subregiuni ale regiunii părinte, conform cererii. | Returnează pagina tabelei cerută de la server, încărcată cu o listă de subregiuni ale regiunii părinte, conform cererii. |
| 71. | public JqGridJsonResult ListRegionsGridAjax(JqGridRequest request) | *request* – informații despre pagina tabelei cerute de la server. | Pagina tabelei cerută de la server, încărcată cu o listă de regiuni, conform cererii. | Returnează pagina tabelei cerută de la server, încărcată cu o listă de regiuni, conform cererii. |
| 72. | public ActionResult ExportRegions(JqGridRequest request, ExportType exportType, long? parentId) | *request* – informații despre pagina tabelei cerute de la server;  *exportType* – tipul de exportare;  *parentId* – numărul de identificare al regiunii părinte. | Acțiunea de exportare a paginii de subregiuni ale regiunii părinte, conform cererii. | Returnează acțiunea de exportare a paginii de subregiuni ale regiunii părinte, conform cererii. |
| 73. | public ActionResult CreateUpdateRegion(long parentId, long? regionId) | *parentId* – numărul de identificare al regiunii părinte;  *regionId* – numărul de identificare al regiunii. | Acțiunea de creare sau actualizare a regiunii. | Returnează acțiunea de creare sau actualizare a regiunii. |
| 74. | public ActionResult CreateUpdateRegion(UpdateRegionModel model) | *model* – modelul acțiunii de creare sau actualizare a regiunii. | Acțiunea de creare sau actualizare a regiunii corespunzătoare modelului. | Returnează acțiunea de creare sau actualizare a regiunii corespunzătoare modelului. |
| 75. | public void DeleteRegion(long id) | *id* – numărul de identificare al regiunii. |  | Eliminarea regiunii cu numărul de identificare respectiv. |
| 76. | public void UnDeleteRegion(long id) | *id* – numărul de identificare al regiunii. |  | Restabilirea regiunii cu numărul de identificare respectiv. |
| 77. | public ActionResult SelectRegionType(long? regionId) | *regionId* – numărul de identificare al regiunii. | Acțiunea de selectare a tipurilor de regiuni în baza regiunii. | Returnează acțiunea de selectare a tipurilor de regiuni în baza regiunii. |
| 78. | public IList<RegionType> GetRegionTypes(long? parentId) | *parentId* – numărul de identificare al regiunii părinte. | Tipurile de regiuni în baza regiunii părinte. | Returnează tipurile de regiuni în baza regiunii părinte. |
| 79. | public ActionResult SelectYesNo() |  | Acțiunea de selectare a valorii logice (*Yes* sau *No*) | Returnează acțiunea de selectare a valorii logice (*Yes* sau *No*). |
| 80. | public ActionResult SelectStreetType() |  | Acțiunea de selectare a tipului de stradă. | Returnează acțiunea de selectare a tipului de stradă. |
| 81. | public ActionResult UpdateAdministrativeInfo(long regionId) | *regionId* – numărul de identificare al regiunii. | Acțiunea de actualizare a informației despre administrația regiunii respective. | Returnează acțiunea de actualizare a informației despre administrația regiunii respective. |
| 82. | public ActionResult UpdateAdministrativeInfo(UpdatePublicAdministrationModel model) | *model* – modelul acțiunii de actualizare a informației despre administrația regiunii. | Acțiunea de actualizare a informației despre administrația regiunii corespunzătoare modelului. | Returnează acțiunea de actualizare a informației despre administrația regiunii corespunzătoare modelului. |
| 83. | public ActionResult UpdateCircumscription(long circumscriptionId) | *circumscriptionId* – numărul de identificare al regiunii care reprezintă o circumscripție | Acțiunea de actualizare a circumscripției date. | Returnează acțiunea de actualizare a circumscripției date. |
| 84. | public ActionResult UpdateCircumscription(UpdateCircumscriptionModel model) | *model* – modelul acțiunii de actualizare a circumscripției. | Acțiunea de actualizare a circumscripției corespunzătoare modelului. | Returnează acțiunea de actualizare a circumscripției corespunzătoare modelului. |
| 85. | public ActionResult LinkRegions(long regionId) | *regionId –* numărul de identificare al regiunii | Acțiunea de vizualizare a regiunilor legate ce au legătură cu regiunea dată. | Returnează acțiunea de vizualizare a regiunilor legate ce au legătură cu regiunea dată. |
| 86. | public ActionResult LinkRegions(long regionId, long[] linkedRegionIds) | *regionId –* numărul de identificare al regiunii;  *linkedRegionIds* – lista regiunilor legate | Acțiunea de legare a regiunii respective cu regiunile date. | Returnează acțiunea de legare a regiunii respective cu regiunile date. |
| 87. | public JsonResult GetLinkedRegions(Select2Request request) | *request* – informații despre pagina tabelei cerute de la server. | Pagina de regiuni legate, conform cererii efectuate, serializată în format JSON. | Returnează pagina de regiuni legate, conform cererii efectuate, serializată în format JSON. |
| 88. | public ActionResult StreetRspTypes() |  | Acțiunea de vizu-alizare a tipurilor de străzi din Registrul de Stat al Populației | Returnează acțiunea de vizu-alizare a tipurilor de străzi din Registrul de Stat al Populației. |
| 89. | public JqGridJsonResult ListStreetRspTypeAjax(JqGridRequest request) | *request* – informații despre pagina tabelei cerute de la server | Pagina tabelei cerută de la server, încărcată cu o listă de tipuri de străzi din Registrul de Stat al Populației, conform cererii. | Returnează pagina tabelei cerută de la server, încărcată cu o listă de tipuri de străzi din Registrul de Stat al Populației, conform cererii. |
| 90. | public ActionResult ExportStreetRspTypes(JqGridRequest request, ExportType exportType) | *request* – informații despre pagina tabelei cerute de la server;  *exportType* – tipul de exportare. | Acțiunea de exportare a paginii de tipuri de străzi din Registrul de Stat al Populației, conform cererii. | Returnează acțiunea de exportare a paginii de tipuri de străzi din Registrul de Stat al Populației, conform cererii. |

# Componenta CEC.Web.API

## Subcomponenta Home Controller

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr. | Funcționalitate | Date de intrare | Date de ieșire | Descriere |
| 1. | public ActionResult Index() |  | Pagina principală. | Returnează pagina principală. |

## Subcomponenta Polling Station Controller

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr. | Funcționalitate | Date de intrare | Date de ieșire | Descriere |
| 1. | public async Task<PersonInfo> GetAssigned(CheckTheListRequest checkTheListRequest) | *checkTheListRequest* – cererea trimisă serverului | Informații despre persoana cu numărul de identitate din cerere. | Verifică codul *Recaptcha* de securitate conform cererii și, în caz de succes, returnează asincron informații despre persoana cu numărul de identitate respectiv. |
| 2. | public string GetRecaptchaPublicKey () |  | Cheia publică *Recaptcha*. | Returnează cheia publică *Recaptcha*. |
| 3. | public PersonFullInfo GetAssignedForRegistration(string idnp) | *idnp* – numărul de identitate al persoanei | Informații complete despre persoana cu numărul de identitate respectiv. | Caută persoana cu numărul de identitate respectiv. Ulterior, se verifică dacă persoana respectivă este deja înregistrată pentru scrutinul electoral peste hotarele țării. Dacă nu este înregistrată, se returnează informații despre această persoană, în caz contrar se returnează un mesaj de avertizare. |
| 4. | public ElectionInfo GetCurrentActiveElection() |  | Informații despre scrutinul electoral curent activ. | Returnează informații despre scrutinul electoral curent activ. |
| 5. | public bool GetAbroadDeclaration() |  | Indicatorul faptului că există un scrutin electoral curent. | Verifică dacă există un scrutin electoral curent. |
| 6. | public void SaveReport(ReportModel model) | *model* – modelul raportului declara-ției de ședere peste hotare. |  | Salvarea declarației de ședere peste hotare în baza informației din model și notificarea persoanei respec-tive prin e-mail. |

## Subcomponenta Statistic Controller

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr. | Funcționalitate | Date de intrare | Date de ieșire | Descriere |
| 1. | public IEnumerable <StatisticInfo> GetAssigned() |  | Lista de informații statistice referi-toare la pozițiile geolocale a adre-selor persoanelor cu drept de vot din străinătate care au depus declarație de ședere. | Returnează lista de informații statistice referitoare la pozițiile geolocale a adre-selor persoanelor cu drept de vot din străinătate care au depus declarație de ședere. |
| 2. | public IEnumerable <StatisticGrouped> GetVotersRegistration () |  | Lista de informații statistice grupate referitoare la nu-mărul de alegători care au depus declarație de ședere în fiecare țară. | Returnează lista de informații statistice grupate referitoare la numărul de alegători care au depus declarație de ședere în fiecare țară. |
| 3. | public IEnumerable <StatisticDateGrouped> GetTimelineStatistics () |  | Lista de informații statistice grupate referitoare la nu-mărul de alegători care au depus de-clarație de ședere în străinătate la fiecare dată. | Returnează lista de informații statistice grupate referitoare la numărul de alegători care au depus declarație de ședere în străinătate la fiecare dată. |

## Subcomponenta Where To Vote Controller

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr. | Funcționalitate | Date de intrare | Date de ieșire | Descriere |
| 1. | public IEnumerable <IdentityInfo> GetRegionsOfLevel1 (string filter) | *filter* – filtrul de căutare a regiunilor | Informații despre regiunile de pe nivelul 1 care conțin filtrul respectiv. | Returnează informații despre regiunile de pe nivelul 1 care conțin filtrul respectiv. |
| 2. | public IEnumerable <IdentityInfo> GetRegions(long parentId, string filter) | *parentId* – numărul de identificare al regiunii părinte;  *filter* – filtrul de căutare a regiunilor | Informații despre subregiunile regiunii părinte date care conțin filtrul respectiv. | Returnează informații despre subregiunile regiunii părinte date care conțin filtrul respectiv. |
| 3. | public JsonResult GetRegions (StreetRequest streetRequest) | *streetRequest* – informații despre pagina de străzi cerută de la server. | Pagina cerută de la server, încărcată cu o listă de străzi din regiunea indicată în cerere. | Returnează pagina cerută de la server, încărcată cu o listă de străzi din regiunea indicată în cerere. |
| 4. | public Boolean RegionHasStreets(long regionId) | *regionId* – numărul de identificare al regiunii | Indicatorul fap-tului că regiunea cu numărul de identificare res-pectiv are străzi. | Verifică dacă regiunea cu numărul de identificare res-pectiv are străzi. |
| 5. | public IEnumerable <PollingStationInfo> GetPollingStations (long regionId) | *regionId* – numărul de identificare al regiunii | Informații despre secțiile de votare din regiunea res-pectivă. | Returnează informații despre secțiile de votare din regiunea respectivă. |
| 6. | public IEnumerable <AddressInfo> GetAddresses(long streetId) | *streetId* – numărul de identificare al străzii | Informații despre adresele de pe strada respectivă. | Returnează informații despre adresele de pe strada respectivă. |

# Componenta CEC.Web.Results.API

## Subcomponenta Home Controller

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr. | Funcționalitate | Date de intrare | Date de ieșire | Descriere |
| 1. | public ActionResult Index() |  | Pagina principală. | Returnează pagina principală. |

## Subcomponenta Image Route Handler

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr. | Funcționalitate | Date de intrare | Date de ieșire | Descriere |
| 1. | public IHttpHandler GetHttpHandler (RequestContext requestContext) | *requestContext* – cererea efectuată | O nouă instanță de *PolitPartyLogoImageHandler* | Returnează o nouă instanță de *PolitPartyLogoImageHandler* |

## Subcomponenta Live Election Turnout Controller

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr. | Funcționalitate | Date de intrare | Date de ieșire | Descriere |
| 1. | public HttpResponseMessage Get() |  | Un mesaj HTTP de răspuns care indică că protoco-lul sau versiunea protocolului a fost modificată. | Se instanțiază un handler *LiveElectionTurnoutHandler*, care se acceptă de contextul curent HTTP în calitate de *WebSocketHandler*. Se returnează un mesaj HTTP de răspuns care indică că protocolul sau versiunea protocolului a fost modifi-cată. |

## Subcomponenta Live Election Turnout Handler

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr. | Funcționalitate | Date de intrare | Date de ieșire | Descriere |
| 1. | public override void OnOpen() |  |  | Execută acțiuni ce necesită a fi efectuate la deschiderea conexiunii (adăugarea handler-ului curent în calitate de client, inițializarea timpului, încărcarea informației despre votare, etc.). |
| 2. | public override void OnClose() |  |  | Execută acțiuni ce necesită a fi efectuate la închiderea conexiunii (eliminarea hand-ler-ului curent din lista de clienți). |
| 3. | public override void OnMessage(string message) | *message* – mesajul |  | Execută acțiuni ce necesită a fi efectuate la transmiterea unui mesaj în cadrul cone-xiunii (nici o acțiune spe-cială). |

## Subcomponenta Live Results Controller

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr. | Funcționalitate | Date de intrare | Date de ieșire | Descriere |
| 1. | public HttpResponseMessage Get(Int64 pollId) | *pollId* – numărul de identificare | Un mesaj HTTP de răspuns care indică că protoco-lul sau versiunea protocolului a fost modificată. | Se instanțiază un handler *LiveResultsHandler*, care se acceptă de contextul curent HTTP în calitate de *WebSocketHandler*. Se retur-nează un mesaj HTTP de răspuns care indică că protocolul sau versiunea protocolului a fost modifi-cată. |

## Subcomponenta Live Results Handler

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr. | Funcționalitate | Date de intrare | Date de ieșire | Descriere |
| 1. | public override void OnOpen() |  |  | Execută acțiuni ce necesită a fi efectuate la deschiderea conexiunii (adăugarea handler-ului curent în calitate de client, inițializarea timpului, încărcarea informației despre votare, etc.). |
| 2. | public override void OnClose() |  |  | Execută acțiuni ce necesită a fi efectuate la închiderea conexiunii (eliminarea hand-ler-ului curent din lista de clienți). |
| 3. | public override void OnMessage(string message) | *message* – mesajul |  | Execută acțiuni ce necesită a fi efectuate la transmiterea unui mesaj în cadrul cone-xiunii (nici o acțiune spe-cială). |
| 4. | DateTime VotingHours(DateTime votingDay, int hourIn, int minuteIn) | *votingDay* – ziua votării;  *hourIn* – numărul de ore;  *minuteIn* – numărul de minute | Momentul de timp din ziua votării, după scurgere numărului de ore și minute dat. | Returnează momentul de timp din ziua votării, după scurgere numărului de ore și minute dat. |

## Subcomponenta Polit Party Logo Image Handler

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr. | Funcționalitate | Date de intrare | Date de ieșire | Descriere |
| 1. | public void ProcessRequest(HttpContext context) | *context –* contextul HTTP |  | Procesarea cererii prin desenarea insignei fiecărui partid. |
| 2. | public bool IsReusable |  | Adevăr | Returnează adevăr. |
| 3. | public static Image ByteArrayToImage( byte[] imageByte) | *imageByte* – imaginea, în formă de șir de biți | Imaginea reală. | Transformă șirul de biți în imagine. |
| 4. | public static byte[] ImageToByteArray (Bitmap bmp) | *bmp* - imaginea | Șirul de biți, care păstrează imaginea dată | Transformă o imagine în șir de biți. |
| 5. | public static byte[] LoadResource(Assembly assembly, string resource) | *assembly –* asamblarea;  *resource* – denumirea resursei | Imaginea ca șir de biți, încărcată din resursa respective a asamblării date. | Returnează imaginea ca șir de biți, încărcată din resursa respectivă a asamblării date. |

## Subcomponenta Live Results Logic

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr. | Funcționalitate | Date de intrare | Date de ieșire | Descriere |
| 1. | public List<VotersActivity> GeneralOnTimeResults (DateTime openVote, DateTime currentTime, int interval) | *openVote* – momentul de timp când s-a deschis votarea;  *currentTime* – momentul de timp curent;  *interval* – intervalul de timp. | Lista de activități ale alegătorilor (numărul de alegători la diverse momente de timp, repartizate la un anumit interval) | Returnează lista de activități ale alegătorilor (numărul de alegători la diverse momente de timp, repartizate la un anumit interval). |
| 2. | public List<VotersActivity> GeneralOnTimeAddResults(DateTime openVote, DateTime currentTime, int interval) | *openVote* – momentul de timp când s-a deschis votarea;  *currentTime* – momentul de timp curent;  *interval* – intervalul de timp. | Lista de activități ale noilor alegători (numărul de alegători la diverse momente de timp, repartizate la un anumit interval) | Returnează lista de activități ale noilor alegători (numărul de alegători la diverse momente de timp, repartizate la un anumit interval). |
| 3. | public int GeneralTotalVoters() |  | Numărul total de alegători. | Returnează numărul total de alegători. |
| 4. | public List<  StatPreliminaryResult> GeneralPartidResults() |  | Rezultatele preliminare obținute de partide. | Returnează rezultatele preliminare obținute de partide. |
| 5. | public DateTime VotingDayById() |  | Data votării scrutinului electoral curent | Returnează data votării scrutinului electoral curent. |
| 6. | public byte[] LogoByCandidateId(int id) | *id* – numărul de identificare al candidatului | Logoul candidatului respectiv | Returnează logoul candidatului respectiv, în formă de șir de biți. |
| 7. | public static string HexToWebImage(byte[] hex) | *hex* – imaginea în formă de șir de biți | Șirul de biți, transformat în șir de caractere în baza 64 | Transformă șirul de biți în șir de caractere în baza 64. |
| 8. | public List<AssignedPollingStation> GetAssignedPollingStations() |  | Lista de secții de votare asociata scrutinului electoral curent. | Returnează lista de secții de votare asociata scrutinului electoral curent. |
| 9. | public int GetTotalVotersReceivedBallotsPerPollingStation(long pollingStation) | *pollingStation* – numărul de identificare al secției de votare | Numărul total de buletine de vot primite de alegătorii secției de votare respective. | Returnează numărul total de buletine de vot primite de alegătorii secției de votare respective. |
| 10. | public int GetTotalVotersInSupplimentaryListPerPollingStation(long pollingStation) | *pollingStation* – numărul de identificare al secției de votare | Numărul total de alegători în lista suplimentară în cadrul secției de votare respective. | Returnează numărul total de alegători în lista suplimentară în cadrul secției de votare respective. |
| 11. | public BallotPapersCountData GetBallotPapers() |  | Numărul total de buletine de vot. | Returnează numărul total de buletine de vot. |
| 12. | public TotalVotesData GetTotalValidVotes() |  | Numărul total de buletine de vot valide. | Returnează numărul total de buletine de vot valide. |

## Subcomponenta Sql Factory

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr. | Funcționalitate | Date de intrare | Date de ieșire | Descriere |
| 1. | public IDbConnection OpenConnect() |  | Conexiunea deschisă | Deschide o conexiune cu baza de date ce păstrează rezultatele alegerilor și returnează conexiunea deschisă. |
| 2. | IDbCommand CreateSPCommand (IDbConnection connection, string procNameWithParams) | *connection* – conexiunea;  *procNameWithParams* – denumirea procedurii stocate cu parametri | Comanda creată | Creează o comandă capabilă să execute procedura stocată cu parametri dată pe baza de date reprezentată de conexiunea respectivă. |
| 3. | public List<VotersActivity> GeneralOnTimeResults (int electionId, DateTime startTime,  DateTime endTime,  int? interval) | *electionId* – numărul de identificare al scrutinului electoral;  *startTime* – momentul de timp de început;  *endTime* – momentul de timp de sfârșit;  *interval* – intervalul de timp. | lista de activități de votare în perioada respectivă la intervalul dat de timp. | Returnează lista de activități de votare în perioada respectivă la intervalul dat de timp. |
| 4. | public List<VotersActivity> GeneralOnTimeAddResults(int electionId, DateTime openVote, DateTime currentTime, int interval) | *electionId* – numărul de identificare al scrutinului electoral;  *openVote* – momentul de timp de început al procesului de votare;  *currentTime* – momentul de timp curent;  *interval* – intervalul de timp. | lista de activități noi de votare în perioada respectivă la intervalul dat de timp. | Returnează lista de activități noi de votare în perioada respectivă la intervalul dat de timp. |
| 5. | public int GeneralTotalVoters(int electionId) | *electionId* – numărul de identificare al scrutinului electoral | Numărul total de alegători la scrutinul electoral respectiv. | Returnează numărul total de alegători la scrutinul electoral respectiv. |
| 6. | public List<GeneralPerPartidsResult> GeneralPartidResults (int electionId) | *electionId* – numărul de identificare al scrutinului electoral | Rezultatele generale pe partide la scrutinul electoral respectiv. | Returnează rezultatele generale pe partide la scrutinul electoral respectiv. |
| 7. | public DateTime VotingDayById(int electionId) | *electionId* – numărul de identificare al scrutinului electoral | Data scrutinului electoral respectiv. | Returnează data scrutinului electoral respectiv. |
| 8. | public byte[] LogoByCandidateId(int id) | *id* – numărul de identificare al candidatului | Sigla candidatului dat ca șir de biți | Returnează sigla candidatului dat ca șir de biți. |
| 9. | public List<TResult> ReadDbElementsList<TResult>(string procName) | *TResult* – tipul generic al elemen-telor returnate de procedura stocată;  *procName* – denumirea proce-durii stocate | Lista de elemente returnată de pro-cedura stocată. | Returnează lista de elemente returnată de procedura sto-cată. |
| 10. | public TResult ReadDbElement<TResult>(string procName) | *TResult* – tipul generic al elemen-telor returnate de procedura stocată;  *procName* – denumirea proce-durii stocate | Un element retur-nat de procedura stocată. | Returnează un element retur-nat de procedura stocată. |
| 11. | public object ReadDbElement(string procName) | *procName* – denumirea proce-durii stocate | Un element retur-nat de procedura stocată. | Returnează un element retur-nat de procedura stocată. |
| 12. | public List<AssignedPollingStation> GetAssignedPollingStations(int electionId) | *electionId* – numărul de identificare al scrutinului electoral | Lista de secții de votare asociate scrutinului electoral dat. | Returnează lista de secții de votare asociate scrutinului electoral dat. |
| 13. | public int GetTotalVotersReceivedBallotsPerPollingStation(int electionId, long pollingStation) | *electionId* – numărul de identi-ficare al scrutinului electoral;  *pollingStation* – numărul de identi-ficare al secției de votare | Numărul total de buletine de vot primite de alegătorii secției de votare respective în cadrul scrutinului electoral dat. | Returnează numărul total de buletine de vot primite de alegătorii secției de votare respective în cadrul scrutinului electoral dat. |
| 14. | public int GetTotalVotersInSupplimentaryListPerPollingStation(int electionId, long pollingStation) | *electionId* – numărul de identi-ficare al scrutinului electoral;  *pollingStation* – numărul de identi-ficare al secției de votare | Numărul total de alegători pe lista suplimentară în secția de votare respectivă la scrutinul electoral dat. | Returnează numărul total de alegători pe lista suplimentară în secția de votare respectivă la scrutinul electoral dat. |
| 15. | public List<BallotPapersCountData> GetBallotPapers(int electionId) | *electionId* – numărul de identi-ficare al scrutinului electoral | Numărul total de buletine de vot în cadrul scrutinului electoral dat. | Returnează numărul total de buletine de vot în cadrul scrutinului electoral dat. |
| 16. | public List<TotalVotesData> GetTotalValidVotes(int electionId) | *electionId* – numărul de identi-ficare al scrutinului electoral | Numărul total de buletine de vot valide la scrutinul electoral dat. | Returnează numărul total de buletine de vot valide la scrutinul electoral dat. |

# Componenta CEC.Web.Public.Modules

## Scriptul Java CEC

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr. | Funcționalitate | Date de intrare | Date de ieșire | Descriere |
| 1. | function updateClock() |  |  | Actualizează timpul. |
| 2. | function initCheckTheList() |  |  | Inițializează controalele pentru verificarea listei electorale. |
| 3. | function addValidationToIdnp(e) | *e* – parametru fictiv |  | Adaugarea regulilor de validare pentru IDNP. |
| 4. | function initGoogleMaps(lat, long, pollingStation) | *lat* – latitudinea;  *long* – longitudinea;  *pollingStation* – secția de votare |  | Inițializează harta Google, indicând secția de votare respectivă în poziția dată de latitudinea și longitudinea respectivă. |
| 5. | function getPersonByIdnp1() |  |  | Setează controalele pentru a extrage persoana după IDNP. |
| 6. | function getPersonPollingStation(json, idnp) | *json* – informații despre persoană, serializate în format JSON, obținute de la server;  *idnp* – numărul de identitate al persoanei, introdus de utilizator. |  | Încarcă controalele cu informațiile despre persoana respectivă. În caz de eroare, se afișează eroarea. |
| 7. | function showCriticalErrorMessage() |  |  | Afișarea erorii de accesare server. |
| 8. | function showInfoMessage(idnp, warningMessage, reCaptchaError) | *idnp* – numărul de identitate al persoanei;  *warningMessage* – mesajul de avertizare;  *reCaptchaError* –eroarea de la con-trolul ReCaptcha care furnizează cheia publică de securitate. |  | Afișarea mesajului de avertizare și a mesajului de informare că persoana cu numărul de identificare respectiv nu a fost găsită. |
| 9. | function showIncorrectIdnpMessage(idnp) | *idnp* – numărul de identitate al persoanei |  | Afișarea mesajului de incorectitudine a IDNP-ului introdus. |
| 10. | function checkForInputErrors(idnp) | *idnp* – numărul de identitate al persoanei | Indicatorul corectitudinii IDNP-ului introdus. | Verifică corectitudinea IDNP-ului introdus. În caz de incorectitudine se afișează mesajul de avertizare. |
| 11. | function showOrHideIdnpHelpImage() |  |  | Modifică vizibilitatea imaginii de ajutor în determinare IDNP-ului persoanei. |
| 12. | function clearFields() |  |  | Curăță valorile controalelor. |
| 13. | function hideMap() |  |  | Ascunde harta. |
| 14. | function showResultTables() |  |  | Afișarea tabelei cu rezultate. |
| 15. | function hideResultTables() |  |  | Ascunderea tabelei cu rezultate. |
| 16. | function showErrorByIdnp() |  |  | Afișarea controlului de căutare după IDNP. |
| 17. | function hideErrorByIdnp() |  |  | Ascunderea controlului de căutare după IDNP. |

## Scriptul Java CEC2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr. | Funcționalitate | Date de intrare | Date de ieșire | Descriere |
| 1. | function InitializeRegionControl() |  |  | Inițializarea controlului de selectare a regiunii. |
| 2. | function InitializeRegion2Control() |  |  | Inițializarea controlului de selectare a subregiunii. |
| 3. | function InitializeStreetControl() |  |  | Inițializarea controlului de selectare a străzii. |
| 4. | function getAddresses() |  |  | Extrage lista de adrese de pe strada selectată de la server. |
| 5. | function AddAddressesInTheList(addresses) | *addresses* – adresele obținute de la server |  | Afișarea adreselor în controale. |
| 6. | function addressSelected (address) | *address –* adresa selectată |  | Afișarea secției de votare a adresei la selectarea adresei date. |
| 7. | function getPollingStations() |  |  | Extrage lista de secții de votare din regiunea selectată de la server. |
| 8. | function AddPollingStationsInTheList(pollingStations) | *pollingStations* – secțiile de votare obținute de la server |  | Afișarea secțiilor de votare în controale. |
| 9. | function showPollingStationInfo(pollingStation) | *pollingStation* – secția de votare |  | Afișarea informației despre secția de votare dată. |
| 10. | function setPollingStationVisibility(visibility) | *visibility* – vizibilitatea secției de votare |  | Setarea vizibilității secției de votare, în corespundere cu valoarea de adevăr transmisă ca parametru. |
| 11. | function setStreetsVisibility (visibility) | *visibility* – vizibilitatea străzilor |  | Setarea vizibilității străzilor, în corespundere cu valoarea de adevăr transmisă ca parametru. |
| 12. | function setPollingStationsVisibility(visibility) | *visibility* – vizibilitatea secțiilor de votare |  | Setarea vizibilității secțiilor de votare, în corespundere cu valoarea de adevăr transmisă ca parametru. |
| 13. | function setInfoVisibility(pollingStationVisibility, streetsVisibility) | *pollingStationVisibility* – vizibilitatea secției de votare;  *streetsVisibility* – vizibilitatea străzilor. |  | Setarea vizibilității secției de votare și a străzilor, în corespundere cu valorile de adevăr transmise ca parametru. |
| 14. | function setVisibility(div, visibility) | *div* – controlul căruia i se setează vizibilitatea;  *visibility* – vizibilitatea. |  | Setarea vizibilității contro-lului respectiv în corespundere cu valoarea de adevăr a vizibilității transmisă ca parametru. |
| 15. | function clearPollingStationInfo() |  |  | Curățirea valorilor controa-lelor ce conțin informații despre secția de votare . |
| 16. | function clearErrors() |  |  | Curățirea valorilor controa-lelor ce conțin mesajele de eroare. |
| 17. | function normalizeInputSearch(filter) | *filter* - filtrul |  | Normalizarea filtrului de căutare. |
| 18. | function setStreetAndPollingStationsVisibility() |  |  | Setarea vizibilității străzii și secțiilor de votare în corespundere cu existența de străzi în regiunea selectată. |

## Scriptul Java Election Turnout

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr. | Funcționalitate | Date de intrare | Date de ieșire | Descriere |
| 1. | function Turnout() |  |  | Resetează datele statistice despre rezultatele alegerilor. |
| 2. | function onWsMessage(event) | *event –* evenimentul care se realizează |  | Încărcarea informației din *event.data.* |
| 3. | function Init3() |  |  | Inițializarea hărții. |
| 4. | function getMarkers() |  | Markerii de pe hartă corespun-zători secțiilor de votare. | Returnează markerii de pe hartă corespunzători secțiilor de votare. |
| 5. | function initChartStatistics() |  |  | Inițializarea diagramelor care prezintă date statistice. |
| 6. | function ShowErrorMessage() |  |  | Ascunde informația afișată atunci când apare un mesaj de eroare. |
| 7. | function drawColumnChart(voterOnBaseList, votersReceivedBallots) | *voterOnBaseList* – numărul de alegători în lista de bază;  *votersReceivedBallots* – numărul de alegători care au primit buletine de vot. |  | Desenează diagrama de tip Coloane care prezintă datele statistice despre activitatea de votare. |

## Scriptul Java Live Results

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr. | Funcționalitate | Date de intrare | Date de ieșire | Descriere |
| 1. | function StatInfo() |  |  | Resetează informația statis-tică despre rezultatele live ale procesului electoral. |
| 2. | function reloadCharts() |  |  | Reîncărcarea diagramelor. |
| 3. | function onWsOpen() |  |  | Acțiuni efectuate la deschiderea socketului web (nici o acțiune specială). |
| 4. | function onWsError(event) | *event* – evenimentul care se realizează |  | Acțiuni efectuate la apariția unei erori a socketului web (nici o acțiune specială). |
| 5. | function onWsMessage(event) | *event* – evenimentul care se realizează |  | Acțiuni efectuate la apariția unui mesaj din partea socketului web (încărcarea informației din *event.data*). |
| 6. | function onWsSend(callback) | *callback –* funcție apelată de socketul web. |  | Apelează funcția *callback* cu mesajul din control. |
| 7. | function MorrisChartCEC1 (graphControl) | *graphControl –* controlul care va conține diagrama. |  | Construiește diagrama ce prezintă numărul de alegători pe lista de bază și numărul de alegători pe lista suplimentară. |
| 8. | function MorrisChartCEC2 (graphControl) | *graphControl –* controlul care va conține diagrama. |  | Construiește diagrama ce prezintă procentajul de persoane care au votat și procentajul de persoane care nu au votat. |
| 9. | function MorrisChartCEC3 (graphControl) | *graphControl –* controlul care va conține diagrama. |  | Construiește diagrama ce prezintă rezultatele preliminare ale candidaților electorali. |
| 10. | function showNoPreliminaryMessage() |  |  | Afișează mesajul de imposibilitate de accesare a rezultatelor preliminare înainte de începerea calculării acestora. |
| 11. | function existMorris (graphControl) | *graphControl –* controlul care va conține diagrama. | Indicatorul faptului că controlul grafic există. | Verifică dacă controlul grafic există. |

## Scriptul Java Preliminaries

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr. | Funcționalitate | Date de intrare | Date de ieșire | Descriere |
| 1. | function PreliminaryInfo() |  |  | Resetează informația statis-tică preliminară aferentă procesului electoral. |
| 2. | function ConnectWS() |  |  | Inițializează conexiunea socketului web. |
| 3. | function loadCharts (preliminary) | *preliminary* – infor-mații preliminare |  | Încărcarea diagramelor cu informația preliminară dată. |
| 4. | function onWsOpen() |  |  | Acțiuni efectuate la deschiderea socketului web (nici o acțiune specială). |
| 5. | function onWsError(event) | *event* – evenimentul care se realizează |  | Acțiuni efectuate la apariția unei erori a socketului web (inițializarea socketului web). |
| 6. | function onWsMessage(event) | *event* – evenimentul care se realizează |  | Acțiuni efectuate la apariția unui mesaj din partea socketului web (încărcarea informației din *event.data*). |
| 7. | function onWsSend(callback) | *callback –* funcție apelată de socketul web. |  | Apelează funcția *callback* cu mesajul din control. |
| 8. | function drawLabColumnChart (preliminaryData) | *preliminaryData* – informații preliminare |  | Construiește diagrama informațiilor preliminare. |
| 9. | function showMessageBeforPreliminary() |  |  | Afișează mesajul de imposibilitate de accesare a rezultatelor preliminare înainte de începerea calculării acestora. |

## Scriptul Java Register

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr. | Funcționalitate | Date de intrare | Date de ieșire | Descriere |
| 1. | function initialize() |  |  | Inițializarea completării automate a locației. |
| 2. | function LoadBootstrapValidatorScript(callback) | *callback –* funcția apelată. |  | Apelarea funcției date cu validarea preliminară de către *bootstrap*. |
| 3. | function DemoFormValidator() |  |  | Validarea adresei de e-mail. |
| 4. | function initCheckRegisterTheList() |  |  | Inițializarea validării înregistrării în străinătate. |
| 5. | function verificationIsSetAbroadDeclaration() |  |  | Verificarea dacă există deja declarația de ședere. |
| 6. | function initializeElectionName() |  |  | Inițializarea scrutinului electoral. |
| 7. | function getCurrentElection (json) | *json* – informații despre scrutinul electoral curent, serializate în format JSON, obținute de la server. |  | Încărcarea controalelor cu informații despre scrutinul electoral curent. |
| 8. | function saveAbroadRegistration() |  |  | Salvarea declarației de ședere în străinătate. |
| 9. | function addValidationToIdnp(e) | *e –* parametru fictiv |  | Adăugarea regulilor de validare pentru IDNP. |
| 10. | function getPersonFullInfoByIdnp() |  |  | Încărcarea controalelor cu informații complete despre persoană în baza IDNP-ului. |
| 11. | function getPersonFullInfoPollingStation(json, idnp) | *json –* informații despre persoană;  *idnp* – numărul de identitate al persoa-nei introdus de utilizator. |  | Încărcarea controalelor cu informații complete despre persoana cu numărul de identitate dat. În caz de neexistență a persoanei cu numărul de identitate dat se afișează un mesaj informativ. |
| 12. | function showCriticalErrorMessageOfVoter() |  |  | În caz de obținere a unui mesaj de eroare de la server, se afișează acest mesaj. |
| 13. | function showIncorrectIdnpMessageOfVoter(idnp) | *idnp* – numărul de identitate al persoa-nei. |  | Se afișează mesajul care indică faptul că IDNP-ul respectiv este incorect. |
| 14. | function checkForInputErrorsofVoters(idnp) | *idnp* – numărul de identitate al persoa-nei. | Indicatorul faptului că IDNP-ul respectiv este valid. | Verifică corectitudinea IDNP-ului dat. |

## Scriptul Java Statistics

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr. | Funcționalitate | Date de intrare | Date de ieșire | Descriere |
| 1. | function Init3() |  |  | Inițializează harta. |
| 2. | function getMarkers() |  | Markerii de pe hartă corespun-zători locațiilor prezente în statis-tici. | Returnează markerii de pe hartă corespunzători locațiilor prezente în statistici. |
| 3. | function initChartStatistics() |  |  | Inițializarea diagramei care prezintă date statistice referitoare la numărul de declarații depuse în fiecare țară. |
| 4. | function getStatsData() |  | Datele statistice grupate, obținute de la server. | Returnează datele statistice grupate, obținute de la server. |
| 5. | function ShowErrorMessage() |  |  | Afișează mesajul de eroare de afișare a datelor. |
| 6. | function drawChart() |  |  | Construirea diagramei care prezintă date statistice referitoare la numărul de declarații depuse la fiecare dată. |

# Componenta Quartz

## Subcomponenta CEC.QuartzServer.Console.Program

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr. | Funcționalitate | Date de intrare | Date de ieșire | Descriere |
| 1. | static void Main(string[] args) | *args –* parametrii aplicației consolă. |  | Pornește aplicația consolă care inițializează și startează serverul Quartz, destinat executării proceselor de lungă durată. |

## Subcomponenta CEC.QuartzServer.Core.CecQuartzServer

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr. | Funcționalitate | Date de intrare | Date de ieșire | Descriere |
| 1. | public void Initialize() |  |  | Inițializează planificatorul Quartz al serverului. |
| 2. | public void Start() |  |  | Startează execuția planifica-torului. |
| 3. | public void Stop() |  |  | Stopează execuția planifica-torului. |
| 4. | public void Pause() |  |  | Întrerupe temporar execuția planificatorului. |
| 5. | public void Resume() |  |  | Continuă execuția planifica-torului. |
| 6. | public void Dispose() |  |  | Acțiuni efectuate la distrugerea instanței serveru-lui (finisarea activităților planificatorului). |

## Subcomponenta CEC.QuartzServer.Core.Configuration

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr. | Funcționalitate | Date de intrare | Date de ieșire | Descriere |
| 1. | public static string ServiceName |  | Denumirea serviciului | Returnează denumirea servi-ciului. |
| 2. | public static string ServiceDisplayName |  | Denumirea de afi-șare a serviciului | Returnează denumirea de afișare a serviciului. |
| 3. | public static string ServiceDescription |  | Descrierea serviciului | Returnează descrierea serviciului. |
| 4. | public static string ServerImplementationType |  | Denumirea tipului de implementare a serviciului | Returnează denumirea tipului de implementare a serviciului |

## Subcomponenta CEC.QuartzServer.Core.JobBase

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr. | Funcționalitate | Date de intrare | Date de ieșire | Descriere |
| 1. | public void Execute (IJobExecutionContext context) | *context* – contextul de executare |  | Execută procesele din context |

## Subcomponenta CEC.QuartzServer.Core.QuartzSecurityContext

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr. | Funcționalitate | Date de intrare | Date de ieșire | Descriere |
| 1. | public IdentityUser GetCurrentUser() |  | Utilizatorul curent | Returnează utilizatorul curent |

## Subcomponenta CEC.QuartzServer.Core.QuartzServerFactory

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr. | Funcționalitate | Date de intrare | Date de ieșire | Descriere |
| 1. | public static IQuartzServer CreateServer() |  | Serverul Quartz creat | Creează un server Quartz și returnează serverul creat. |

## Subcomponenta CEC.QuartzServer.Core.SaiseDatabaseFactoryProvider

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr. | Funcționalitate | Date de intrare | Date de ieșire | Descriere |
| 1. | public static ISessionFactory GetFactory |  | Creatorul de se-siuni de conectare la aplicația SAISE | Construiește un creator de se-siuni de conectare la aplicația SAISE. |

## Subcomponenta CEC.QuartzServer.Core.UnityConfig

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr. | Funcționalitate | Date de intrare | Date de ieșire | Descriere |
| 1. | public static void RegisterComponents() |  |  | Înregistrarea componentelor (creatorul de sesiuni, tipurile, etc.). |

## Subcomponenta CEC.QuartzServer.Core.UnityJobFactory

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr. | Funcționalitate | Date de intrare | Date de ieșire | Descriere |
| 1. | public IJob NewJob(TriggerFiredBundle bundle, IScheduler scheduler) | *bundle* – pachetul de procese;  *scheduler* – planificatorul | Procesul nou în baza parametrilor dați. | Returnează procesul nou în baza parametrilor dați. |
| 2. | public void ReturnJob(IJob job) | *job –* procesul de lungă durată |  | Nici o acțiune specială. |

## Subcomponenta CEC.QuartzServer.Jobs.Export.SaiseExporterJob

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr. | Funcționalitate | Date de intrare | Date de ieșire | Descriere |
| 1. | protected internal override void ExecuteInternal (IJobExecutionContext context) | *context* – contextul de executare |  | Execută exportarea către SAISE în cadrul contextului dat. |
| 2. | public void Interrupt() |  |  | Întrerupe execuția exportării către SAISE. |

## Subcomponenta CEC.QuartzServer.Jobs.Export.SaiseExportStageExecuter

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr. | Funcționalitate | Date de intrare | Date de ieșire | Descriere |
| 1. | public void ExecuteState() |  |  | Execută etapa procesului și setează starea. |
| 2. | public SaiseExporterStage SaiseExporterStage |  | Exportatorul pe etape | Returnează exportatorul pe etape. |
| 3. | public void Interrupt() |  |  | Întrerupe executarea etapei. |
| 4. | protected void SetError(string errorMessage) | *errorMessage* – mesajul de eroare |  | Setarea mesajului de eroare. |
| 5. | protected IStatelessSession Session |  | Sesiunea curentă | Returnează sesiunea curentă. |
| 6. | protected void WriteLog(LogType logType,string message, Exception exception = null) | *logType* – tipul de jurnalizare;  *message* – mesajul;  *exception* – excepția (opțional) |  | Jurnalizează mesajul respec-tiv conform tipului de jurnalizare și excepției apărute. |

## Subcomponenta CEC.QuartzServer.Jobs.Export.SaiseRepository

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr. | Funcționalitate | Date de intrare | Date de ieșire | Descriere |
| 1. | public SaiseElection GetElection(long saiseElectionId) | *saiseElectionId* – numărul de identi-ficare al scrutinului electoral | Scrutinul electoral cu numărul de identificare respectiv. | Returnează scrutinul electoral cu numărul de identificare respectiv. |
| 2. | public IList <SaisePollingStation> GetPollingStation(long electionId) | *electionId* – numărul de identi-ficare al scrutinului electoral | Secțiile de votare asociate scrutinului electoral | Returnează secțiile de votare asociate scrutinului electoral. |
| 3. | public Voter GetSaiseVoterByIdnp (long idnp) | *idnp* – numărul de identitate al alegătorului. | Alegătorul cu numărul de identitate respectiv. | Returnează alegătorul cu numărul de identitate respectiv. |
| 4. | public void SaveOrUpdate (SaiseEntity entity) | *entity* - entitatea |  | Salvează sau actualizează entitatea respectivă. |
| 5. | public IStatelessSession Session |  | Sesiunea curentă | Returnează sesiunea curentă |
| 6. | public AssignedVoter GetAssignedVoter (Voter voter, SaiseElection saiseElection) | *voter –* alegătorul;  *saiseElection* – scrutinul electoral. | Alegătorul alocat la scrutinul elec-toral respectiv. | Returnează alegătorul alocat la scrutinul electoral res-pectiv. |
| 7. | public IEnumerable <SaiseElection> GetElectionsByDate (DateTime electionDate) | *electionDate* – data scrutinului electoral | Lista scrutinelor electorale din data respectivă. | Returnează lista scrutinelor electorale din data respectivă. |
| 8. | public IEnumerable <SaiseDistrict> GetDistrictsForElection(long electionId) | *electionId –* numărul de identi-ficare al scrutinului electoral | Lista raioanelor secțiilor de votare asociate cu scruti-nul electoral res-pecttiv. | Returnează lista raioanelor secțiilor de votare asociat cu scrutinul electoral respectiv. |
| 9. | public IEnumerable<AssignedPollingStation> GetOpenedPollingStations(long electionId) | *electionId –* numărul de identi-ficare al scrutinului electoral | Lista secțiilor de votare deschise associate cu scrutinul electoral respectiv. | Returnează lista secțiilor de votare deschise associate cu scrutinul electoral respectiv. |
| 10. | public long GetBaseListVotedCount(long electionId, long pollingStationId, TimeSpan controlTime) | *electionId –* numărul de identi-ficare al scrutinului electoral;  *pollingStationId* – numărul de identi-ficare al secției de votare;  *controlTime* – timpul de control. | Numărul de alegători care au votat pe lista de bază până la momentul de timp de control. | Returnează numărul de alegători care au votat pe lista de bază până la momentul de timp de control. |
| 11. | public long GetSupplimentaryListVotedCount(long electionId, long pollingStationId, TimeSpan controlTime) | *electionId –* numărul de identi-ficare al scrutinului electoral;  *pollingStationId* – numărul de identi-ficare al secției de votare;  *controlTime* – timpul de control. | Numărul de alegători care au votat pe lista suplimentară până la momentul de timp de control. | Returnează numărul de alegători care au votat pe lista suplimentară până la momentul de timp de control. |
| 12. | public IEnumerable <SaiseVillage> GetAllVillagesForDistrict(long districtId, long electionId ) | *districtId –* numărul de identificare al raionului;  *electionId –* numărul de identi-ficare al scrutinului electoral. | Lista de localități din raionul res-pectiv, asociate cu scrutinul elec-toral respectiv. | Returnează lista de localități din raionul respectiv, asociate cu scrutinul electoral respectiv. |
| 13. | public IEnumerable <SaisePollingStation> GetAllPollingStationsByVillage(long electionId, long villageId) | *electionId –* numărul de identi-ficare al scrutinului electoral;  *villageId –* numărul de identificare al localității. | Lista secțiilor de votare din locali-tatea respectivă, asignate la scruti-nul electoral res-pectiv. | Returnează lista secțiilor de votare din localitatea respectivă, asignate la scrutinul electoral respectiv. |
| 14. | public void InsertVotersInTemporaryTable(List<Voter> voters) | *voters –* lista de alegători. |  | Salvează alegătorii într-o tabelă temporară. |
| 15. | public void CopyDataFromTemporaryTableToVoter() |  |  | Copierea alegătorilor din tabela temporară în tabela cu alegători. |
| 16. | public void CreateVoterTemporaryTable() |  |  | Crearea tabelei temporare pentru alegători. |
| 17. | public void AssignVoterToElection (long saiseElectionId) | *saiseElectionId* – numărul de identi-ficare al scrutinului electoral. |  | Asignarea alegătorilor la scrutinul electoral respectiv. |
| 18. | public void ClearAssignedVoters (long electionId) | *electionId* – numărul de identi-ficare al scrutinului electoral. |  | Elimină asignările la scrutinul electoral dat. |
| 19. | public void DisableIndexesOnVoter () |  |  | Dezactivarea indecșilor la tabela de alegători. |
| 20. | public void EnableIndexesOnVoter() |  |  | Activarea indecșilor la tabela de alegători. |
| 21. | public void UpdateAssignedVoterVillage(long electionId) | *electionId* – numărul de identi-ficare al scrutinului electoral. |  | Actualizarea localității conform localității secției de votare asociate pentru alegătorii cu localitatea lipsă asignați scrutinului electoral dat. |
| 22. | public void DropVoterTemporaryTable() |  |  | Elimină tabela temporară a alegătorilor. |
| 23. | private void ClearVoterTemporaryTable() |  |  | Curăță tabela temporară a alegătorilor. |
| 24. | private void ExecuteQuery(string query) | *query –* interogarea SQL |  | Execută interogarea SQL data. |
| 25. | public void FixElectionTurnout (long electionId) | *electionId* – numărul de identi-ficare al scrutinului electoral. |  | Se apelează procedura stocată *pDK\_Final\_FixElectionTurnout*, transmițând numărul de identificare al scrutinului electoral ca parametru. |

## Subcomponenta CEC.QuartzServer.Jobs.Export.UpdateConflictedSaiseVoterStageExecuter

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr. | Funcționalitate | Date de intrare | Date de ieșire | Descriere |
| 1. | protected override void ExecuteStateInternal() |  |  | Execută o etapă a actualizării alegătorilor aflați în conflict și setează starea. |

## Subcomponenta CEC.QuartzServer.Jobs.Export.UpdateSaiseVoterStageExecuter

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr. | Funcționalitate | Date de intrare | Date de ieșire | Descriere |
| 1. | protected override void ExecuteStateInternal() |  |  | Execută o etapă a actualizării alegătorilor și setează starea. |

## Subcomponenta CEC.QuartzServer.Jobs.Export.ValidateElectionStageExecuter

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr. | Funcționalitate | Date de intrare | Date de ieșire | Descriere |
| 1. | protected override void ExecuteStateInternal() |  |  | Execută o etapă a validării scrutinului electoral și setează starea. |

## Subcomponenta CEC.QuartzServer.Jobs.Export.ValidatePollingStationsStageExecuter

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr. | Funcționalitate | Date de intrare | Date de ieșire | Descriere |
| 1. | protected override void ExecuteStateInternal() |  |  | Execută o etapă a validării secțiilor de votare și setează starea. |

## Subcomponenta CEC.QuartzServer.Jobs.Export.ValidateRegIonsStageExecuter

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr. | Funcționalitate | Date de intrare | Date de ieșire | Descriere |
| 1. | protected override void ExecuteStateInternal() |  |  | Execută o etapă a validării regiunilor și setează starea. |

## Subcomponenta CEC.QuartzServer.Jobs.Import.AlegatorProcessorJob

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr. | Funcționalitate | Date de intrare | Date de ieșire | Descriere |
| 1. | protected internal override void ExecuteInternal (IJobExecutionContext context) | *context* – contextul de executare |  | Execută procesarea tuturor alegătorilor noi în vederea importării acestora. |

## Subcomponenta CEC.QuartzServer.Jobs.Import.RsaUserProcessorJob

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr. | Funcționalitate | Date de intrare | Date de ieșire | Descriere |
| 1. | protected internal override void ExecuteInternal (IJobExecutionContext context) | *context* – contextul de executare |  | Execută procesarea tuturor utilizatorilor RSA noi în vederea importării acestora. |

## Subcomponenta CEC.QuartzServer.Jobs.Import.RspModificationImporterJob

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr. | Funcționalitate | Date de intrare | Date de ieșire | Descriere |
| 1. | protected internal override void ExecuteInternal (IJobExecutionContext context) | *context* – contextul de executare |  | Execută procesarea modifi-cărilor din RSP în vederea importării acestora. |
| 2. | public void Interrupt() |  |  | Întreruperea execuției importatorului. |

## Subcomponenta CEC.QuartzServer.Jobs.Import.RspModificationProcessorJob

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr. | Funcționalitate | Date de intrare | Date de ieșire | Descriere |
| 1. | protected internal override void ExecuteInternal (IJobExecutionContext context) | *context* – contextul de executare |  | Execută procesarea tuturor modificărilor noi din RSP în vederea importării acestora. |

## Subcomponenta CEC.QuartzServer.Jobs.Import.RemoveStayStatementsJob

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr. | Funcționalitate | Date de intrare | Date de ieșire | Descriere |
| 1. | protected internal override void ExecuteInternal (IJobExecutionContext context) | *context* – contextul de executare |  | Execută procesarea anulării tuturor declarațiilor de ședere. |
| 2. | public void Interrupt() |  |  | Întreruperea execuției procesului. |

## Subcomponenta CEC.QuartzServer.Jobs.ListsPrinting.ListsPrintingJob

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr. | Funcționalitate | Date de intrare | Date de ieșire | Descriere |
| 1. | protected internal override void ExecuteInternal (IJobExecutionContext context) | *context* – contextul de executare |  | Execută procesarea exportării listelor în vederea tipăririi acestora. |

## Subcomponenta CEC.QuartzServer.Jobs.Reporting.PrintOutBaseJob

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr. | Funcționalitate | Date de intrare | Date de ieșire | Descriere |
| 1. | public NetworkCredential GetCredentials() |  | Utilizatorul și parola serviciului de raportare. | Returnează utilizatorul și parola serviciului de raportare. |
| 2. | protected void TryParseAndSetIgnorableDistrictsNumbers() |  |  | Setează numerele raioanelor ignorabile. Aruncă o excepție în cazul când transformarea numărului raionului din șirul de caractere din setările de configurare eșuează. |
| 3. | protected void WriteToFile(string path, string fileName, XDocument xdoc) | *path* – calea spre fișier;  *fileName* – denumirea fișierului;  *xdoc* – documentul pentru salvarea |  | Salvarea documentului în fișier. |
| 4. | protected XDocument CreateXDocForSiteExport<T>(IEnumerable<T> list, RefAction<T, string, string> action) | *list* – lista de entități pentru exportare;  *action* – acțiunea de extragere a număru-lui de identificare și a textului din fiecare entitate. |  | Creează un *XDocument* în baza listei de entități date, pentru exportarea ulterioară în raport. |
| 5. | protected void LogRequestException(WebException webException) | *webException* – excepția |  | Jurnalizarea excepției date. |

## Subcomponenta CEC.QuartzServer.Jobs.Reporting.ResultsReportsPrintOutJob

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr. | Funcționalitate | Date de intrare | Date de ieșire | Descriere |
| 1. | protected internal override void ExecuteInternal (IJobExecutionContext context) | *context* – contextul de executare |  | Execută raportarea rezulta-telor în vederea tipăririi acestora. |

## Subcomponenta CEC.QuartzServer.Jobs.Reporting.TurnOutReportsPrintOutJob

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr. | Funcționalitate | Date de intrare | Date de ieșire | Descriere |
| 1. | protected internal override void ExecuteInternal (IJobExecutionContext context) | *context* – contextul de executare |  | Execută raportarea prezenței la vot în vederea tipăririi acesteia. |

## Subcomponenta CEC.QuartzServer.Jobs.SAISE.TurnOutFixationJob

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr. | Funcționalitate | Date de intrare | Date de ieșire | Descriere |
| 1. | protected internal override void ExecuteInternal (IJobExecutionContext context) | *context* – contextul de executare |  | Execută procesarea fixării prezenței la vot. |
| 2. | public void Interrupt() |  |  | Întreruperea procesării fixării prezenței la vot. |

## Subcomponenta CEC.QuartzServer.Service.Program

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr. | Funcționalitate | Date de intrare | Date de ieșire | Descriere |
| 1. | static void Main() |  |  | Inițializează și startează serverul Quartz, destinat executării proceselor de lungă durată. |

## Subcomponenta CEC.QuartzServer.Service.QuartzService

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr. | Funcționalitate | Date de intrare | Date de ieșire | Descriere |
| 1. | protected override void OnStart(string[] args) | *args* – parametrii serviciului |  | Inițializează și startează serverul Quartz la startarea serviciului. |
| 2. | protected override void OnStop() |  |  | Stopează serverul Quartz la stoparea serviciului. |
| 3. | protected override void Dispose(bool disposing) | *disposing* – indicatorul finisării executării serviciu-ilor serverului |  | Finisează activitatea serveru-lui Quartz la finisarea activității serviciului. |

## Subcomponenta RSPChecker.Program

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr. | Funcționalitate | Date de intrare | Date de ieșire | Descriere |
| 1. | static void Main(string[] args) | *args –* parametrii aplicației consolă. |  | Obține datele despre persoana cu numărul de identitate din primul parametru. Dacă numărul de parametri transmiși nu este 1, se aruncă o excepție. |

# Componenta CEC.SRV.Domain

## Entitatea Abroad Voter Registration

|  |  |  |
| --- | --- | --- |
| Nr. | Proprietate | Descriere |
| 1. | Person Person | persoana care se înregistrează |
| 2. | string AbroadAddress | adresa din străinătate |
| 3. | string AbroadAddressCountry | țara aflării |
| 4. | double AbroadAddressLat | latitudinea adresei |
| 5. | double AbroadAddressLong | longitudinea adresei |
| 6. | string ResidenceAddress | adresa de reședință |
| 7. | string Email | adresa de e-mail |
| 8. | string IpAddress | adresa IP a calculatorului |
| 9. | DateTimeOffset Created | data creării |
| 10. | DateTime CreationDate | data creării |

## Entitatea Additional User Info

|  |  |  |
| --- | --- | --- |
| Nr. | Proprietate | Descriere |
| 1. | string FirstName | prenumele utilizatorului |
| 2. | string LastName | numele utilizatorului |
| 3. | string FullName | numele complet |
| 4. | DateTime? DateOfBirth | data nașterii |
| 5. | string Email | adresa de e-mail |
| 6. | string LandlinePhone | numărul de telefon fix |
| 7. | string MobilePhone | numărul de telefon mobil |
| 8. | string WorkInfo | informații despre serviciu |
| 9. | Gender Gender | genul utilizatorului |

## Entitatea Address

|  |  |  |
| --- | --- | --- |
| Nr. | Proprietate | Descriere |
| 1. | Street Street | strada adresei |
| 2. | int? HouseNumber | numărul casei |
| 3. | string Suffix | sufixul numărului casei |
| 4. | PollingStation PollingStation | secția de votare asociată |
| 5. | BuildingTypes BuildingType | tipul de apartament |
| 6. | GeoLocation GeoLocation | poziția geolocală |

## Entitatea Address With Count Of People

|  |  |  |
| --- | --- | --- |
| Nr. | Proprietate | Descriere |
| 1. | Address Address | adresa |
| 2. | int PeopleCount | numărul populației |

## Entitatea Doc Type Code

|  |  |  |
| --- | --- | --- |
| Nr. | Proprietate | Descriere |
| --- | --- | --- |

## Entitatea Reg Type Code

|  |  |  |
| --- | --- | --- |
| Nr. | Proprietate | Descriere |
| --- | --- | --- |

## Entitatea Sex Code

|  |  |  |
| --- | --- | --- |
| Nr. | Proprietate | Descriere |
| --- | --- | --- |

## Entitatea Street Type Code

|  |  |  |
| --- | --- | --- |
| Nr. | Proprietate | Descriere |
| 1 | string Docprint | numărul documentului |
| 2. | long RspStreetTypeCodeId | numărul de identificare al codului tipului de stradă din RSP |

## Entitatea Election

|  |  |  |
| --- | --- | --- |
| Nr. | Proprietate | Descriere |
| 1 | ElectionType ElectionType | tipul scrutinului electoral |
| 2. | DateTime ElectionDate | data scrutinului electoral |
| 3. | long? SaiseId | numărul de identificare din SAISE |
| 4. | bool AcceptAbroadDeclaration | indicatorul faptului că se acceptă sau nu declarații de ședere în străinătate |
| 5. | string Comments | comentarii |

## Entitatea Event

|  |  |  |
| --- | --- | --- |
| Nr. | Proprietate | Descriere |
| 1. | EventTypes EventType | tipul evenimentului |
| 2. | string EntityType | tipul entității |
| 3. | long EntityId | numărul de identificare al entității |

## Entitatea Geo Location

|  |  |  |
| --- | --- | --- |
| Nr. | Proprietate | Descriere |
| 1. | double? Latitude | latitudinea poziției geolocale |
| 2. | double? Longitude | longitudinea poziției geolocale |

## Entitatea Identity User Region View

|  |  |  |
| --- | --- | --- |
| Nr. | Proprietate | Descriere |
| 1. | Region Region | regiunea |
| 2. | SRVIdentityUser IdentityUser | utilizatorul |

## Entitatea Linked Region

|  |  |  |
| --- | --- | --- |
| Nr. | Proprietate | Descriere |
| 1. | IReadOnlyCollection<Region> Regions | regiunile legate prin intermediul legăturii curente |

## Entitatea Linked Regions Full Name

|  |  |  |
| --- | --- | --- |
| Nr. | Proprietate | Descriere |
| 1. | LinkedRegion LinkedRegion | legătura de regiuni |
| 2. | string FullyQualifiedName | denumirea completă |

## Entitatea Notification

|  |  |  |
| --- | --- | --- |
| Nr. | Proprietate | Descriere |
| 1. | string Message | mesajul notificării |
| 2. | Event Event | evenimentul notificat |
| 3. | IReadOnlyCollection<NotificationReceiver> Receivers | receptorii notificării |
| 4. | DateTime CreateDate | data creării notificării |

## Entitatea Notification Receiver

|  |  |  |
| --- | --- | --- |
| Nr. | Proprietate | Descriere |
| 1. | Notification Notification | notificarea |
| 2. | SRVIdentityUser User | utilizatorul notificat |
| 3. | bool NotificationIsRead | indică faptul că notificarea este citită sau nu |

## Entitatea Person

|  |  |  |
| --- | --- | --- |
| Nr. | Proprietate | Descriere |
| 1. | string FirstName | prenumele persoanei |
| 2. | string MiddleName | patronimicul |
| 3. | string Surname | numele |
| 4. | string FullName | numele complet |
| 5. | string Idnp | numărul de identitate IDNP |
| 6. | Gender Gender | genul |
| 7. | DateTime DateOfBirth | data nașterii |
| 8. | long? AlegatorId | numărul de identificare al alegătorului |
| 9. | PersonDocument Document | documentul |
| 10. | bool ExportedToSaise | indică faptul că a fost exportat către SAISE sau nu |
| 11. | IReadOnlyCollection <PersonAddress> Addresses | lista de adrese |
| 12. | IReadOnlyCollection <PersonStatus> PersonStatuses | lista de statute |
| 13. | PersonStatus CurrentStatus | statutul curent |
| 14. | PersonAddress EligibleAddress | adresa eligibilă |
| 15. | int Age | vârsta |
| 16. | string Comments | comentarii |

## Entitatea Person Address

|  |  |  |
| --- | --- | --- |
| Nr. | Proprietate | Descriere |
| 1. | Person Person | persoana |
| 2. | Address Address | adresa persoanei |
| 3. | PersonFullAddress PersonFullAddress | adresa completă a persoanei |
| 4. | DateTime DateOfRegistration | data înregistrării |
| 5. | DateTime? DateOfExpiration | data expirării |
| 6. | int? ApNumber | numărul apartamentului |
| 7. | string ApSuffix | sufixul numărului apartamentului |
| 8. | bool IsEligible | indică faptul că adresa este eligibilă sau nu |
| 9. | PersonAddressType PersonAddressType | tipul de adresă de persoană |

## Entitatea Person By Conflict

|  |  |  |
| --- | --- | --- |
| Nr. | Proprietate | Descriere |
| 1. | Person Person | persoana |

## Entitatea Person Document

|  |  |  |
| --- | --- | --- |
| Nr. | Proprietate | Descriere |
| 1. | string Seria | seria documentului |
| 2. | string Number | numărul |
| 3. | DateTime? IssuedDate | data eliberării |
| 4. | string IssuedBy | cine a eliberat |
| 5. | DateTime? ValidBy | data expirării |
| 6. | DocumentType Type | tipul documentului |
| 7. | string DocumentNumber | numărul documentului |

## Entitatea Person Full Address

|  |  |  |
| --- | --- | --- |
| Nr. | Proprietate | Descriere |
| 1. | Region Region | regiunea adresei |
| 2. | bool RegionHasStreets | indică faptul că regiunea are străzi sau nu |
| 3. | PollingStation PollingStation | secția de votare |
| 4. | Person Person | persoana |
| 5. | PersonAddressType PersonAddressType | tipul adresei de persoană |
| 6. | string FullAddress | adresa completă |
| 7. | int? ApNumber | numărul apartamentului |
| 8. | string ApSuffix | sufixul numărului apartamentului |
| 9. | bool IsEligible | indică faptul că adresa este eligibilă sau nu |
| 10. | IdentityUserRegionView AssignedUser | utilizatorul asignat |

## Entitatea Person Status

|  |  |  |
| --- | --- | --- |
| Nr. | Proprietate | Descriere |
| 1. | Person Person | persoana |
| 2. | PersonStatusType StatusType | tipul statutului |
| 3. | string Confirmation | confirmarea |
| 4. | bool IsCurrent | indicatorul faptului că statutul este curent sau nu |

## Entitatea Polling Station

|  |  |  |
| --- | --- | --- |
| Nr. | Proprietate | Descriere |
| 1. | string Number | numărul secției de votare |
| 2. | string SubNumber | circumscripția |
| 3. | Region Region | regiunea |
| 4. | string ContactInfo | informații de contact |
| 5. | IReadOnlyList<Address> Addresses | adresele asociate |
| 6. | long? SaiseId | numărul de identificare pentru SAISE |
| 7. | GeoLocation GeoLocation | poziția geolocală |
| 8. | string Location | locația |
| 9. | Address PollingStationAddress | adresa secției de votare |
| 10. | OwingCircumscription | circumscripția proprie |
| 11. | string FullNumber | numărul complet |
| 12. | PollingStationTypes PollingStationType | tipul secției de votare |

## Entitatea Polling Station Statistics

|  |  |  |
| --- | --- | --- |
| Nr. | Proprietate | Descriere |
| 1. | long PollingStationId | numărul de identificare al secției de votare |
| 2. | long RegionId | numărul de identificare al regiunii |
| 3. | string RegionName | denumirea regiunii |
| 4. | string PollingStation | secția de votare |
| 5. | int VotersCount | numărul de alegători |

## Entitatea Polling Station With Full Address

|  |  |  |
| --- | --- | --- |
| Nr. | Proprietate | Descriere |
| 1. | PollingStation PollingStation | secția de votare |
| 2. | string FullAddress | adresa completă |
| 3. | int TotalAddress | numărul total de adrese |

## Entitatea Problematic Data Polling Station Statistics

|  |  |  |
| --- | --- | --- |
| Nr. | Proprietate | Descriere |
| 1. | long RegionId | numărul de identificare al regiunii |
| 2. | string FullRegionName | denumirea completă a regiunii |
| 3. | string PollingStation | secția de votare |
| 4. | int VotersCount | numărul de alegători |

## Entitatea Public Administration

|  |  |  |
| --- | --- | --- |
| Nr. | Proprietate | Descriere |
| 1. | string Name | prenumele conducătorului |
| 2. | string Surname | numele conducătorului |
| 3. | Region Region | regiunea |
| 4. | ManagerType ManagerType | tipul de conducător |

## Entitatea Region With Fully Qualified Name

|  |  |  |
| --- | --- | --- |
| Nr. | Proprietate | Descriere |
| 1. | Region Region | regiunea |
| 2. | long? SaiseId | numărul de identificare în SAISE |
| 3. | string FullyQualifiedName | denumirea completă |

## Entitatea Region Without Saise Id

|  |  |  |
| --- | --- | --- |
| Nr. | Proprietate | Descriere |
| 1. | Region Region | regiunea |
| 2. | string FullyQualifiedName | denumirea completă |

## Entitatea SRV Identity User

|  |  |  |
| --- | --- | --- |
| Nr. | Proprietate | Descriere |
| 1. | bool IsBlocked | indicatorul faptului că utilizatorul este blocat sau nu |
| 2. | int LoginAttempts | numărul de încercări de autentificare |
| 3. | string Comments | comentarii |
| 4. | bool IsBuiltIn | indicatorul faptului că utilizatorul a fost creat în mod programatic sau nu |
| 5. | DateTimeOffset? BlockedDate | data blocării |
| 6. | DateTimeOffset? LastLogout | ultima delogare |
| 7. | bool IsOnline | indicatorul faptului că utilizatorul este online |
| 8. | DateTimeOffset? LastLogin | data ultimei autentificări |
| 9. | AdditionalUserInfo AdditionalInfo | informații adiționale referitoare la profilul utilizatorului |
| 10. | DateTimeOffset Created | data creării |
| 11. | SRVIdentityUser CreatedBy | utilizatorul care a creat utilizatorul curent |
| 12. | IReadOnlyCollection<Region> Regions | regiunile alocate |

## Entitatea Stay Statement

|  |  |  |
| --- | --- | --- |
| Nr. | Proprietate | Descriere |
| 1. | Person Person | persoana care a depus declarația de ședere. |
| 2. | PersonAddress BaseAddress | adresa de bază a persoanei |
| 3. | PersonAddress DeclaredStayAddress | adresa declarată a persoanei |
| 4. | Election ElectionInstance | scrutinul electoral |

## Entitatea Street With count Of Addresses

|  |  |  |
| --- | --- | --- |
| Nr. | Proprietate | Descriere |
| 1. | Street Street | strada. |
| 2. | int HousesCount | numărul de case |

## Entitatea Export Polling Station

|  |  |  |
| --- | --- | --- |
| Nr. | Proprietate | Descriere |
| 1. | string StatusMessage | mesajul de statut |
| 2. | PrintSession PrintSession | sesiunea de tipărire |
| 3. | PollingStation PollingStation | secția de votare |

## Entitatea Print Entity

|  |  |  |
| --- | --- | --- |
| Nr. | Proprietate | Descriere |
| 1. | DateTimeOffset? StartDate | data de începere a tipăririi |
| 2. | DateTimeOffset? EndDate | data de finisare a tipăririi |
| 3. | PrintStatus Status | statutul de tipărire |

## Entitatea Print Session

|  |  |  |
| --- | --- | --- |
| Nr. | Proprietate | Descriere |
| 1. | Election Election | scrutinul electoral |
| 2. | IReadOnlyList <ExportPollingStation> ExportPollingStations | secțiile de votare destinate exportării |

## Entitatea Classifier

|  |  |  |
| --- | --- | --- |
| Nr. | Proprietate | Descriere |
| 1. | string Name | denumirea clasificatorului |
| 2. | string Namerus | denumirea clasificatorului în limba rusă |

## Entitatea Document Type

|  |  |  |
| --- | --- | --- |
| Nr. | Proprietate | Descriere |
| 1. | bool IsPrimary | indicatorul faptului că tipul documentului este primar sau nu |

## Entitatea Election Type

|  |  |  |
| --- | --- | --- |
| Nr. | Proprietate | Descriere |
| --- | --- | --- |

## Entitatea Gender

|  |  |  |
| --- | --- | --- |
| Nr. | Proprietate | Descriere |
| --- | --- | --- |

## Entitatea Gender

|  |  |  |
| --- | --- | --- |
| Nr. | Proprietate | Descriere |
| --- | --- | --- |

## Entitatea Lookup

|  |  |  |
| --- | --- | --- |
| Nr. | Proprietate | Descriere |
| 1. | string Name | denumirea |
| 2. | string Description | descrierea |

## Entitatea Manager Type

|  |  |  |
| --- | --- | --- |
| Nr. | Proprietate | Descriere |
| --- | --- | --- |

## Entitatea Person Address Type

|  |  |  |
| --- | --- | --- |
| Nr. | Proprietate | Descriere |
| --- | --- | --- |

## Entitatea Person Status Type

|  |  |  |
| --- | --- | --- |
| Nr. | Proprietate | Descriere |
| 1. | bool IsExcludable | indicatorul faptului că tipul statutului de persoană este excludabil. |

## Entitatea Region

|  |  |  |
| --- | --- | --- |
| Nr. | Proprietate | Descriere |
| 1. | Region Parent | regiunea părinte |
| 2. | RegionType RegionType | tipul regiunii |
| 3. | IReadOnlyCollection<Region> Children | regiunile copii |
| 4. | long? SaiseId | numărul de identificare pentru aplicația SAISE |
| 5. | long? RegistruId | numărul de identificare în Registru |
| 6. | long? StatisticCode | codul statistic |
| 7. | long? StatisticIdentifier | numărul de identificare statistic |
| 8. | GeoLocation GeoLocation | poziția geolocală |
| 9. | bool HasStreets | indicatorul faptului că regiunea are străzi sau nu |
| 10. | int Level | nivelul regiunii |
| 11. | PublicAdministration PublicAdministration | administrația publică a regiunii |
| 12. | int? Circumscription | numărul circumscripției |

## Entitatea Region Type

|  |  |  |
| --- | --- | --- |
| Nr. | Proprietate | Descriere |
| 1. | byte Rank | rangul tipului de regiune |

## Entitatea Street

|  |  |  |
| --- | --- | --- |
| Nr. | Proprietate | Descriere |
| 1. | long? RopId | numărul de identificare extern din Registrul Populației |
| 2. | long? SaiseId | numărul de identificare pentru aplicația SAISE |
| 3. | Region Region | regiunea străzii |
| 4. | StreetType StreetType | tipul străzii |
| 5. | IReadOnlyCollection<Address> Addresses | lista de adrese |

## Entitatea Street Type

|  |  |  |
| --- | --- | --- |
| Nr. | Proprietate | Descriere |
| --- | --- | --- |

## Entitatea Alegator Data

|  |  |  |
| --- | --- | --- |
| Nr. | Proprietate | Descriere |
| 1. | long? AlegatorId | numărul de identificare al alegătorului |
| 2. | long? AlegatorPollingStationId | numărul de identificare al secției de votare a alegătorului |
| 3. | string DocSeria | seria documentului |
| 4. | string DocNumber | numărul documentului |
| 5. | string StreetName | denumirea străzii |
| 6. | string HouseNr | numărul casei |
| 7. | string Apartment | apartamentul |
| 8. | int PersonStatus | statutul persoanei |

## Entitatea Import Statistics

|  |  |  |
| --- | --- | --- |
| Nr. | Proprietate | Descriere |
| 1. | DateTime Date | data importării |
| 2. | long New | numărul de persoane noi |
| 3. | long Conflicted | numărul de persoane în conflict |
| 4. | long Updated | numărul de persoane actualizate |
| 5. | long Total | numărul total de persoane |
| 6. | long ChangedStatus | numărul de modificări de statut |
| 7. | long ResidenceChnaged | numărul de modificări de adresă de reședință |
| 8. | long Error | numărul de erori |
| 9. | long Region | numărul de identificare al regiunii |

## Entitatea Mapping Address

|  |  |  |
| --- | --- | --- |
| Nr. | Proprietate | Descriere |
| 1. | long SrvAddressId | numărul de identificare al adresei SRV |
| 2. | long RspAdministrativeCode | codul administrativ RSP |
| 3. | long RspStreetCode | codul străzii RSP |
| 4. | int? RspHouseNr | numărul casei RSP |
| 5. | string RspHouseSuf | sufixul numărului casei RSP |

## Entitatea Person Raw

|  |  |  |
| --- | --- | --- |
| Nr. | Proprietate | Descriere |
| 1. | string Idnp | numărul de identitate al persoanei |
| 2. | string FirstName | prenumele |
| 3. | string MiddleName | patronimicul |
| 4. | string Surname | numele |
| 5. | DateTime? DateOfBirth | data nașterii |
| 6. | string Gender | genul |

## Entitatea Raw Address

|  |  |  |
| --- | --- | --- |
| Nr. | Proprietate | Descriere |
| 1. | long? ExternalRegionId | numărul de identificare extern al regiunii |
| 2. | string RegionName | denumirea regiunii |
| 3. | long? ExternalStreetId | numărul de identificare extern al străzii |
| 4. | string StreetName | denumirea străzii |
| 5. | string HomeNr | numărul casei |
| 6. | string HomeSuffix | sufixul numărului casei |

## Entitatea Raw Data

|  |  |  |
| --- | --- | --- |
| Nr. | Proprietate | Descriere |
| 1. | SourceEnum Source | sursa |
| 2. | RawDataStatus Status | statutul |
| 3. | string StatusMessage | mesajul statutului |
| 4. | DateTimeOffset Created | data creării |
| 5. | DateTimeOffset? StatusDate | data statutului |
| 6. | string Comments | comentarii |

## Entitatea Raw Document

|  |  |  |
| --- | --- | --- |
| Nr. | Proprietate | Descriere |
| 1. | string Serial | seria documentului |
| 2. | string Number | numărul documentului |
| 3. | string ExternalType | tipul extern |

## Entitatea RSA User

|  |  |  |
| --- | --- | --- |
| Nr. | Proprietate | Descriere |
| 1. | Region Region | regiunea |
| 2. | string LoginName | denumirea de autentificare |
| 3. | string Password | parola |

## Entitatea RSP Conflict Data

|  |  |  |
| --- | --- | --- |
| Nr. | Proprietate | Descriere |
| 1. | string Idnp | numărul de identitate al persoanei în conflict |
| 2. | string LastName | numele |
| 3. | string FirstName | prenumele |
| 4. | string SecondName | patronimicul |
| 5. | DateTime Birthdate | data de naștere |
| 6. | string SexCode | codul sexului |
| 7. | bool Dead | indicatorul faptului că persoana este decedată sau nu. |
| 8. | bool CitizenRm | indicatorul faptului că persoana este cetățean al Republicii Moldova sau nu. |
| 9. | int Doctypecode | codul tipului de document |
| 10. | string Series | seria documentului |
| 11. | string Number | numărul documentului |
| 12. | DateTime? Issuedate | data eliberării documentului |
| 13. | DateTime? Expirationdate | data expirării documentului |
| 14. | bool Validity | indicatorul faptului că documentul este valid sau nu |
| 15. | int? RegTypeCode | codul tipului regiunii |
| 16. | string Region | regiunea |
| 17. | string Locality | localitatea |
| 18. | int Administrativecode | codul administrativ |
| 19. | int Streetcode | codul străzii |
| 20. | int? HouseNr | numărul casei |
| 21. | string HouseSuffix | sufixul numărului casei |
| 22. | int? ApNr | numărul apartamentului |
| 23. | string ApSuffix | sufixul numărului apartamentului |
| 24. | DateTime DateOfRegistration | data înregistrării |
| 25. | DateTime? DateOfExpiration | data expirării |
| 26. | ConflictStatusCode StatusConflictCode | codul conflictului de status |
| 27. | ConflictStatusCode AcceptConflictCode | codul conflictului de acceptare |
| 28. | ConflictStatusCode RejectConflictCode | codul conflictului de respingere |
| 29. | SourceEnum Source | sursa |
| 30. | RawDataStatus Status | statutul |
| 31. | string StatusMessage | mesajul de statut |
| 32. | DateTimeOffset Created | data creării |
| 33. | DateTimeOffset? StatusDate | data statutului |
| 34. | string Comments | comentarii |
| 35. | string StreetName | denumirea străzii |
| 36. | bool IsInConflict | indicatorul faptului că înregistrarea e în conflict |

## Entitatea RSP Conflict Data Admin

|  |  |  |
| --- | --- | --- |
| Nr. | Proprietate | Descriere |
| 1. | string Idnp | numărul de identitate al persoanei în conflict |
| 2. | string LastName | numele |
| 3. | string FirstName | prenumele |
| 4. | string SecondName | patronimicul |
| 5. | DateTime Birthdate | data de naștere |
| 6. | string SexCode | codul sexului |
| 7. | bool Dead | indicatorul faptului că persoana este decedată sau nu. |
| 8. | bool CitizenRm | indicatorul faptului că persoana este cetățean al Republicii Moldova sau nu. |
| 9. | int Doctypecode | codul tipului de document |
| 10. | string Series | seria documentului |
| 11. | string Number | numărul documentului |
| 12. | DateTime? Issuedate | data eliberării documentului |
| 13. | DateTime? Expirationdate | data expirării documentului |
| 14. | bool Validity | indicatorul faptului că documentul este valid sau nu |
| 15. | int? RegTypeCode | codul tipului regiunii |
| 16. | string Region | regiunea |
| 17. | string Locality | localitatea |
| 18. | int Administrativecode | codul administrativ |
| 19. | int Streetcode | codul străzii |
| 20. | int HouseNr | numărul casei |
| 21. | string HouseSuffix | sufixul numărului casei |
| 22. | int ApNr | numărul apartamentului |
| 23. | string ApSuffix | sufixul numărului apartamentului |
| 24. | DateTime DateOfRegistration | data înregistrării |
| 25. | DateTime? DateOfExpiration | data expirării |
| 26. | ConflictStatusCode StatusConflictCode | codul conflictului de status |
| 27. | ConflictStatusCode AcceptConflictCode | codul conflictului de acceptare |
| 28. | ConflictStatusCode RejectConflictCode | codul conflictului de respingere |
| 29. | SourceEnum Source | sursa |
| 30. | RawDataStatus Status | statutul |
| 31. | string StatusMessage | mesajul de statut |
| 32. | DateTimeOffset Created | data creării |
| 33. | DateTimeOffset? StatusDate | data statutului |
| 34. | string Comments | comentarii |
| 35. | string StreetName | denumirea străzii |
| 36. | bool IsInConflict | indicatorul faptului că înregistrarea e în conflict |

## Entitatea RSP Modification Data

|  |  |  |
| --- | --- | --- |
| Nr. | Proprietate | Descriere |
| 1. | string Idnp | numărul de identitate al persoanei în conflict |
| 2. | string LastName | numele |
| 3. | string FirstName | prenumele |
| 4. | string SecondName | patronimicul |
| 5. | DateTime Birthdate | data de naștere |
| 6. | string SexCode | codul sexului |
| 7. | bool Dead | indicatorul faptului că persoana este decedată sau nu. |
| 8. | bool CitizenRm | indicatorul faptului că persoana este cetățean al Republicii Moldova sau nu. |
| 9. | long DocumentTypeCode | codul tipului de document |
| 10. | string Seria | seria documentului |
| 11. | string Number | numărul documentului |
| 12. | DateTime? IssuedDate | data eliberării documentului |
| 13. | DateTime? ValidBydate | data expirării documentului |
| 14. | bool Validity | indicatorul faptului că documentul este valid sau nu |
| 15. | ConflictStatusCode StatusConflictCode | codul conflictului de status |
| 16. | ConflictStatusCode AcceptConflictCode | codul conflictului de acceptare |
| 17. | ConflictStatusCode RejectConflictCode | codul conflictului de respingere |
| 18. | IList<RspRegistrationData> Registrations | lista de înregistrări |

## Entitatea RSP Registration Data

|  |  |  |
| --- | --- | --- |
| Nr. | Proprietate | Descriere |
| 1. | int? RegTypeCode | codul tipului de regiune |
| 2. | string Region | regiunea |
| 3. | string Locality | localitatea |
| 4. | int Administrativecode | codul administrativ |
| 5. | long StreetCode | codul străzii |
| 6. | int? HouseNumber | numărul casei |
| 7. | string HouseSuffix | sufixul numărului casei |
| 8. | int? ApartmentNumber | numărul apartamentului |
| 9. | string ApartmentSuffix | sufixul numărului apartamentului |
| 10. | DateTime DateOfRegistration | data înregistrării |
| 11. | DateTime? DateOfExpiration | data expirării |
| 12. | bool IsInConflict | indicatorul faptului că înregistrarea e în conflict sau nu |
| 13. | RspModificationData RspModificationData | modificările de date RSP asociate înregistrării |
| 14. | string StreetName | denumirea străzii |

## Entitatea Saise Exporter

|  |  |  |
| --- | --- | --- |
| Nr. | Proprietate | Descriere |
| 1. | bool ExportAllVoters | indicatorul faptului că se exportă toate înregistrările sau nu |
| 2. | SaiseExporterStatus Status | statutul exportatorului SAISE |
| 3. | IEnumerable <SaiseExporterStage> Stages | lista de etape |
| 4. | string ErrorMessage | mesajul de eroare |
| 5. | DateTimeOffset? StartDate | data de început |
| 6. | DateTimeOffset? EndDate | data de sfârșit |

## Entitatea Saise Exporter Stage

|  |  |  |
| --- | --- | --- |
| Nr. | Proprietate | Descriere |
| 1. | SaiseExporterStageStatus Status | statutul etapei de exportare către SAISE |
| 2. | SaiseExporterStageType StageType | tipul etapei |
| 3. | string Statistics | statisticile |
| 4. | string ErrorMessage | mesajul de eroare |
| 5. | string Description | descrierea |
| 6. | DateTimeOffset? StartDate | data de început |
| 7. | DateTimeOffset? EndDate | data finalizării |

## Entitatea Address With Polling Station

|  |  |  |
| --- | --- | --- |
| Nr. | Proprietate | Descriere |
| 1. | Address Address | adresa |
| 2. | string FullAddress | denumirea completă a adresei |
| 3. | long RegionId | numărul de identificare al regiunii |

## Entitatea Configuration Setting

|  |  |  |
| --- | --- | --- |
| Nr. | Proprietate | Descriere |
| 1. | string Name | denumirea parametrului de configurare |
| 2. | string Value | valoarea parametrului de configurare |
| 3. | string ApplicationName | denumirea aplicației |

## Entitatea RSP Address Mapping

|  |  |  |
| --- | --- | --- |
| Nr. | Proprietate | Descriere |
| 1. | long SrvRegionId | numărul de identificare al regiunii din RSA |
| 2. | RegionWithFullyQualifiedName SrvRegion | regiunea din RSA, împreună cu denumirea completă |
| 3. | long SrvAddressId | numărul de identificare al adresei din RSA |
| 4. | string SrvFullAddress | denumirea completă a adresei din RSA |
| 5. | RegionWithFullyQualifiedName RspRegion | regiunea din RSP, împreună cu denumirea completă |
| 6. | long RspRegionCode | codul CUATM al regiunii din RSP |
| 7. | long? RspStreetCode | codul CUATM al străzii din RSP |
| 8. | string RspStreetName | denumirea străzii din RSP |
| 9. | int? RspHouseNumber | numărul casei din RSP |
| 10. | string RspHouseSuffix | sufixul casei din RSP |